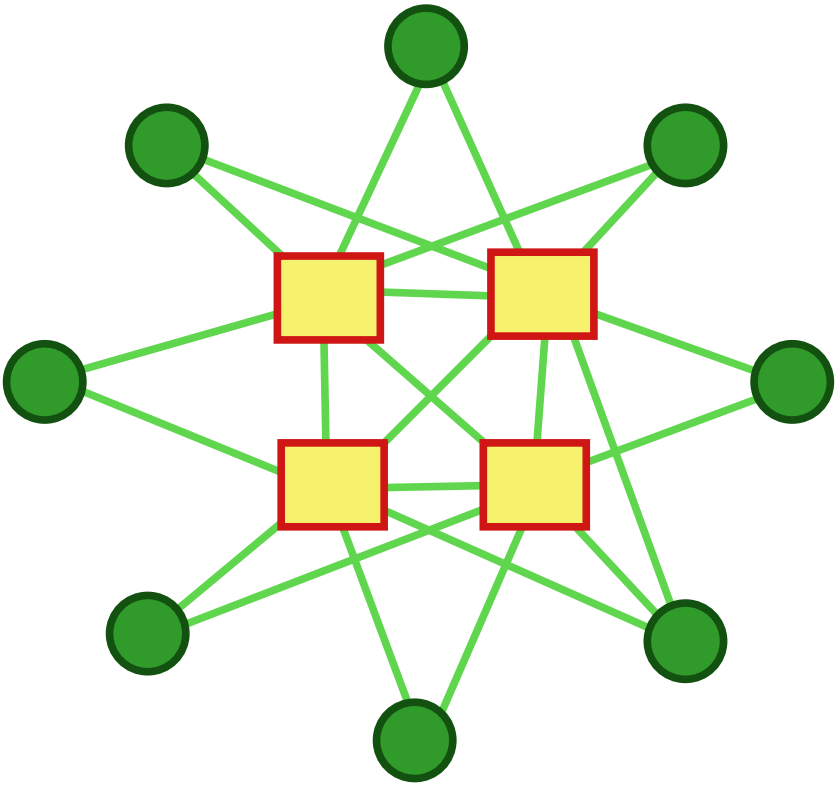
Algorithms and Models for Network Analysis and Design

R. G. Addie



October 23, 2006

Contents

List of Examples

List of Exercises

List of Figures

List of Tables

Preface

Network analysis and design is a great subject. It has everything: theory, practise, power, ideas, service, and a wonderful collection of puzzles. It is a subject which illustrates the application of mathematics to technology, and it is a subject where the power of technology to change the way we live is constantly brought to mind.

The objective of this book is to present the key ideas and models required to make smart decisions about how to manage and design modern integrated terrestrial networks. The careful and hardworking reader will gain a deep understanding of network design concepts and will be able to devise simple and clever solutions to network design problems.

As with any good theory, as a result of reading this book, the reader should emerge with confidence that by means of their own common sense, they can tackle the problems confronting them in their work of planning and designing networks, and arrive at the right decisions.

Structure and conventions of this book

The book is divided into chapters, chapters into sections, and so on. Each chapter also contains a number of examples and exercises. The examples provide guidance for how to solve the exercises. A list of all exercises and examples is provided just after the table of contents.

At the end of the book, an index is provided, as well as a bibliography, and one appendix, which concerns an XML-based language for the description of networks. The index is intended to contain all unfamiliar technical terms. If you are looking for the definition, or an explanation, of an unfamiliar technical term, look it up in the index and refer first of all to the entry in which the page number is italicised. This entry is the one where the term is defined.

Citations to references in the bibliography appear like so: [1]. The fact that a reference is cited is not, by itself, meant to imply that you should obtain that reference and read it. In cases where the reference should be used, in addition to this text, this will be clearly stated.

Introduction to network analysis and design

Some other books with a similar general intent, or appearance, as the present include: [1, 2, 3, 4, 5, 6]. This book differs from other renditions of the subject because this one is committed to a strong working relationship between theory and practice but nevertheless focusses on material which is directly relevant to the working network planner or administrator. Theory which is not needed to support the work of a network administrator is not included. This simplifies the subject considerably. On the other hand, all the methods of analysis and design presented here are rigorously justified on the basis of the well established understanding of traffic and network performance which the reader will obtain from a careful reading of the first part of the book. The book is therefore not a recipe manual containing just quick solutions to an administrators problems.

This is a hard subject to teach and to learn, primarily because the ground is shifting beneath our feet at such a rapid rate, that the entire basis for cost-effective, practical, network design can change overnight – or that's what it feels like anyway.

For example, take a look at the textbooks cited above. Four of these ([2, 3, 4, 7]) purport to be books about designing networks. And yet the approach in each of these books is quite different. How important is switching? What is the role of ATM? What about gigabit ethernet? And how important is it to be able to analyse the performance of networks?

Unfortunately, any book which gives a definite answer to these questions is likely to become rapidly and obviously out of date in the near future, when a new technology comes along and makes the ones we are excited about now seem old-hat.

What about the last question though: "how important is it to be able to analyse the performance of networks?"? A good example of another area of science like the subject of this book is statistics, or, to spell it out more thoroughly, probability and statistics. Probability theory is the foundation for statistics. Without a thorough understanding of probability theory it is impossible to undertake sound statistical analysis. Nevertheless, the character and style of probability theory is quite different from that of statistics.

Probability theory is, as the name suggests, much more theoretical. In probability theory there is a great deal of work on models which do not have a great currency in day-to-day earthly existence. Statistics, on the other hand, is largely focussed on situations which do arise regularly in daily life.

Fortunately, the parts of probability theory which need to be understood in order to tackle statistics are now well understood. It is not necessary to study *all* of probability theory in order to become a good statistician. The aim of this book is to establish the same relationship between performance analysis and network design: to present the theory of how to analyse networks which we need in order to be practical network administrators.

Through all the changes in technology, the principles of performance analysis have remained surprisingly stable.

An intuitive understanding of the issues which ensure good operation of networks can be gained by a variety of means. Perhaps for some, such an understanding comes naturally. However, this is certainly not the case for everyone, and for those of us for whom such an understanding is but a foggy image in the mist, some work on understanding performance issues in networks is a must.

The relationship between theory and practise is often difficult, but also often very productive. In theory, the theory guides the practise, but in practise, practise is based on common sense. But where does the common sense come from? According to the theorist, it comes from the theorists. According to the practitioner, it doesn't come from anywhere: its just common sense. However, as many people have observed, common sense is not so common. And sometimes, also, common sense is quite wrong. As someone once said:

> *Every complex problem has a simple solution which is wrong.*

To find the right solution might require some deeper searching.

So, here is the role of theory: to educate and guide common sense, so that our common sense is well-developed and comes up with the right, or nearly right answers most of the time; and to help solve those really difficult problems where the first solution which comes to mind is not correct.

This book is about this type of theory: the theory which is able to produce the right sort of common sense for practical network administrators and which is able to help sort out the complex problems as they arise.

References

[1] Bruce Davie, Paul Doolan, and Yakov Rekhter. *Switching in IP Networks*. Morgan Kauffman, 1998.

[2] Diane Teare, editor. *Designing CISCO Networks*. CISCO Press, 1999.

[3] Howard C. Berkowitz. *Designing Routing and Switching Architectures for Enterprise Networks*. MacMillan Technical Publishing, 1999.

[4] William Stallings. *High-Speed Networks: TCP/IP and ATM Design Principles*. Morgan Kauffman, 1999.

[5] Pete Loshin, editor. *Big Book of IPsec RFCs: Internet Security Architecture*. Morgan Kauffman, 1999.

[6] James Roberts, Ugo Mocci, and Jorma Virtamo. *Broadband Network Teletraffic, Final Report of Action COST 242*. Springer, 1996.

[7] James D. McCabe. *Practical Computer Network Analysis and Design*. Morgan Kauffman, 1998.

Chapter 1

Overview of Network Analysis and Design

In this chapter we shall develop an understanding of the issues involved in network design and analysis *in the round*. In particular, we shall make a start upon the difficult, but fascinating subject of network architecture, including, in particular, the concept of layers. This will in turn lead us to the important distinction between switching and routing as it applies in packet switched networks.

This discussion will include examples of networks for small, medium, and large organizations. Some of these examples will be used later.

The important subject of documentation for such networks will also be introduced and we shall make a start upon the task of identifying the details which one would need to identify for the purposes of design and analysis.

1.1 Introduction

Network analysis is the science of predicting the behaviour of a network given information about the design of the network and the type of uses to which it will be put. The primary concern is with the *performance* of the network, by which we mean: how many messages, or packets, will be lost, what sort of delay will these messages or packets experience, and how much will these delays vary over time. Periods during which a network is unable to function correctly because of equipment failure must be taken into account in the estimation of performance. Also, failures of *network security* which allow the network to be used for any purpose which is "not permissible" will need to be accounted as failures of network performance.

Network design is the discipline of choosing the components out of which a network should be built, both the types of components and how many of each is needed, and how they should be put together. The goal of all the decisions concerning these components – type, quantity and placement – is to meet the prescribed standards of performance of the network in regard to loss, delay, reliability and security.

As you can see from the definitions, analysis is the more theoretical of the two of these subjects. "Design" can be viewed as "construction" – choosing the components, and putting them together. Whereas network analysis is more of a desk job – putting down all the details on paper and making sense of them.

So what is the link between these two tasks? Do we really need to be able to do analysis to be able to do network design? Let's not answer this question just yet. Suspense is not meant to be a major device of the technical writer, but in the present instance a little suspense can do no harm.

1.1.1 Overview

This book describes practical techniques for analysis and design of integrated communication networks. At all stages the simplest possible technique, or rule of thumb, which achieves the desired outcome, will be emphasized. Networks of a wide range of sizes are considered. The concept of networks being subdivided into layers is emphasized and examples are drawn from a wide range of layers .

The subject matter is subdivided according to three *dimensions*:

1st **Dimension: Phases**

1. Analysis and modeling
2. Measurements
3. Requirements analysis
4. Architecture
5. Equipment choice
6. Design (choice of quantities, placement and routing)

2nd **Dimension: Aspects of performance**

A. Reliability

B. Delay,

C. Loss, and

D. Security.

3rd **Dimension: Examples**

(i) A home

(ii) A Laboratory

(iii) School

(iv) A Campus

(v) Multi-site business

(vi) Multi-location ISP

(vii) A national carrier, or telecommunications company (telco)

1.2 The Network Model

Underlying the study of networks is a *model* that we use intuitively and quite often quite consciously to help us understand how networks work.

But what is a model? Models are used in every aspect of science. In fact, it is likely that virtually everyone uses models unconsciously as a means to guide their interaction with the world on a daily basis. A good example of a model is a *map*. The map shows us how the real world works, in a limited way – it shows how to get from one place to another, for example. Most of the details of the real world are omitted from the map.

In general terms, a model is a representation of something else in which *many* details have been removed, leaving something which can be understood (and used for experiments, or simulations) much more easily.

Real world networks contain cables, ducts, pits, connectors, power supplies, welded or bolted joints, housings and cases, electronic equipment, optical fibers, and so on. The models we use for these networks are usually made up of lines and circles! The interpretation of these lines and circles (or rectangles), varies considerably from case to case.

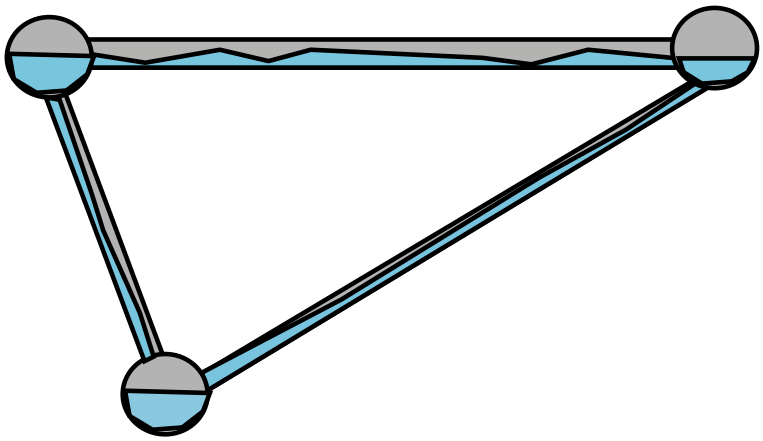
In the real world, the *operation* of a network entails electronic signals passing along the cables, fibers, printed circuit boards, and radio links, and through the chips of the equipment. In the majority of cases, these signals can be *interpreted* as a sequence of bits, bytes, and packets.

In the models, we shall call these signals *traffic* , and our concern with them will be limited to the logical impact that the traffic has on the operation of the network, and the degree to which this *traffic* occupies the resources of the network, i.e. the links and the nodes.

Thus, traffic is an abstraction of the signals occupying the hardware of our network. If a link is capable of conducting *messages* from one point (a node – the source of the link) to another (the destination), and the transmission system sends a fixed holding pattern when such messages are not being sent, we shall regard the signals making up the messages as the *traffic* and the holding pattern signal as *no traffic*. Thus, in our model, the nodes and the links are considered to be either *idle*, or *occupied*.

Figure 1.1 illustrates the model.

Figure 1.1: A Network Model



This model of networks is simple, but sufficiently complex to address quite a number of important issues in the behaviour of networks. The idea, depicted in the diagram by the use of the two colours, is that each network element is *occupied* to a certain level, by whatever is happening at the time in question. We can rescale the level of occupancy of each network element back to a uniform scale, 0 to 1, without loss of generality. In some cases it may be more realistic to restrict occupancy values to either 0 *or* 1. That is to say, a link is either occupied (fully) or it is not occupied. A router is either routing, or it isn't. However, as soon as the activity in a link or node is considered over a longer period of time it makes sense to speak of occupancies taking values *between* 0 and 1 as well as occupancies *equal* to 0 or 1, so we might as well allow the more general values right from the start.

We will need to introduce some additional complexities in the model later, in particular, we will need to introduce *buffering*, and *layers*. These concepts will be explained at such time as they need to be introduced and used.

1.2.1 Terminology

We need an extensive range of terms to describe networks, the equipment in these networks, and the models we use to simulate, experiment with, and understand the operation of these networks.

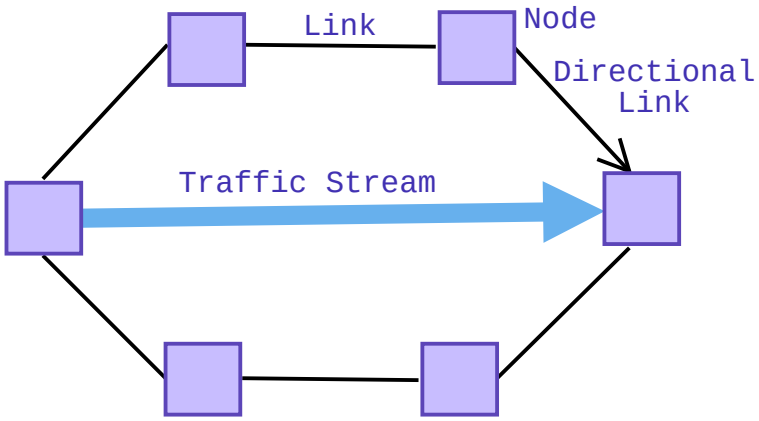
Typically networks are made up of *nodes* and *links*. The nodes carry out processing upon messages or digital signals, or initiate or consume them, whereas the links facilitate the passing of such signals from one node to another. In pictures of networks, the links are drawn as lines and the nodes as circles, or rectangles. Links may be *directional*, which means that the traffic passes in one direction only, or *bidirectional*, meaning that it flows in both directions. The dominant example of a network today is the Internet , which uses the TCP/IP family of protocols (TCP stands for Transmission Control Protocol and IP for Internet Protocol). The Internet and all TCP/IP networks make use, almost exclusively, of links which are bidirectional. Directional links will normally be indicated by the use of an arrowhead at one end of the link, and bidirectional links by a line without arrowheads.

To indicate that there is demand for traffic to pass from one node to another a *traffic stream* may sometimes be used. A traffic stream is always directional, i.e. it has a *source* and a *destination*. Traffic streams are usually denoted by a thick line with a filled arrowhead.

It is also often useful to speak of an origin-destination pair, or, for short, an O-D pair. A O-D pair is just an ordered pair of nodes, except that, by implication, we think of the first in the pair as the *origin* (of a link or a traffic stream) and the second as the *destination*.

Some of these terms are illustrated in Figure 1.2.

Figure 1.2: A Network, with some useful terms



The links in a network may be formed from many different types of equipment, but they always represent some sort of *transmission system* or *communication link*. We may need to distinguish between *leased* transmission facilities and *purchased equipment*. In the case of a leased facility, it is most likely that it is provided by a network owned by the organization providing the service. Purchased equipment may take the form of a cable connection, typically buried in the ground if traversing a significant distance, or a radio link, including the possibility of a radio link which is relayed by a satellite. Cable connections may take the form of *multi-pair cable*, *optical fiber*, or *coaxial cable*. All of these are in regular use all over the world in appropriate contexts..

The signals being passed over links are mostly *digital*, that is to say the signal can be interpreted as made up of numbers, typically 0's and 1's. For this reason it possible to measure the speed of transmission on a link in terms of bits per second (bit/s). Other possible units for measuring transmission speed, some of which are more appropriate for faster links, include bytes/s, kilobits per second (kbit/s), Megabytes per second (Mbyte/s), Megabits per second (Mbit/s), gigabits per second (Gbit/s) , and terabits per second (Tbit/s).

The need for terms for huge quantities of money is fortunately a little less than that for very high transmission speeds. Nevertheless, we shall have occasion to use the terms k$ and M$ for thousands of dollars and millions of dollars, respectively.

Traffic streams can also be measured in the same units as used for links, i.e. kbit/s, Mbit/s, and so on. A traffic stream is a way of representing the amount of traffic which would be carried if there was unlimited capacity, so if we say that a traffic stream had a capacity of 8 Mbit/s we mean that if a direct link was placed between the same source and destination and this link had a much higher capacity than 8 Mbit/s then the amount of traffic which would actually be carried would be 8 Mbit/s.

As we shall see, in Chapter 3, if the capacity of the network between the source and destination is more than the capacity of the traffic stream, but only a small amount more than this, then the amount of traffic which will be carried will be less than 8 Mbit/s. This gives rise to an important distinction, between *offered traffic*, which is the traffic which would be carried if there was unlimited capacity on the intervening network, and *carried traffic*, which is the traffic actually carried on the network which is in place.

Within buildings, and in some cases between nearby buildings, it is common to use *twisted pair* cable. Twisted pair cable typically contains 8 separate strands. As with telephone wiring, it is common to leave *some* of the pairs in a cable unused. For example, in the most common Local Area Network (LAN) – the 10 Mbit/s Ethernet – only

four strands are actually connected, although the other 4 wires in the twisted pair cable have a passive role in the operation of this network, by assisting in protecting the signal from interference by electromagnetic interference.

The nodes, also, can be formed from many different types of equipment, notably hubs – used to connect together the other components of a LAN; switches – also used to connect together the links of a LAN, but the term can be used to refer to much more complex devices; and routers – the nodes which are used to connect different LAN's together to form "an Internet", or to connect different internets together, or to connect an internet to "the Internet".

The term *internet*, means a network which connects other networks (typically LAN's) together. In principle an internet could use any protocol architecture, or collection of protocol architectures, e.g. IPX or the Appletalk protocol architecture. However, in practice, there are fewer and fewer reasons to be using any architecture other than the TCP/IP family for this purpose. The TCP/IP architecture itself is, of course, evolving, and there is always a possibility of a new network architecture dramatically (or gradually) forcing its way into the scheme of things.

The Internet is a particular example of *an* internet, namely the one we use an ISP (Internet Service Provider) to connect to, and which we collect our email from, etc. The only difference in the spelling of these terms is the initial capital used for "the" Internet.

1.3 Examples of Networks

Let us now describe the seven increasingly complex example networks which will be used in all the following chapters as illustrations. These examples will all be familiar, so there should be no difficulty in understanding the networking goals and constraints in each case. The examples have been made specific rather than generic, so that in susbsequent use we do not need to be concerned with filling in any details in order to fully define the problem under consideration.

1.3.1 A Home

The home is actually a fascinating and diverse example all by itself. On the one hand, the everyday home in the suburbs is now, in many cases, already populated by more than one computer, and quite likely a printer. The basic network requirements of file sharing and remote access to a printer are beginning to emerge in many homes. In the evening, there a strong likelihood that more than one person will want to access the Internet simultaneously. For simple reasons of economy, a network is required to provide access via a single gateway to the Internet either via a modem and a telephone line, or via a high speed connection such as Asynchronous Digital Subscriber Loop (ADSL), a cable television access line, or a satellite service. The software to provide this gateway to the Internet is available in off-the-shelf operating systems such as Linux and recent editions of Microsoft Windows.

In addition, virtually all homes already have three other networks already in place: for power, for telephones and for television. These networks all tend to be set up according to an "electrical" model, by which is meant that service at a series of points is achieved by ensuring that electrical connectivity back to a central location is sufficient to ensure service. The advent of digital and cable television services, not to mention Integrated Services Digital Network (ISDN) telephony, is changing all that.

Some shared use of cable between these different networks would seem to be attactive, although in many cases it remains difficult to achieve because of the differing requirements. However, a recent issue of the IEEE Communications Magazine [1] focussed on the use of the use of power lines (in the house or office) as a communication medium.

These basic networking issues are cropping up in a high proportion of homes now. More futuristic applications of networking (like the dialog between your fridge and the supermarket, or systems for keeping track of FLO's (Frequently Lost Objects), the latter as considered in [2]), do not even need to be considered to realize that the home is going to be a focal point for networking in the immediate future.

Despite all this, it is not obvious, on the face of it, where the difficult design decisions are in wiring a home.

Practically speaking, the hardest part about wiring a home is getting physical access from one place to the next. The home computer network requires Category 5 Ethernet cable, ISDN telephone requires the same, and it is quite likely that digital television can also be accommodated on the same *type* of cable. Analog television is the exception: it requires a type of coaxial cable. Clearly, a specific cable designed to handle telephony, Ethernet, and digital television, would be a cost saver.

As in many much larger networks, a significant optimization issue in the home is how to minimize the effort required to install and maintain the physical network.

For this reason, an attractive home networking solution for many households is the *wireless LAN*, using either the 802.11b or 802.11a protocols. Such a network can be set up in many cases by purchasing one relatively inexpensive *access point* together with wireless network interfaces for each computer. Wireless network interfaces are still a little more expensive than conventional ethernet, however the cost can be expected to steadily reduce.

Other issues include *reliability* and *access*. Services capable of providing more than one call at a time on a single telephone line, or Internet access simultaneous with telephone access, are likely to be attractive to the many households.

Another very important issue is security: there is an expectation that the home should be imune from incursion of hackers from the Internet and protected against infection by viruses. This security requirement needs to be addressed primarily in the configuration of the gateway.

1.3.2 A Laboratory

Let us now consider a typical computer laboratory of the type which nowadays exists in universities and schools.

Requirements

(i) The laboratory houses 20 PCs and a server (which is not accessible to students).

(ii) Heavy load at the start of sessions (when PC's load lots of software, possibly including the operating system).

(iii) Reliability should be "good".

(iv) Internet access (under the control of the system administrator).

(v) Security restrictions.

(vi) Remote monitoring.

Issues

The interesting performance issues begin to be quite important in a computer laboratory. Because classes may begin and end with many students attempting to do the same thing simultaneously (rebooting, accessing certain key files, running certain key programs, printing, saving work to a file server), network performance is constantly being tested.

As always, reliability is also important. Equipment failure is likely to disrupt the work of 20 staff and students simultaneously, which is highly disruptive to the normal work of the institution.

Finally, security is a sensitive and interesting issue in the laboratory context. It is normally desirable to ensure that the computers allocated for use by students have restricted (or non-existent) access to parts of the network where staff or academic staff have their computers. Access to the world-wide web on the Internet at large is likely to be restricted to pass through a proxy server, and might be limited to certain times of the day.

1.3.3 A School

Requirements

A normal school is often small enough to be adequately serviced by one Local Area Network (LAN). There may be advantages in providing more than one LAN and connecting these LANs by means of routers, however, especially in a large school.

The vast majority of local area networks at this time are built using the *Ethernet* protocol. Nowadays there are some important variations on the ethernet protocol – fast ethernet, and gigabit ethernet, which are very similar to the original ethernet, only faster.

Let us now fill in our picture of a typical school and its network.

(i) The school has five buildings, including 2 PC Laboratories with 20 machines each, and 15 classrooms, each requiring 4 Local Area Network (LAN) connections. Admin staff, of whom there are five, in two offices, also require LAN connections. The school requires a file server, email services internally and externally, and web services, internally and externally.

(ii) Load is not concentrated at any particular time, although the range from peak load to low load is considerable. At the peak, almost every computer might seem to be in use, while at the opposite extreme there are times when virtually every computer is idle.

(iii) Reliability should be "good", although an outage of the school network for a whole day could be tolerated, with difficulty. Loss of official files stored on the administration system, however, would be completely unacceptable and corruption of or inappropriate access to those files would also be quite disturbing.

(iv) The school also requires Internet access from throughout the network. The local LAN must use private IP addresses, since the school has only one public IP address.

(v) The school requires a file server, email services internally and externally, and web services, internally and externally.

(vi) Security restrictions; three classes of users: admin, teachers, and students. Access between these classes of user should be restricted. Access *from* the Internet should be severely restricted. Some protection against infection by viruses and delivery of SPAM email should be provided.

(vii) Remote monitoring is desirable, since the system administrator might need to take action from a remote location.

1.3.4 A University Campus

Description

This University (the one in this example) has 25 separate buildings, including 40 computer laboratories, 100 lecture theaters and tutorial rooms, and 500 offices. Virtually every room on the university campus requires network access. Internet access is available via this network.

A Campus network is typically made up of a collection of LANs which are joined together by means of one or more routers. Nowadays campus networks also usually make use of switches, partly in place of hubs, and partly in place of routers. In this example, later on, we will consider the options and see if we can determine some clear principles for deciding between routers, hubs, and switches, and deciding which kind of switches should be used.

Requirements

Tight security restrictions are required to prevent free access between the student, staff and administrative areas of the network.

The University uses its network for an intranet; Internet access to staff and students, including email, both internal and external; file services and database access to staff; web, FTP, and sundry minor services based on access to the external Internet; and video-conferencing facilities both for intra-campus and extra-campus communication between staff and students.

The university also has extensive needs for telephone communication.

The LAN services provided on the campus must meet a reliability standard of no more than 4 hours per year of down-time, on average. Internet access has a target reliability standard of less than 8 hours of down-time per year. The telephone network has a reliability target of less than 4 hours of down-time per year. These targets do not include scheduled outages, which can be arranged to occur at times which cause a minimum of disruption.

The standard for packet loss in the campus network is that packet losses should be at insignificant levels, i.e. fewer than 1 in 10^8 packets are lost due to congestion. Connections which include parts of the Internet outside the campus will, of course, often experience much higher levels of packet loss than this.

The standard for packet delay in the campus network is also that packets experience insignificant delays, or, to be more specific, delays in the order of 1 millisecond, and longer than 2 milliseconds with a probability of less than 10^{-5}.

Issues

The standards just specified may seem quite stringent. On the other hand, a campus is an ideal environment for installation of advanced communication facilities because of the very high levels of demand for information technology services. In fact, there should not be any difficulty in meeting the specified standards unless performance monitoring is neglected, so that performance standards are allowed to fall to unacceptable levels without alarms bringing the problem to anyone's attention – until the problem becomes obvious.

1.3.5 A State-wide Retail Organization

Requirements

(i) This organization has 10 retail outlets, throughout the state or nation where it resides, plus a head office, and a warehouse (all at different locations). Each retail outlet, and the warehouse, require low capacity access to a database stored at the head office.

(ii) Traffic levels between head-office and the warehouse are high, 1 Mbit/s during peak periods.

(iii) The organization is setting up a web server to provide e-tail access to its customers.

(iv) Security is required to prevent interception of communication between offices, to prevent access from sources outside the organisation, and to restrict access to services and documents on the basis of identity within the organisation.

(v) Reliability should be "very good", i.e. less than five hours of down-time per year, on average.

1.3.6 A National Internet Service Provider (ISP)

Requirements

(i) This ISP has 20 regional points of presence, plus sites in Brisbane, Sydney, Melbourne, Adelaide, Perth, Hobart and Canberra, a total of 27 sites. It has 120,000 customers and is growing at the rate of 1,000 new customers each month.

(ii) The traffic levels between the different sites vary from 100 kbit/s up to 1 Mbit/s. The traffic demand for access to the Internet has not been measured, directly, and is rather hard to estimate, however carried traffic levels of 3.6 Mbit/s are expected over the coming month, in a period when the *link* to the Internet has a capacity of 8 Mbit/s. .

(iii) The ISP has access to the rest of the Internet via a leased service at the rate 10 Mbit/s in Sydney, to the network of the largest national Internet carrier, plus a second access line at the rate of 2 Mbit/s in Melbourne.

(iv) Reliability is required to be "excellent", i.e. less than one hour of down-time per year, on average.

(v) Privacy of information customers store on ISP servers needs to be preserved. Privacy of communication across the ISP does not require any special attention (users who require privacy should use a protocol which ensures it).

(vi) The network must be designed to be able to expand readily.

1.3.7 A National Carrier

Requirements

(i) This organization has 2 million subscribers who make use of a range of services: telephony, Internet access, broadband access, and cable TV; the organization provides both retail (direct to the customer: business or residential) and wholesale (to another service provider, or to the carrier itself) services and wishes to introduce data-casting.

(ii) The networks offered by this national carrier have a variety of performance targets. This includes the reliability target, and loss and delay targets.

(iii) The carrier has multiple networks, some of which are designed primarily for the carrier itself – that is to say, it provides services for which it is also the main subscriber. The choice of which layers to provide, how to provide services, when to phase out an existing service, and when to construct a new network service, are all of central importance to the viability of the company.

(iv) The cost of the installing and maintaining the carriers networks, and their maintenance, should be as low as possible.

(v) All the networks must be able to expand rapidly without causing significant disruption to service.

(vi) Authentication services are required for the staff of the company, who are required to use remote administration services extensively. Protection of services against malicious attach is essential. Physical security of equipment is also extremely important. The carrier provides some "guarantees" of security and performance and security which it is legally obliged (at the risk of being sued in the event of a performance or security failure) to underwrite by means of consistent and thorough cultivation and maintenance of security and performance standards.

1.4 Modeling and Performance Analysis

The performance of a network is the quality with which it provides the communication service which it is supposed to provide. If a network is incapable of carrying the required amount of traffic, or cannot convey the load from one place to another without losing information, it will not provide a satisfactory service. The concept of *performance* needs to be defined more precisely, which we will do in the remainder of this chapter and will continue doing in Chapter 3

The crucial performance issues in the operation of a network are:

(i) **loss**: Suppose a network is asked to deliver 200 ethernet frames from point A to point B, but it actually only manages to deliver 150 of these packets. The other packets never arrive at B. This sort of behaviour occurs from time to time in today's networks. One quarter of the packets have been lost, so we say that this network has a loss rate of 25%.

(ii) **delay**: Suppose, in the same situation, after some improvements have been made to the network, no packets are now lost, but some take a long time to reach their destination. A very long time. If this *delay* exceeds a certain level, the packet might as well have been lost anyway. For example, if you are listening to a music broadcast over the Internet, and one packet takes 10 seconds longer than it's companions, this packet will probably arrive at the destination computer too late to be slotted into the audio stream being constructed for the listener's ear.

(iii) **reliability**: Suppose now, that yet more work is done, and now, most of the time, packets arrive within a satisfactory delay at the other end. But, every now and again, perhaps once a month, for a few minutes, the network ceases to operate altogether. In some cases behaviour of this sort is hard to avoid and without being welcome is accepted by the users of the network. It is not completely unusual for example, for a university to become disconnected from the Internet from time to time. As we become more dependent on networks, our expectations of the reliability we need will become greater.

(iv) **security**: This is a relatively new area of concern in the field of network analysis and design. It is conventional to take account of the previous three performance issues explicitly in network design and then consider security, as an afterthought. It is not necessarily clear that security *needs* to be factored into the design from the outset, however it is clear that security is of just as much concern to the ultimate users of networks as the other performance issues, so in this book security will always be considered alongside the other performance measures.

These four performance measures are the ones we will normally be concerned about. We will want to design our networks to be able to exhibit satisfactory loss performance, satisfactory delay performance, satisfactory reliability and satisfactory security.

1.5 Measurements

The techniques of making and the interpretation of measurements could well form a substantial study in its own right. However, the subject of measurements tends more often to be neglected. A counter-example of considerable significance is the paper [3].

In our study of measurements, this paper will also form a crucial ingredient although not for the obvious reason. This paper made use of and reported on a lengthy and complex experiment made up of many measurements on a network carrying TCP/IP traffic. This paper also drew some very important empirical conclusions concerning the statistics of traffic, and these conclusions, which were supported by large quantities of experimental data displaying behaviour at a variety of time scales, can be used in a manner perhaps contrary to the practice in the paper itself, to infer behaviour at a wide range of time scales from observations of a more limited scope.

In order to measure security and reliability, the most important technique is the keeping of records – logging of events. Careful record keeping is actually necessary to support measurements of loss and delay also.

1.6 Requirements Analysis

Requirements analysis is the process of recording and try to understand what users want. Anyone who has followed this path, in any field, will probably report the age-old truth that users never really know what they want – they might think that they do, but really it needs a lot of interpretation.

But the user is always right, no?

Perhaps the user is always right, but it is a good idea to put a lot of thought and effort into bringing the user to this right position. In particular, information about what *services* a user is using, currently, should be available, or at least obtainable. This information can then be used to infer more abstract information, such as how much *traffic* can be expected to flow from one location to another.

In order to deal with traffic, it is useful to make use of the concept of a *traffic stream*. A traffic stream has a *source* and a *destination*, and it represents all the *traffic* which goes, or wants to go, from the source to the destination. We allow traffic streams which come from or go to *abstract* nodes, i.e. nodes which are actually made up of a *set* of nodes. This concept makes it much easier to express certain features of real networks.

Traffic streams also have performance requirements – reliability constraints, security constrains, as well as loss and delay objectives and guarantees. It is useful to attach peformance constraints to specific traffic streams because all traffic does not have the same performance requirements.

1.7 Architecture

Architecture is a bit like design, but it comes before design, and it is more abstract. It sets the framework within which design can happen. For example, choice of protocols: that is part of architecture. Choice of *layers* is also part of architecture.

There are two key architectural principles which are used in a variety of ways in networks: layering, and hierarchy. The layering principle is that any service can be subdivided into sub-layers. The hierarchy principle is that any network can be divided into sub-networks.

As well as considering the architecture of networks as a whole, and the protocols used to transport data around those networks, we shall consider also the *security architecture* adopted in these networks, and their *network management architecture*.

1.8 Equipment Selection

Equipment selection is really part of the design process. The types of equipment under consideration includes hubs, switches, routers, and transmission systems of various sorts including optical fiber, rented bandwidth, satellite, spread-spectrum wireless systems, and microwave point-to-point links. We need to understand this range of equipment types and be able to make right choices between one type and another, when such a choice is possible.

Costs and capacities of individual items of equipment are changing all the time. New players enter the market, old players reduce prices for products as they mature and production runs increase, and from time to time products are withdrawn as a consequence of changing visions of the future. Consequently, there is little point in attempting to provide a standardised product list from which to choose at any one time.

A well know strategy for finding the way home from the airport is as follows: find another car going in the right direction and follow it. Ocassionally you might make a mistake about which car to follow, but, no problem, when this happens, pick another car and follow it instead. Most of us use this strategy quite a bit – not necessarily for finding the way home from the airport – but when choosing a word-processing package or a desktop operating system, group-think, it can be argued with some cogency, is in the ascendancy. The strategy of following a convincing lead can be adopted for almost any complex task which requires a whole sequence of decisions, including the task of designing a network and, in particular, selecting the equipment to go into it.

However, occasionally it happens that the cars behind seem to be following you, and the car in front has disappeared from view. In this circumstance it is wise to have some independent skill in making the right choices. Hopefully the reader will develop some skill of this sort in the course of reading this book.

1.9 Design (quantities, placement, and routing)

Some design problems are very simple, however in a book like this we would be worried if all the design problems were simple.

A simple problem is one where, once the traffic to be carried on a certain piece of equipment, or network, has been identified, it is clear how many, or what quantities, or each type of equipment should be installed. A more complex problem is one where it is necessary to think clearly and for a longish time, and perhaps understand an interesting theory, in order to know with reasonable accuracy the right quantity of each type of equipment.

There are some of these interesting problems to be solved in real networks, and some of these problems, as well as the simple problems, will be considered in Chapter 9, and to a lesser extent in other chapters as well.

Factors which influence design choices heavily include: rapid growth, uncertain traffic and traffic growth estimates, rapidly changing technology, decreasing costs, introduction of new services, and rapidly changing economic conditions. These factors tend to reduce the importance of highly accurate, detailed, and optimized design, and to increase the importance of strategic considerations.

Furthermore, there *are* many decision problems in the planning and management of networks which really are simple. But, we need to make sure that we have identified these problems correctly. And when a collection of interconnected decision problems have to be made all roughly at the same time, many of them apparantly of this "simple" type, it is important to be completely confident that the simple problems have been correctly identified and the correct solution identified.

Example 1.1. A Network where cost depends upon traffic

Suppose we need to create a network to connect a collection of sites, as depicted in Figure 1.3. The communication traffic is flowing from every site to every other site. Since we have not studied traffic very much as yet, let us imagine that this communication traffic is like water travelling along a pipe, and what we need to do to provide a satisfactory service is to provide a pipe of the necessary capacity.

The traffic which we need to carry on this network is listed in Table 1.1. We shall assume that the traffic is all *bidirectional* and that the links in the network are all capable of carrying bidirectional traffic. Hence, there is no need to specify the traffic in both directions and the traffic matrix contains entries only in the upper triangle. Also, we have not included traffic which has source and destination equal to the same node because such traffic does not require any network resources.

Now let us make an assumption concerning the cost of the communication links. Let us assume that the cost of these links is proportional to the amount of traffic carried. This is often true, for example, of telecommunication

services obtained from telecommunication companies. To be specific, let us suppose that the cost of a link which carries one unit of traffic is $1 per day. Under these circumstances, even assuming that it is possible to carry traffic via one or more intermediate points, which is normally the case, the cheapest network will be the one in which every site is connected to every other site, as shown in Figure 1.4.

The required capacities of the links in this network are precisely the same as the capacities specified in Table 1.1. The total cost will therefore be simply the sum of the numbers in this matrix, in dollars, per day. There cannot be any cheaper network than this if link costs are proportional to traffic capacity.

□

Figure 1.3: A Network of Traffic Demands

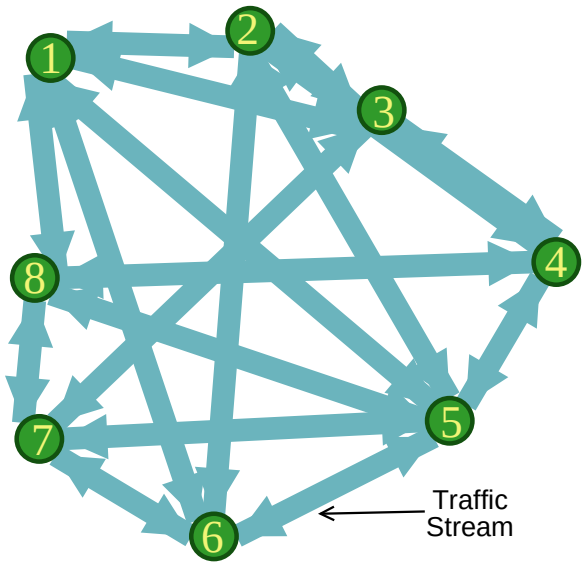


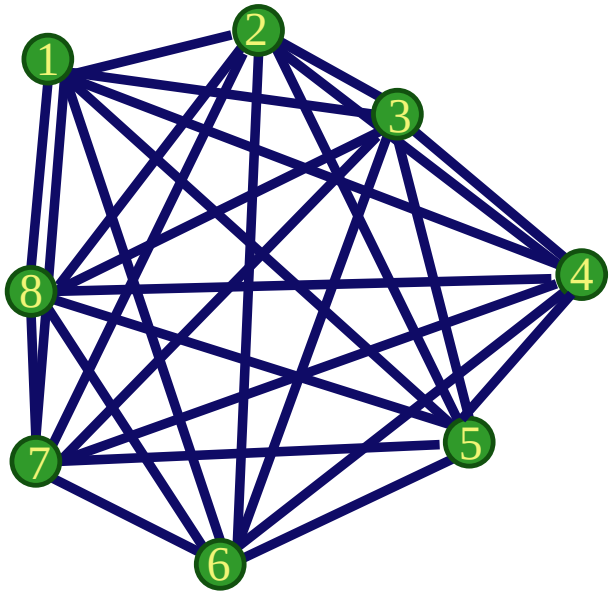
Table 1.1: Traffic Levels in a Network of 8 nodes

Node	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	·	6	3.5	9	5	8	7	6
2	·	·	7.5	9.3	25	6.2	3.2	4
3	·	·	·	3	12	4	2.5	4.4
4	·	·	·	·	6.5	12	3	12
5	·	·	·	·	·	7.3	7.8	3.8
6	·	·	·	·	·	·	7	8
7	·	·	·	·	·	·	·	4
8	·	·	·	·	·	·	·	·

Example 1.2. A Network where cost depends distance

Now let us reconsider the same example under a different assumption, one which is much more realistic in many cases: let us assume, instead, that the cost of links is now proportional to distance and completely independent of

Figure 1.4: A Solution when link cost depends on traffic level



the capacity of the link. In this case, we need the network to be connected and to have minimum total length. The cheapest network turns out to be the *minimal spanning tree*, (See Section 9.1.1 for an explanation of this problem and the algorithms for its solution) as depicted in Figure 1.5. The required capacities of the links are not, apparantly, very important, since the cost of links is not dependent upon their capacity. It is not difficult (although it is a little tedious) to work out the required capacities. This can be done by adding up the traffic levels of all traffic which passes through a link, one by one for each link. For example, the link between nodes 3 and 2 must carry all the traffic from node 3 to anywhere else, hence the capacity of this link must be 3.5+7.5+3+12+4+2.5+4.4=36.9. Calculations of the capacities of the other links are similar, although a little more complicated. All the capacities obtained in this way are shown in Figure 1.5. □

These examples should indicate that the optimal design of a network can vary greatly depending on circumstances. Although these two examples may seem extreme, there are many situations which are not unlike the first example, and also many which are not unlike the second! The difficult examples are those which fall in between these two extremes. However, at first glance, it may be difficult to see whether a problem is similar to Example 1.1 or similar to Example 1.2, or not similar to either. An example of this sort (i.e. it is unclear at first . . .) is provided in Exercise 1.1.

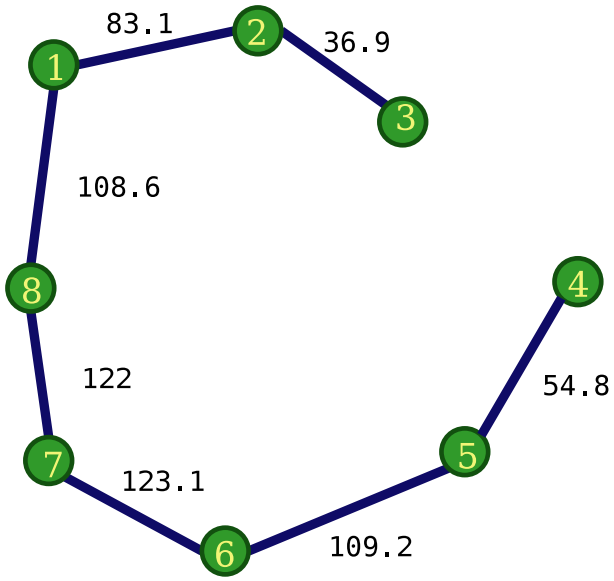
It is interesting to notice that in both these examples, identifying the optimal (cheapest) network was not difficult. This will not always be the case in subsequent examples. However, the multitude of complications which arise in real networks do not always make the task of design harder, as we shall see in Chapter 9.

In real networks, traffic is not really like water and cannot be satisfactorily carried on a link with a capacity just as large as the traffic. In fact, traffic varies randomly from day to day and from moment to moment and as a consequence we need to supply link capacities which are significantly higher than the traffic which seeks passage.

Also, the equipment and the maintenance of the equipment out of which networks are made are not perfect. As a consequence some degree of redundancy and excess capacity is normally provided in networks to allow for the possibility that some components are not functional. In the next chapter we will focus on the ways in which reliability, or the lack of reliability can be taken into account in analysing and designing networks.

Exercise 1.1. Network Design – a Difficult Case?

Figure 1.5: A Solution when link costs depends only on length. The labels indicate the capacity required.



Now consider the last example, and the one before that, with yet another cost model. This time, let us assume that the cost of each link is a mixture of fixed cost per link and traffic dependent cost:

$$\text{Link Cost} = 1 + \lfloor \text{carried traffic}/100 \rfloor,$$

where $\lfloor x \rfloor$ denotes the greatest integer which is less than or equal to x. The total cost of the network, as in the previous examples, is the sum of the cost of the links.

Hint: Start with the solution obtained in Example 1.2 and make improvements. □

References

[1] Niovi Pavlidou, A. J. Han Vinck, Javad Yazdani, and Bahram Honary. Power line communications: State of the art and future trends. *IEEE Communications Magazine*, 41(4):34–40, 2003.

[2] Cory D. Kidd, Robert Orr, Gregory D. Abowd, Christopher G. Atkeson, Irfan A. Essa, Blair MacIntyre, Elizabeth Mynatt, Thad E. Starner, and Wendy Newstetter. The aware home: A living laboratory for ubiquitous computing research. Technical report, Georgia Institute of Technology, 1999.

[3] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, 2:1–15, 1994.

Chapter 2

Reliability

In this chapter we shall deal with reliability in all its aspects: requirements analysis, performance analysis, measurement, control, architecture, equipment choice, and design. We shall consider how to set reliability goals for an organization, how to set up a procedure for measuring the reliability of a network, how to document the reliability goals and expectations for a new network, or an existing network under re-development, and how to choose the architecture, equipment, and topology of a network in order to meet prescribed reliability standards.

2.1 Introduction

The components which make up networks fail from time to time. The staff that look after networks sometimes fail in the approach they take to maintenance of networks. The users of networks sometimes behave in quite unexpected ways. And natural disasters sometimes conspire to defeat the best efforts of technology and the personel who look after it. All this is unavoidable. As a consequence, from time to time, networks fail to provide the services they are designed for.

Reliability is an issue in all networks, from the smallest to the largest. However, reliability tends to be more important in larger networks, and especially important in networks on which other networks depend.

Example 2.1. Signalling Networks

Telephone networks are made of two basic components: transmission links and switches. The transmission links were originally made up of individual pairs of wires on which an electrical signal was transmitted. The first switches were large patch boards which were manipulated by humans. Next, came switches in which the dominant switching component was a *Uniselector*, a device which rotates around as the clicks come down the line, and thereby selects a line to go out on, from the *telephone exchange*. Next came *crossbar* switches, the exciting idea of which was that the electrically controlled "cross-bars" could establish a path through the switch which stayed *lashed up*, once the control signal to the two bars were released.

The next step was the introduction of computer controlled telephone exchanges, then digital telephone exchanges (which were, of course, computer controlled).

At the same time that this revolution in switching and signalling was taking place, driven by digital technology in general and VLSI in particular, transmission technology was *also* undergoing a series of major changes. Transmission over individual pairs of wires was first replaced by coaxial cables which could carry many telephone signals at once, mixed together by shifting signals from one frequency band to another, next by digital transmission systems which allowed many telephone signals to be carried on the one pair of wires by interleaving bits, and more recently by optical fibres in which many more digital telephone signals could be carried on the one system.

Now it is possible to carry 1.6 Tbit/s on one off-the-shelf optical fiber transmission system. One such system is capable of carrying 250 million simultaneous telephone calls. For not the first time, in the history of telecommunications, the problem facing us is not how to cater for demand, but how to find enough demand to make use of the available capacity.

Originally, every telephone call needed a separate *pair* of wires, or, whenever amplification was required (which includes all long distance calls), *two* pairs of wires. The reason that communication requires a *pair* of wires is that

the signal is usually represented by the *voltage between* one wire and the other. The reason *two* pairs of wires are required for the long haul is that in this case each direction of transmission can have its own separate pair. This makes it much easier to introduce signal amplification.

At some point it was realized that, rather than every incoming line to a telephone exchange being capable of, and willing, to send signals (such as a request to set up a call or a request to terminate a call) to every neighbouring telephone exchanges, it might be easier if all telephone exchanges exchanged messages via a *network* set up specifically for signalling.

With the arrival of digital transmission, a single pair of wires could be used to carry 24 digital channels, each with the capacity of 64 kbit/s, sufficient for satisfactory voice communication.This rate of 64 kbit/s is generated from a audio signal when it is sampled at the rate of 8,000 times per second, each sample in the form of an 8 bit number (a byte, also known as an octet in documents from the ITU). A pair of these systems could be used to provide 24 two way channels of communication for use in telephone networks. This was the type of multiplexed digital telephony oriented system originally introduced in the United States and Japan and it was known as a T1 system. The term T1 is still used in the U.S. to denote a communication facility capable of two way transmission of 24 voice channels [1]. In Europe and Australia, a different standard was established, implemented and installed on a widespread basis. Now known as E1, this system is capable of carrying 32 simultaneous voice channels in both directions [2].

Both the T1 and the E1 systems are never used purely to carry digitized voice. By necessity, part of the raw capacity of the transmission system is allocated to two very important *administrative* functions: *framing* and *signalling*. In the original T1 system, the bits for providing framing and signalling were *stolen* from one of the 64 kbit/s voice channels, whereas in the E1 system, one of the 32 64 kilobit/s channels is used for framing and another is used for signalling. The purpose of framing is to coordinate the interpretation of each channel in the T1 or E1 system, i.e. so that each end agrees which incoming channel is allocated to which channel on the E1 system. Signalling, on the other hand, is used to transmit control information such as: “I have a new voice call to 31 9256 coming on channel 22. Please try to connect it through your switch and let me know when its done, or if you can’t make that connection, let me know that you can’t make the connection.”

This brings us back to the subject of signalling. In principle, every telephone exchange (a switch for telephone calls) needs to communicate *signals* (such as “I have a call to 31 9256 . . . ” or “the call to 31 9256 is terminating”) to every other telephone exchange. In a large metropolitan area, with a population of several million people, there could be in excess of 50 telephone exchanges (at least, this was the situation some time ago when this issue of signalling first arose – nowadays telephone exchanges tend to be larger, and so there don’t need to be quite so many), so the number of signalling paths between these telephone exchanges might also need to be very large – up to 50×50 perhaps. Actually, not all pairs of telephone exchanges have a direct connection, one to the other, so the number of signalling paths might not need to be quite this large – nevertheless, the number required will, unless we do something “smart”, be quite large. An illustration of how this network of signalling paths might look is shown in Figure 2.1.

For these reasons, telephone companies around the world, individually and together, developed a network protocol and architecture for signalling communication between telephone exchanges.

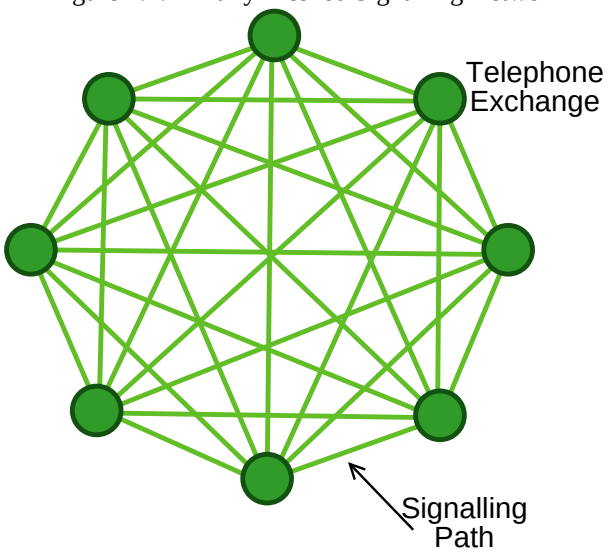
If this work was done today, this protocol and network would probably be based on the protocols of the Internet (TCP/IP), not because these protocols are inherently superior to anything else, but because they exist already, and because equipment and software implementing them is available cheaply.

However, instead of building on an existing data communication architecture, the standards bodies and telecommunication organizations developed a network architecture known as *Signalling System Number 7*. One of the ideas of this protocol was that there might exist special nodes in the network whose role was nothing except transferring signalling messages from one telephone exchange to another. These transit points were known as Signalling Transfer Points, or STP’s. A network making use of these is depicted in Figure 2.2.

This new architecture has some cost and reliability advantages. In particular, each telephone exchange only has to be connected to two STP’s. In fact, one STP would be sufficient, but at least two are used for reliability reasons.

In the old architecture, assuming that telephone exchanges are not able to act as transit points for the signalling network, once a signalling connection is disabled, it becomes impossible for calls to be set up to that location. Mind you, this might not be an issue if the reason for the lack of communication has also caused the transmission path for the calls to become unavailable as well. Anyway, the new architecture ensures that no single transmission failure or STP failure can break communication between one telephone exchange and all the others.

Figure 2.1: A Fully Meshed Signalling Network



However, if a failure does occur in this signalling network, the consequences could be dramatic.

In fact, a dramatic failure of a signalling network occurred in a very widespread area of the United States east coast approximately 20 years ago [3]. The failure was complex and difficult to diagnose, continued for almost a whole day, and affected millions of people in some of the most densely populated areas of the United States – the north-east coastal region. It manifested itself as heavy signalling traffic leading to overloaded STP's, which auto-rebooted as a recovery mechanism, and then propagated the problem by overloading their STP neighbours as part of their restart behaviour. The problem was eventually tracked down to a C program in which a continue statement had been placed where there should have been a break statement, or conversely.

This example illustrates an interesting example of a *single point of failure*, the software. Because all the STP's were using the *same software*, an error in this one component was able to manifest itself as a catastrophic source of failure for an entire network.

Because of this historical incident, it was decided in the case of the Australian Signalling System Number 7 network, that software from two independent sources would be used in the STP's [4].

□

2.1.1 Terminology

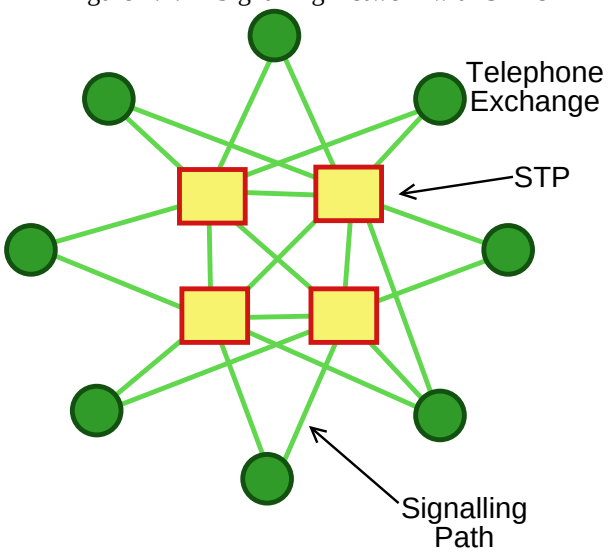
Before considering how the reliability of networks can be improved we should define a few important terms. An individual component which is not perfectly reliable will pass through a succession of states. From the point of view of reliability, in the present context, such a component is either working (*up*), or not working (*down*). The average time between up periods is known as the *mean time between failures* (*mtbf*) . This period should be considered to start at the start of one up period, and finish at the start of the next up period. The average length of a down period is known as the *mean time to repair* (*mttr*). This period should be measured from the start of a down period to the end of a down period.

The frequency of down periods, measured in incidents per unit time, is simply the inverse of *mtbf*. For example, if a failure occurs once every four months, on average (a period of 0.33 years), the average frequency of occurrence is 3 times per year.

Another way to measure reliability is by the *average proportion of time during which the component is up*. This is known as the *availability* of the component. We have the following simple formula relating these three quantities:

$$\text{availability} = \frac{\text{mtbf} - \text{mttr}}{\text{mtbf}}.$$

Figure 2.2: A Signalling Network with STP's



Availability is here expressed as a measure of the proportion of time during which an element, or a system, or a service, is operational. However, it can also be considered as a probability – the probability that the element, system, or service is operational. This equivalence is valid so long as the system under study is *ergodic*, which is almost always a reasonable assumption. A system is termed *ergodic* if the average behaviour of a large class of systems is the same as the average behaviour of one system observed over a long time.

Exercise 2.1. Availability expressed in minutes per year

Calculate how many hours, minutes, and seconds per year a system would be down, on average, if it had an availability of 99.5%.

Suppose an availability of 2 minutes down time per year was required for a certain service. What does this correspond to when expressed as a probability?

What is the availability of a system whose down time over the last twelve months has been, in minutes in each month, from January to December: 12, 8, 35, 2, 2, 24, 11, 18, 5, 22, 12, 20. □

We might need to make a distinction between cases where a service is fully operational, partially operational, and totally unavailable. This leads to two possible definitions of availability. We could say that a service is *up* if it is *partially up*, or we could insist that the service is *fully operational* before we declare it to be *up*. The most useful definition seems to be the one where we call a service *up* so long as *some* functional service is available. If necessary (which it will be, in some situations), the definition shall be made more precise in the context of the discussion.

2.2 Analysis of Network Reliability

When a component in a network fails, it does not necessarily prevent communication between a certain pair of nodes in the network. Consider the nodes A and B in the network depicted in Figure 2.12. In one fairly reasonable scenario, so long as there is *a path* through the network which avoids any of the failed components, communication between A and B will continue. Because there are two paths between A and B, in this network, it will require two failures to prevent communication between A and B.

When we analyze availability of networks, it is normal to assume that the failure events of separate components are statistically independent. This simplifies the analysis of network availability considerably. We now need to recall two simple laws of probability for calculating probabilities of compound events. Suppose U and V are two statistically independent events. Then

$$P\{U \cap V\} = P\{U\} \times P\{V\} \tag{2.1}$$

and

$$P\{U \cup V\} = P\{U\} + P\{V\} - P\{U\} \times P\{V\} \approx P\{U\} + P\{V\}, \tag{2.2}$$

in which $U \cup V$ denotes the *union* of the events U and V, i.e. the event in which either U happens or V happens, and $U \cap V$ denotes the *intersection* of events U and V, i.e. the event in which both U and V happen.

Events such as U and V can be thought of as *sets*. According to this view, there is a universe, $\mathcal{U}$ say, of possible *outcomes*. Each outcome is a complete enumeration of the state of the world, i.e. all the details which might be of interest are completely specific in each outcome. An event is a set of possible outcomes. This is why the event in which both event U and event V happen at the same time is thought of as the *intersection* of events U and V, $U \cap V$ and the event in which *either* event U or event V happen is thought of as the *union* of events U and V, $U \cup V$.

This way of looking at events is not particularly important in the study of reliability. It is mainly important just to understand why we talk of unions and intersections of events. However, when we come to Chapter 3, this way of looking at events becomes much more important. If it seems a little technical or fussy, please allow a little lattitude to the technical writer. This way of looking at events has a long and successful history in the study of probability and statistics and adopting this viewpoint will reward the reader amply, if not immediately, a little further along the path in their study of networks.

Approximation (2.2) applies when the probabilities of U and V are small. In particular, if these represent *failure* events, for example if U is the failure of transmission between A and B and V is the failure of transmission between A and C, both probabilities, $P\{U\}$ and $P\{V\}$, will be quite small and so the probability $P\{U \cap V\} = P\{U\} \times P\{V\}$ will be *very* small, so small that it is reasonable to neglect it.

2.2.1 An Enumeration Algorithm

Network availability can be computed to any desired accuracy by the method described in [5], which reduces to an enumeration of all states, terminating when the sum of the probabilities of states considered is sufficiently close to 1. Suppose we wish to determine the availabilities with error in probability less than ε. This algorithm works as follows:

Set $\text{cumprob} = 0$. Enumerate all possible up/down states of the network, stopping when $\text{cumprob} \geq 1 - \varepsilon$. While doing so, accumulate the following quantities:

(i) the cumulative sum of the probabilities of the enumerated states - call this cumprob;

(ii) for each O-D pair, (O,D), the cumulative sum of the probabilities of states in which communication between the the nominated origin and destination is possible - call this $\text{cumprob}(O,D)$.

Example 2.2. Unavailability Calculation Using the Enumeration Algorithm

Let us estimate the unavailability of the origin-destination pair *A*-*B* in Figure 2.5.

In an enumeration of all possible states in this network, we naturally start with the state where all links are up. This has probability $0.99^4 \times 0.999^6 = 0.9606 \times 0.994 = 0.955$. Next, we consider all the states when one link is down and all the other links are up. The total probability of all these states is $4 \times 0.01 \times 0.99^3 \times 0.999^6 + 6 \times 0.001 \times 0.99^4 \times 0.999^5 = 0.0443145$. The total probability accounted for by the states where at most one link is down is 0.9991613, which might be sufficient to achieve the desired accuracy, depending on the accuracy which is desired. Since the network remains fully connected in all of these states, we can see already that the availability of all paths is at least 0.9991613. It is straightforward to enter these calculations into a spreadsheet.

The cumulative probability of the states where no more than two links are down turns out to be 0.9999919, so considering only states where at most two links are down will be sufficient to obtain a good approximation of the availability of any origin-destination pair in this network. Now, all we need to do is to identify which states, with two links down, cause the path from *A* to *D* to be down, in order to estimate the availability of this origin-destination communication path.

A little thought reveals that in order for this origin-destination pair to be disconnected, the link from *A* to *E* must be down, *or* the link from B to C must be down together with one of the links making up the upper path from *A* to *C*. The sum of the probability of these combinations is $2 \times 0.001 \times (3 \times 0.01 \times 0.99^3 \times 0.999^5 +$

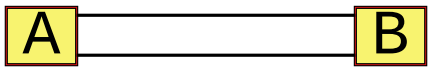
$0.001 \times 0.99^4 \times 0.999^6) = 0.00005984$, which is a lower bound on the probability that this origin-destination pair will be disconnected. An upper bound is this number plus the sum of the probabilities of the states which have not been considered (i.e. the ones with 3 or more links down), which comes to $0.00005984 + 0.000008089 = 0.00000.00006793$. So the unavailability of this origin-destination pair is between 0.00006 and 0.00007. □

2.2.2 Another Algorithm

Another more pragmatic approach to calculating network availability usually works well in practice. In some cases, this second approach needs to be combined with the first approach to solve a problem quickly and effectively.

This second approach is basically a direct application of the rules (2.1) and (2.2). If a network contains, for example, two links in parallel, between nodes *A* and *B*, as in Figure 2.3 the combined link from *A* to *B* is up whenever either the first link is up *or* the second link is up. Therefore, we can calculate the reliability of a single link which could replace this pair of links without changing the reliability of any connections across this network by means of rule (2.2).

Figure 2.3: A network with parallel links



Similarly, if a network contains two links in series, between nodes *A* and *B* via node *C*, for example, the combined link from *A* to *B* will be up whenever both the first link *and* the second link is up, so we can replace this pair of links by a single *equivalent* link by using rule (2.1). By repeating this procedure as many times as possible we can often reduce a network to a level where the availability calculations are trivial. If necessary, we can apply the previous method to a simplified network.

Figure 2.4: A network with serial links

Quite often it is easier to work in *unavailabilities* rather than availabilities. The mathematics is much the same, but the actual calculations are often easier.

If a network contains two links in parallel, between nodes *A* and *B*, the combined link from *A* to *B* is *down* whenever *both* the first link is down *and* the second link is down. Hence the *unavailability* of the combined link is the *product* of the unavailabilities of the individual links.

Similarly, if a network contains two links in series, the combined link is *down* whenever one or the other link is down, and so the unavailability of the combined link is (approximately) the sum of the unavailabilities of the individual links. This is the first application of the second equation in (2.2) that we have considered. It is quite an important application, however, because links in series occur quite often, and adding the unavailabilities of the individual links to obtain the unavailability of the combined link will usually be justified and much more convenient than the more precise method of multiplying availabilities to obtain the availability of the combined link.

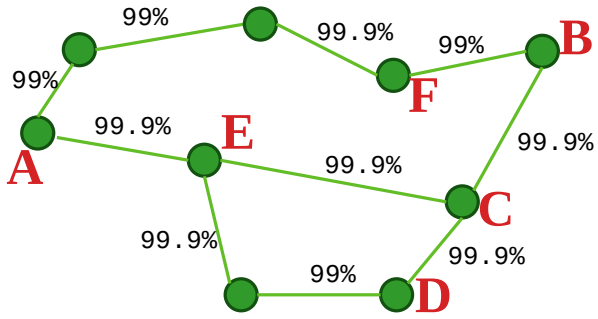
Example 2.3. The Parallel Serial Reduction Method

Let us apply this approach to the network depicted in Figure 2.5 with a view to re-calculating the availability of the path from *A* to *B* which was calculated previously in Example 2.2.

This network has a series of links in series across the top, from *A* to *B*. This series of links is equivalent to a single link with a certain reliability, *x*. What value must we choose for *x*? Here is where we use rule (2.2). The series of links from *A* to *B* will be *down* if any one of the links in the path is down. So the *down* event has probability, approximately,

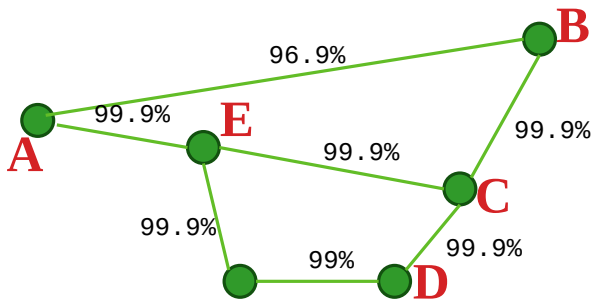
$$x = 0.01 + 0.001 + 0.01_+0.01 = 0.031.$$

Figure 2.5: A Network with unreliable links



Therefore, an equivalent network, from the point of view of the availability from A to B is depicted in Figure 2.6. If we carry out the calculation of the availability of this upper path more precisely, by multiplying the availabilities rather than by adding unavailabilities, the unavailability turns out to be 0.030671299. We shall carry these more precise calculations throught this example so that the degree of inaccuracy introduced by the simpler formula, i.e. using the approximation at (2.2), is clear.

Figure 2.6: An equivalent Network



Next, let's find a link equivalent to the little network which joins E to C. Two steps are required here. First, let us replace the lower path with a single link with probability of being down

$$x_1 = 0.001 + 0.01 + 0.001 = 0.012.$$

This is the appropriate availability because this path is *down* if and only if one (or more) of the links in this path is down, so we can apply (2.2) and obtain the formula for x_1 just given. The more precise calculation gives the answer 0.011979010.

Next, we are left with two links in parallel between E and C, the upper one with availability 99.9% and the lower with availability $1 - 0.012 = 98.8\%$. To combine these two links together we can use rule (2.1), because in order for the combined path to be *down* it is necessary that *both* links be *down*. So the equivalent link between E and C has down probability

$$x_2 = 0.012 \times 0.001 = 0.000012.$$

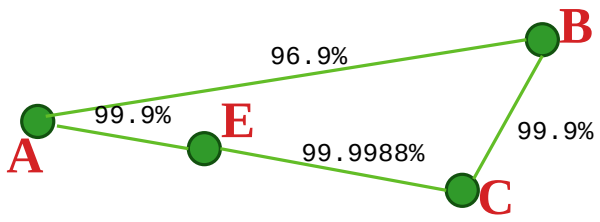
and therefore availability 1-0.000012 = 99.9988%.

The more precise calculation gives the answer 1-0.0000119790 = 99.99880210%. The equivalent network is depicted in Figure 2.7. The resulting network is now made up of two parallel paths. The bottom path is equivalent to a link with down probability

$$x_3 = 0.001 + 0.000012 + 0.001 = 0.002012.$$

and the upper path (a link) has down probability 0.031. A more precise calculation of the availability of the lower path gives the result that the availability is 0.997989045 = 1-0.002010955.

Figure 2.7: An equivalent Network



Because these links are in parallel, we use rule (2.1) to calculate the down probability of the next equivalent link, which turns out to be

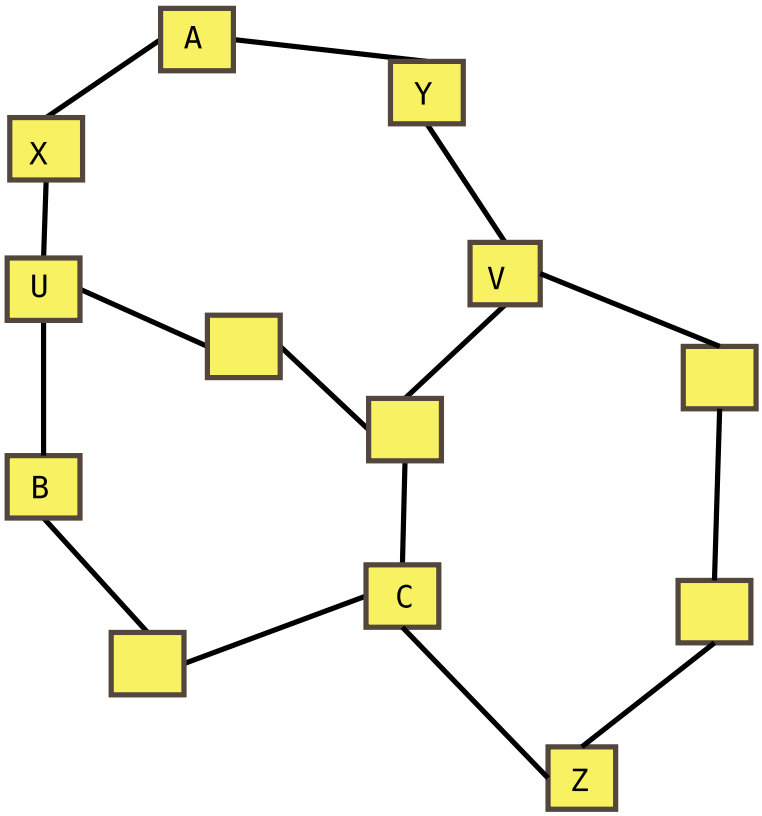
$$x_4 = 0.002012 \times 0.031 \approx 0.000062.$$

The more precise calculation gives the result $0.002010955 \times 0.030671299 = 0.000061679$. So the answer we sought is this, that the availability of the network from *A* to *B* is 1 - 0.000062 = 99.994%, or, by the more precise calculations, 99.9938321%. It seems that the approximate calculations are really quite accurate, and there is no doubt that they are a lot easier. □

Example 2.4. The method of the perfect middle

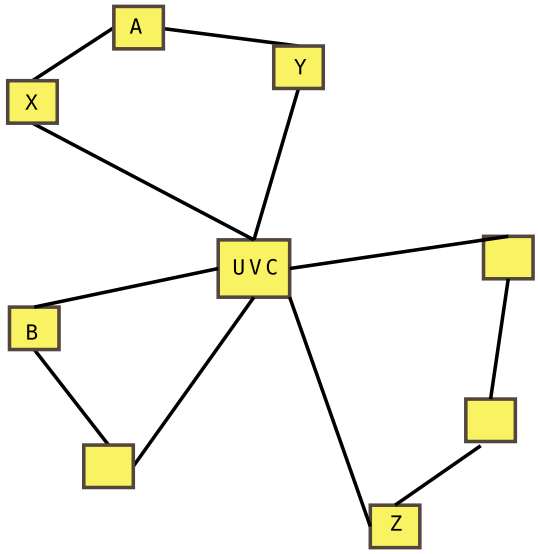
In a lot of examples, a very simple approach proves to be quite accurate and much easier to carry out. This method is well decribed as the *method of the perfect middle*, since it is based on the idea that in many networks, the core, or middle of the network, behaves as if it is perfectly reliable.

Figure 2.8: A Network of several rings



Consider the network depicted in Figure 2.8, in which each link has availability 99.8%. We would like to calculate the availability of communication between X, Y and Z. Observe that the availability of paths from U to V, from U to C and from V to C is very high. This is because there are *three disjoint paths* between each of these pairs of nodes. In the method of the perfect middle, we simply assume that these nodes have perfectly reliable inter-communication, and hence from the point of view of the rest of the network, it is as if these nodes were coalesced into one node. The resulting network is shown in Figure 2.9.

Figure 2.9: A Network with a perfect middle



This network can now be easily simplified further by the parallel-serial reduction method, after which it becomes equivalent, from the point of view of communication between X, Y and Z, to the network shown in Figure 2.10.

We can now calculate the availability of the two independent ways for X to communicate with Z as 0.96 and 0.94, and hence communication between X and Z has unavailability $0.04 \times 0.06 = 0.0024$. Similarly, the two ways for X and UVC to communicate have availability 0.96 and 0.94, so the path from X to UVC has unavailability $0.04 \times 0.06 = 0.0024$ also. By roughly the same argument the path from Y to UVC has unavailability $0.02 \times 0.08 = 0.0016$. The unavailability of the path from UVC to Z, on the other hand, has unavailability $0.02 \times 0.06 = 0.0012$.

We can now find the unavailability of the path from X to Z by adding the unavailability of X to UVC to that of UVC to Z, to obtain an overall figure of 0.0036 and the unavailability of Y to Z can be obtained by adding the unavailability of Y to UVC, 0.0016, to that of UVC to Z, 0.0012, to obtain 0.0028.

□

Example 2.1. A Signalling Network (continued)

Let us now analyze the network depicted in Figure 2.2, to work out its availability. Let us suppose that the availability of the links is a_l and the availability of the STP's is a_s. The individual telephone exchanges will be supposed to have perfect availability because there is no need for the signalling system to be functional unless the telephone exchange itself is working.

The first simplifying observation is that we might as well assume perfect availability of the network between the STP's. There are three disjoint ways for a signal to pass from one STP to another – on the direct path, or via either of the other two STP's. Hence, the availability of the path from one STP to another is

$$1 - (1 - a_l)(3 - 2a_l - a_s)^2 \approx 1.$$

This implies that the STP network can be replaced by a single node with availability a_s, without altering the availability performance of the signalling network at all. This situation is depicted in Figure 2.11.

Figure 2.10: A Network further simplified

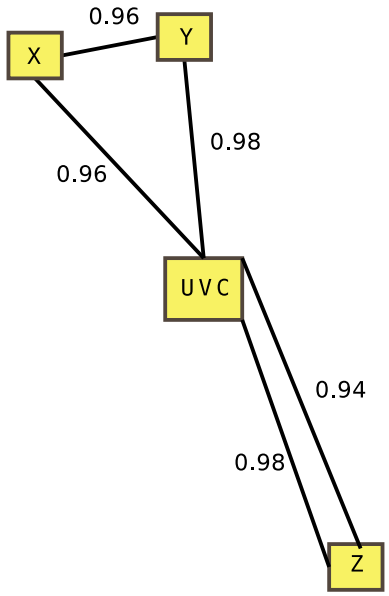
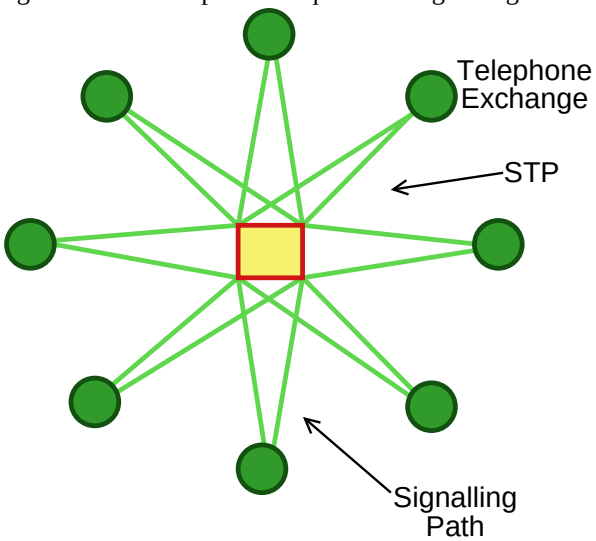


Figure 2.11: A Simpler but Equivalent Signalling Network



The availability of an end-to-end path from a telephone exchange to *anywhere* on the STP network (i.e. to the single node which we substitute for it) is

$$1-(2-a_l-a_s)^2.$$

We arrive at this number by observing that in order to reach the signalling network, we have *two* options, precisely, and each of these options will work only if both the link and the STP are up. Hence, each option has *unavailability* $1-a_l+1-a_s=2-a_l-a_s$. The unavailability of the link which is equivalent to these two parallel paths therefore has unavailability $(2-a_l-a_s)^2$. The availability of this equivalent path to the STP network is also very close to 1, although not quite so close as that of the path from one STP to another, so this time we will not assume that this availability is close enough to 1.

The end-to-end path, exchange to exchange, requires that we get to the STP network from a telephone exchange, and then go from the STP network to the destination exchange. In at least some cases (the worst cases), the STP we arrive at and the one we leave from must be different. This end-to-end path therefore has availability (in this worst case)

$$1-2(2-a_l-a_s)^2.$$

For example, if the switches and links have availability 99.5 %, the availability of the end-to-end path will be

$$1-2\times 0.0001=99.98\%.$$

There are $365\times 24\times 60=525,600$ minutes in a year, so the expected down-time in minutes of the signalling network in this situation will be 105 minutes, or an hour and twenty five minutes.

It is quite likely that this would be considered *unsatisfactory* for a signalling network. A simple way to improve the availability would be to connect each exchange to at least *three* STP's. The calculation method just used is still applicable (with some provisos), and we can conclude that the availability of this design would then be

$$1-2(2-a_l-a_s)^3=1-2\times 0.000001=99.9998\%, \tag{2.3}$$

so the expected down time per year would now be 1 minute per year. This is the sort of target which is set for a signalling network. Since the additional cost of adding the extra link is quite small anyway, the choice to add the extra link would be almost certainly be made.

In this case, where each exchange is connected to *three* STP's, there wouldn't actually be any *worst cases*, where the origin exchange and destination exchange are not connected to any common STP's. However, (2.3) is still a lower bound on the true availability and this calculation is still valid for a worst case which *would* exist if the network was much larger, e.g. if there were 6 or more STP's.

It should be noted that the decision to use a special network architecture for the signalling network is based strongly on historical issues, and might not be made again today, although it is probably still true that a large telephone network would need its signalling network to be totally separate from any public networks. □

Exercise 2.2. Analyze the reliability of a network

Consider the network depicted in Figure 2.12. Assume that each link in this network has availability 99.5%. Calculate the end-to-end *availability* of the connections from A to B, A to C and B to C. □

Exercise 2.3. Another network availability problem

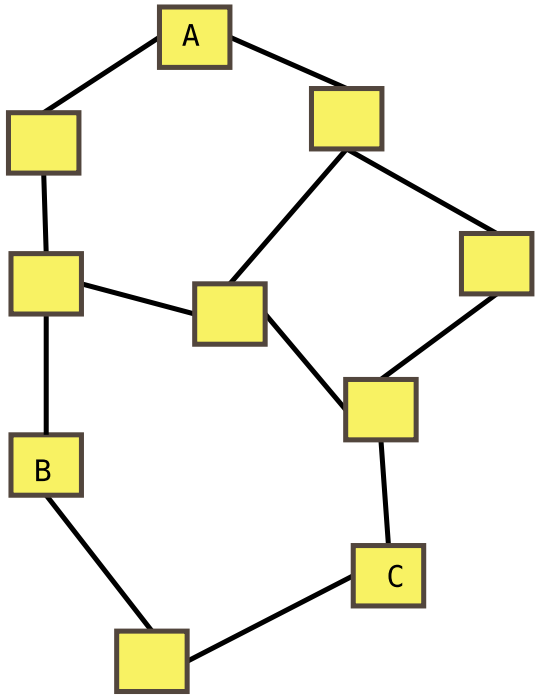
Consider the network depicted in Figure 2.5, but now consider the case where an additional link from E to F has been added. Suppose that this link has availability 0.99. What is the availability of the path from A to B now? □

Exercise 2.4. Analyze the reliability of another network

Consider the network depicted in Figure 2.8. Assume that each link in this network has availability 99.8%. Calculate the end-to-end *availability* of the connections from A to B, A to C and B to C.

□

Figure 2.12: A Network with loops



2.3 Network Architecture

2.3.1 Layering

By far the most important concept in network architecture is layering. Layering of networks is of fundamental importance in their analysis, design and architecture. Decisions about layering of networks dominate the whole issue of architecture.

How can layering of networks impact on reliability?

A simple way in which layering can be used to enhance any performance issue is to insert a layer with the specific role of enhancing this performance measure. We could, in theory, have a network layer which enhances (reduces) delay, a layer which reduces loss, or a layer which increases reliability. Alternatively, we could look for a layer which *reduces cost*!

It is not necessarily possible to do all this with layers, and sometimes a good way to reduce the cost of providing a network might turn out to be eliminating a layer!

> *When a layer is good, it is very very good, and when it is bad it is costly.*

The concept of layering deserves careful study, without assuming that a special layer is *always* a good idea.

The point of view that a network is made up of layers might sometimes be more of an *interpretation*, than a well-established sub-division of functions. This *way of looking* at networks is, however, useful and appropriate, and it is highly recommended to the reader to develop a keen eye for this point of view.

For example, the lowest layer in most communication networks is usually considered to be the *physical layer*, which is made up of the optical fibers, cables of twisted pairs, microwave towers, and so on. This lowest layer does not provide a useful service in its own right because these physical transmission media need transmission devices to be added at each end in order for communication to be even possible. Maybe we should consider the transmission equipment to be part of the physical layer. But even then, the physical layer only provides point-to-point communication. It might even be useful to consider a layer *below* the physical layer – the duct layer. This layer is made up of the pipes, ducts, and pits in which the physical transmission facilities are stored, or buried.

2.3.2 Definition of Layering

With all this talk of layering, it is high time that we defined the concept.

Definition 2.1 *A* network layer *is a collection of transmission and/or switching equipment which provides a collection of* communication services*, possibly with the assistance of a (single)* sublayer.

Definition 2.2 *A* sublayer *is a network layer which provides services to another layer.*

Note: all except the bottom layer of a collection of network layers making up a network *do* make use of a sublayer. In this way, the layers which make up a network are strictly ordered, from bottom to top. Each layer provides services to the layer immediately above and makes use of services provided by the layer immediately below.

Definition 2.3 *A* communication service *is a facility which enables communication between two remote locations.*

Example 2.5. A Connection-oriented Packet Layer

Suppose a network layer provides a connection-oriented packet communication service for connecting any two nodes to which it is adjoined. Such a network layer, at the very minimum must provide the services:

1. Setup connection *A* to *B* (return connection number, *n*);
2. Transfer packet *m* from *A* to *B* on connection *n*;
3. Clear down connection *n*.

□

2.3.3 A Transmission Facility Network

A layer for providing better reliability is normally provided quite explicitly in the telecommunication infrastructure. The layer below is made up of optical fibers and the the optical fiber transmission equipment. The layer above is made up of switches, e.g. telephone exchanges, which connect services to the transmission resources they need.

It is usually convenient to combine the provision of reliability and *grooming* of transmission facilities. An optical fiber can readily be equipped, nowadays, to provide in excess of 1 Terabit/s of transmission capacity, however it is not necessarily convenient or appropriate to connect transmission facilities of this capacity directly to a switch for a higher level service. For example, telephone switches traditionally deal in modules of 24 (or 30) telephone channels, which can be accommodated in 1.4 (or 2) Mbit/s. Breaking down the large capacities provided by transmission equipment into more modest modular capacities as required by service-oriented switches is known as grooming.

2.3.4 SONET and the Synchronous Digital Hierarchy

After an initial period in which new transmission systems for optical fibers were developed in standardized manner, but without adequate attention to synchronization, a plan emerged in the late 1980s to define a *standard* for optical fiber transmission systems with the following characteristics:

(i) optical fibers should be able to be joined together, glass-to-glass, without having to ensure that the manufacturers of the equipment at the distant ends being the same;

(ii) the transmission protocol should allow for very accurate synchronization of the end-points, thereby leading to a fully synchronized network;

(iii) it should be possible to extract and insert signals at a wide variety of transmission rates without having to *demultiplex* an entire transmission system;

(iv) the standard should be open-ended with regard to the transmission speed of the optical fiber systems being defined, i.e. there should be no upper bound to the speed of systems coming under the purview of this standard.

This new standard became known as *SONET* in the United States and as *SDH* in Europe [6, 7, 8].

The first property of the new standard listed above merely ensures that the standard is sufficiently precise in definition and implementation that equipment from different manufacturers will be compatible whenever both items of equipment fully meet the standard.

The second issue implies that each *node* in the network of transmission systems will have a clock, and that all these clocks will be synchronized (to a higher degree that they would be otherwise – perfect synchronization is not possible). The purpose of this synchronization is to reduce the overhead required to take into account the remaining lack of synchronization to a minimum.

Every transmission system requires some sort of *framing* , and SDH systems are not significantly different as far as this is concerned. Framing is usually achieved by ensuring that a certain *bit pattern* is retransmitted every so often. The distance between these repetitions of this pattern is fixed in advance, except for a bit or two which might be added or removed to account for a small lack of synchronism between the two transmission endpoints.

There will always be some small degree of asynchronism. The remaining asynchronism needs to be taken into account by inserting (*stuffing*) additional bits into the bit-stream, when necessary – this is known as *bit stuffing*, or, occasionally, removing some bits from the bit-stream. Naturally there has to be a particular place, relative to the *frame* of the transmission system, where these extra bits are normally located.

There is another purpose of framing in addition to ensuring that the two end-points are synchronized. This second purpose is to ensure that the two end-points agree on the interpretation of the component bit-streams contained within the transmission system.

Individual bit-streams are *byte interleaved* in the SONET standard. Each bit-stream can potentially occupy any number of bytes within the basic frame. Within the SDH standard there is an allowance for framing to occur at several *levels*. The lowest level, and the one already referred to, is associated with synchronization of the two end-points. Within this base-level frame, there are a certain number of bytes allocated to overhead functions – i.e. these slots are reserved for supporting SDH functionality rather than carrying user data. In particular, some of these slots are used to store a *pointer* to the start of the next higher level frame. User data is stored in particular positions relative to this next higher layer frame.

The transmission speeds which are defined in the SONET/SDH standard include, potentially, any multiple of the basic SONET rate, which is 51.840 Mbit/s. The system in which the multiple is 1 is known as OC-1. In practice, not all multiples of the basic rate are manufactured because there is very little saving in cost likely to acrue from using a system with a much higher capacity than required. The rates which are flagged for use are currently: OC-1, OC-3, OC-12, OC-48, OC-192 [9] and the rate OC-768 ($\approx$ 40 Gbit/s) appears to be under development [10]. The use of *Wave Division Multiplexing (WDM)* can increase the total capacity of a single fiber well above these rates. Systems capable of transmitting 160 different wavelengths on the same fiber, each independently carrying 10 Gbit/s, giving a total capacity of 1.2 Terabits/sec are already in production [11].

A Tutorial prepared by the The International Engineering Consortium contains a pictorial representation of the framing scheme and more details of how it works.

2.3.5 Add-drop Multiplexors and Network Reconfiguration

The framing structure of SONET/SDH transmission systems facilitates much more economical switching of bit-streams from one transmission system to another than earlier transmission technologies were capable of supporting. Furthermore, the flexibility with which such systems can be provided and configured is much greater with the SONET/SDH standard. In some instances we may want to totally re-order and redirect bit-streams from one collection of transmission systems into another collection of transmission systems. This sort of situation is depicted in Figure 2.13.

Another typical configuration is depicted in Figure 2.14. In this case, the total number of bit-streams which are reconfigured is quite small. Even though the quantity of bits passing through a device of this sort might be very large, the complexity, and cost, of such a device might be quite small, largely on account of the SONET/SDH transmission framing structure.

Figure 2.13: A Digital Cross-connect

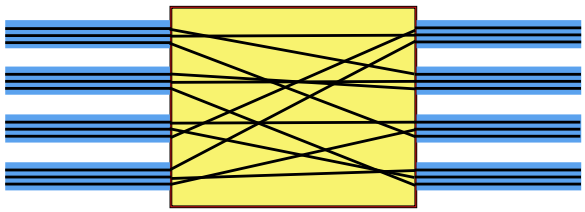
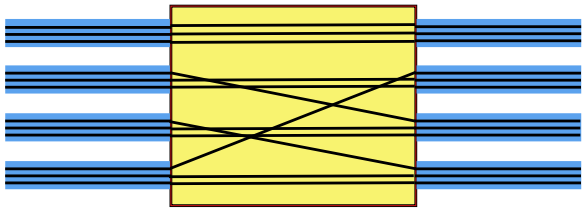


Figure 2.14: An Add-drop multiplexor



Suppose the speed of all the ports in a device of the sort in Figure 2.14 is increased by a factor of 4, e.g from OC-192 to OC-768. Assuming that ports of this speed (40 Gbit/s) can be built, the complexity of the device will increase by a relatively small amount.

As a consequence of all these factors, digital cross-connects and add-drop multiplexors are ideal as the switching facilities in the nodes of a reconfigurable network. In particular, such equipment can be used to effect the re-routing required to ensure that a network with multiple paths provides a high standard of reliability.

2.4 Design for Reliability

In this section we concentrate on reliability aspects of design, but we don't want to completely ignore traffic. In order to take *traffic levels* into account, we shall treat it like a fluid flowing through pipes. Our job is to make sure that there is sufficient capacity in the pipes to carry the flow, most of the time.

The availability standard we set for ourselves can be interpreted in some slightly different ways. Suppose the availability standard we are aiming at is 99.9%.

The first, and simplest interpretation of this standard is that for each origin, *A*, and each destination, *B*, in the network, 99.9% of the time there should be *a path* between *A* and *B*. Let us call this the *weak* interpretation of an availability standard.

The second, slightly more complex, interpretation, setting a stricter availability standard, is that 99.9% of the time there is sufficient capacity in the network for all the traffic in the network to be transported from its origin to its destination. This is the *strict* interpretation.

The weak interpretation is much easier to interpret and design to, hence, whenever a problem becomes a bit tough we shall use the weak interpretation. Also, when tackling design problems, we shall start with the weak interpretation and, if progress is sufficiently easy, then move on to the strict interpretation of availability.

The key to minimizing cost of networks designed to a reliability standard is to make use of *path diversity* (the fact that there is more than one, preferably *disjoint*, path connecting any two places) to provide a high standard of availability. Two paths are *disjoint* if they have no common links or nodes except the origin and the destination. Path diversity can be achieved without the use of very many additional links. In many cases, just one additional link is sufficient to achieve quite high levels of availability.

The minimum number of links required to connect n nodes is $n-1$. A connected network with this minimum number of links is called a *tree*.

A network is said to be *2-connected* if there are two *disjoint paths* between every pair of nodes. A good example of a 2-connected network is a *ring network*, which takes the form of a single path which passes through every node

of the network precisely once. A ring network connecting n nodes has precisely n links, which is only one greater than the number of links required to ensure connectivity. Clearly, if a ring network is adequate for connecting a collection of nodes, it is likely to be economical. Any 2-connected network will have *much* higher availability by virtue of the fact that there are two completely distinct ways to go from any place to any other.

Sometimes one ring is not enough to provide sufficient path diversity to ensure adequate availability is achieved. Suppose a ring network contains 200 nodes and all links have availability 99%. If two nodes are separated by 100 links, the probability that the first (upper) of the two paths will be up will be

$$0.99^{100} = 0.366$$

and of course this is also the probability that the other path will be up. So the probability that at least one of the paths will be up will be

$$1-(1-0.366)^2 \approx 0.6.$$

This is unlikely to be an adequate availability for a real network.

So, for large networks, more than one ring will be required in order to acheive satisfactory availability.

Exercise 2.5. Availability Calculation

Suppose two nodes, A and B in a ring containing N nodes, in which the availability of each link is a_l, are separated by n links, so that one path contains n links and the other path contains $N-n$ links. Derive a formula for the availability of the O-D pair (A,B). (Recall that an O-D pair is just an ordered pair of nodes, the first interpreted as the origin and the second as the destination). □

2.4.1 Design of a Network of Rings

The design of an optical fiber (and therefore SDH/SONET based) network can be based on the idea that any 2-connected network can be viewed as made up of rings. Use of paths outside the immediate ring to connect any two nodes is to be discouraged because use of paths longer than the minimal available length reduces traffic efficiency. So, with a view to developing some simple but effective design guidelines, suppose we wish to achieve availability a_r for connections from one node to another *on the same ring* from links with availability a_l.

Suppose the rings we make our network of have length n. What value should we choose for n. Well, two nodes, at the worst, shall be connected by two paths of length $n/2$, so that the availability of the network, as far as these two nodes is concerned, is

$$1-(1-a_l^{n/2})^2$$

So, the right value of n is the solution of

$$1-(1-a_l^{n/2})^2 = a_r.$$

This equation has the solution

$$n = 2\frac{\log\left(1-\sqrt{1-a_r}\right)}{\log a_l}. \tag{2.4}$$

giving a real value of n, which we should round up to the nearest integer. For sufficiently small values of a_1 and n, a good approximation can also be obtained by solving

$$(n(1-a_l)/2)^2 = 1-a_r,$$

so

$$n \approx \left\lceil \frac{2\sqrt{1-a_r}}{1-a_l} \right\rceil, \tag{2.5}$$

in which $\lceil x \rceil$ denotes the smallest integer which exceeds the number x.

For example, suppose the links have availability 99.8% and the standard sought is an end-to-end availability of 99.9%. Then, using (2.5), $n \approx 32$. Equation (2.4) suggests $n \approx 32$ also.

If the links have availability 99%, on the other hand, we find, using (2.4) or (2.4), that

$$n = 6.$$

The simpler formula appears to be adequate over a wide range of parameter values. However, the assumption that path diversity provided by just *two* alternative diverse paths is probably not adequate in the case where the availability of the component links is rather poor.

The figure of 99% is a rather pessimistic estimate for link availability in any network. On the other hand, the reliability expected of networks is often quite high.

What value should we choose for a_r, the availability across a ring? In a small network, where one ring reaches every node, a_r will be just the desired network availability. But in a larger network, with N nodes say, it will be necessary to pass across several rings, m say (up to N/n of them), to get all the way from the desired origin to the desired destination. In this case, if there was only one way to traverse the rings, the relationship between the ring availability, a_r and the network availability would be $1-a_n = m(1-a_r)$, or looking at the "worst" case, $1-a_n = \frac{N(1-a_r)}{n}$.

However, it is more likely that there would be more than one way (e.g. 2 ways) to traverse the rings, in which case the unavailability of the access across the *home* rings at each end of the path will dominate the unavailability of the end-to-end path, i.e. $1-a_n = 2-2a_r$.

In a network with N nodes, in which link availability is a_l and the desired network availability is a_n, we are lead to the following equation for n, the number of links in each component ring of the network:

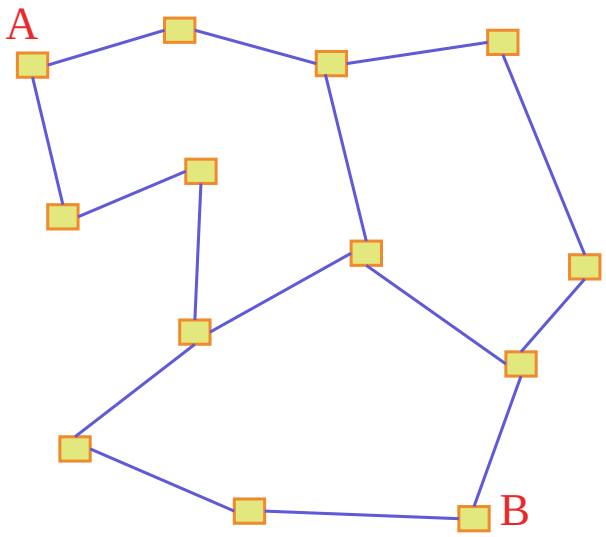
$$n \approx \left\lceil \frac{\sqrt{2(1-a_n)}}{1-a_l} \right\rceil .$$

For example, if $a_n = 99.99\%$, $a_l = 99.9\%$, and $N = 100$, this formula suggests that each ring should have 7 nodes! (Note: N is not actually relevant here.)

Example 2.6. A network of SDH Rings

A network of SDH Rings is depicted in Figure 2.15. Each of the nodes in this network is capable of redirecting byte streams along any of the available paths and thereby recovering from a failure (assuming that the optical fibres contain sufficient spare capacity – how to ensure this will be discussed in Chapter 9).

Figure 2.15: An SDH Ring Network

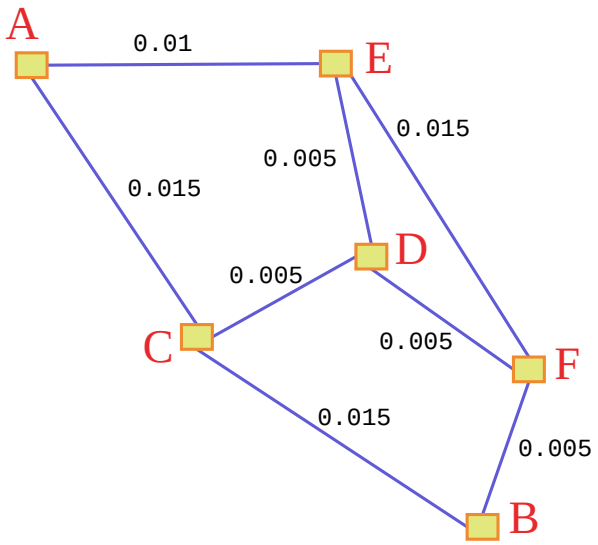


Assuming the availability of each link in this network is $a_l = 99.5\%$, let us calculate the availability of the O-D pair A to B, which should be amongst the most problematic in this network.

From the point of view of availability from A to B, the network is equivalent to the network shown in Figure 2.16, in which the labels represent unavailability.

Dividing the analysis into two cases, one where the link from C to D is up, which happens with probability 0.995 and one where this link is down, which happens with probability 0.005, we find that the unavailability from

Figure 2.16: A Network equivalent to the SDH Ring Network



A to B is $a_{AB} = 0.005 \times x + 0.995 \times y$ where x is the availability from A to B under the assumption that the link CD is down and y is the availability from A to B under the assumption that it is up. It is not difficult to determine, by parallel and serial reduction, that

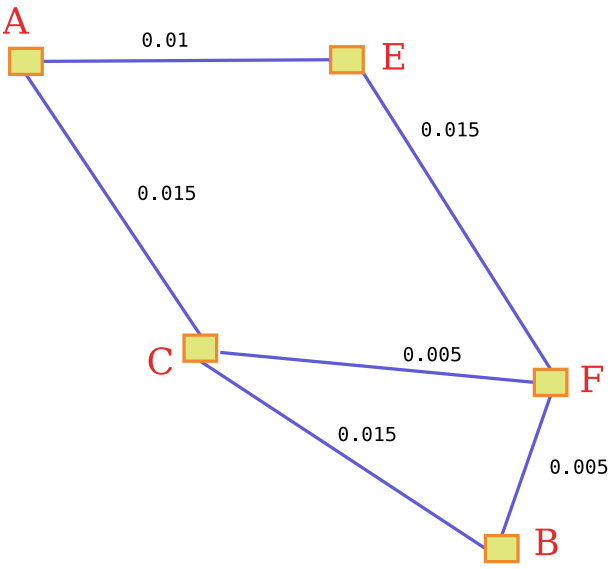
$$x = (0.01 + 0.015 \times (0.005 + 0.005) + 0.005) \times (0.015 + 0.015) \approx 0.00045.$$

On the other hand, to calculate y we again need to consider two cases, one where the link ED is up and one where it is down, giving

$$y = 0.995 \times z + 0.005 \times w$$

in which w is the availability of the path from A to B when CD is up but ED is down. This situation is depicted in Figure 2.17. So

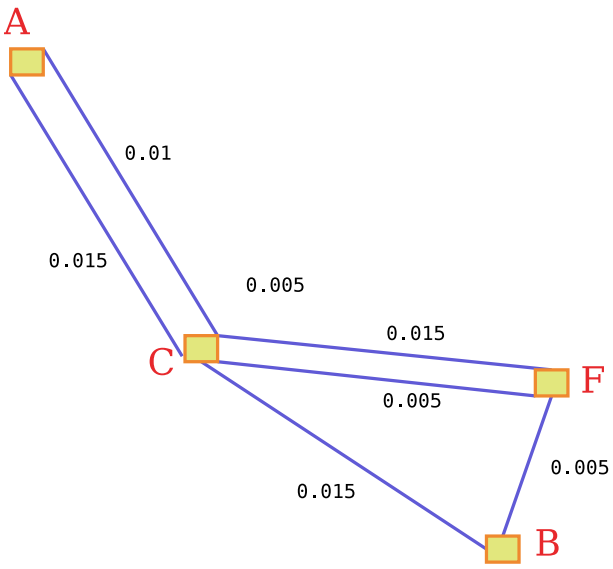
Figure 2.17: The equivalent SDH Ring Network when CD is up and ED is down



$$w = 0.005 \times 0.015 + 0.015 \times (0.01 + 0.015) \approx 0.00045.$$

Now z is the availability from A to B when CD is up and ED is up, which corresponds to the situation depicted in Figure 2.18.

Figure 2.18: The equivalent SDH Ring Network when CD is up and ED is up



So

$$z = 0.015 \times 0.01 + 0.015 \times (0.005 + 0.015 \times 0.005) = 0.00015 + 0.000075 = 0.000225$$

and therefore $y \approx z = 0.000225$ and therefore $a_{AB} \approx 99.9775\% \approx 99.98\%$.

The dominant term in this approximation of the unavailability of the OD pair AB is z, which arises in the case where there is a failure in either the left or the right path of the ring on which A lies, *or* there is a failure on either the left or the right path in the ring on which B lies. All other terms in the preceding calculation turned out to be negligable. In other words, it is as if the links in the core part of the network, where there are quite a few alternative paths, are perfect. This is exactly what was discovered in the analysis of large ring networks in the discussion preceding this example.

Now, extending this example a little, suppose the network is about 4 times as large, as depicted in Figure 2.19. What is the availability of the path from A to B in this network?

By generalization of the example just considered, and by the reasoning in the discussion preceding it, the unavailability will still be dominated by the event that one of the paths on the ring which contains A fails or one of the links in the ring which contains B fails, hence the unavailability from A to B is, again, ≈ 0.00025.

□

Note that we will not attempt to work out the required capacity of any of the links in the networks considered in this chapter. This aspect of network design will be considered in Chapter 9.

Example 2.7. Adding Two Nodes while Preserving Availability

Suppose it is required to add two additional nodes, X and Y, to the network shown in Figure 2.20 (which is the same as Figure 2.5). These additional nodes are to be added near node B and can be connected cheaply to any of B, C, or D, or to each other. It is necessary to achieve an availability of 99.9% for all paths in the network, and the cost of the additional links required will be $50,000.00 for links of availability 99% and $100,000.00 for links of availability 99.9%.

How should the additional nodes be connected to the network to achieve minimum cost?

A minimum cost would be achieved by using just two links, of minimum cost, one for each node. However, the availability of any path to these new nodes would then be at most 99%, which is not adequate. Even if the more

Figure 2.19: A Larger Network of SDH Rings

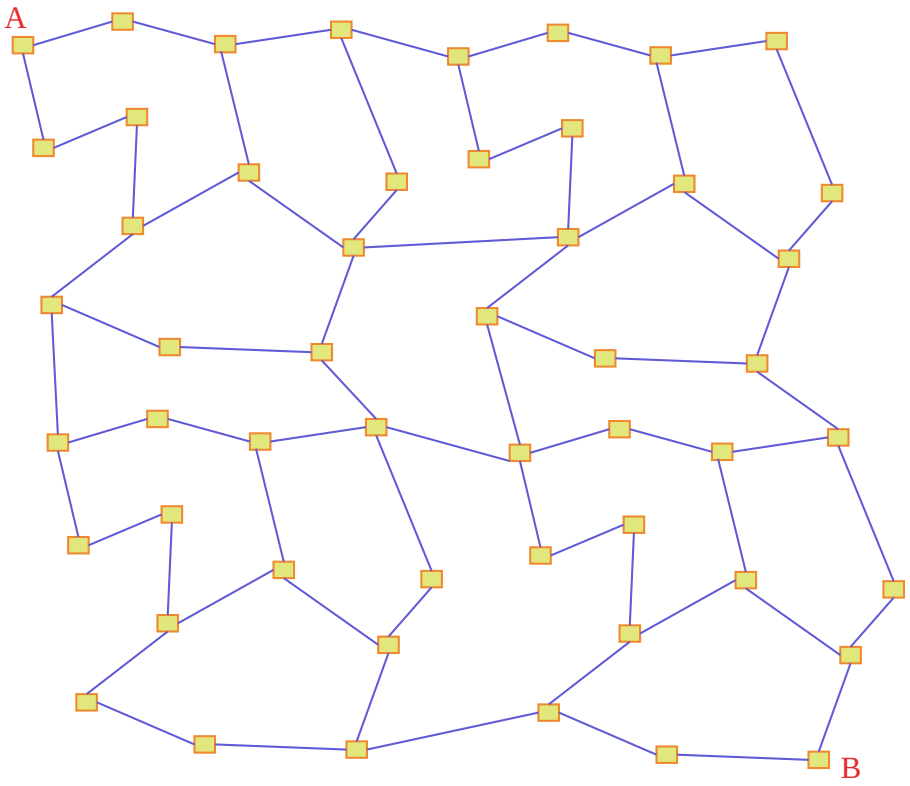
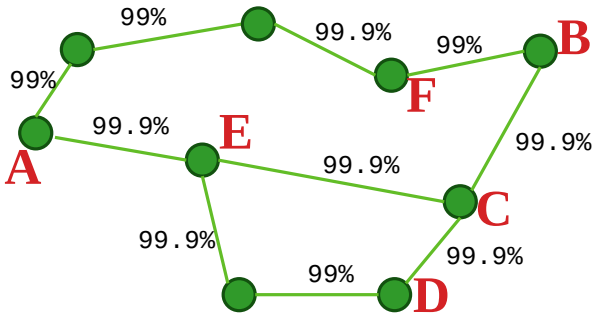


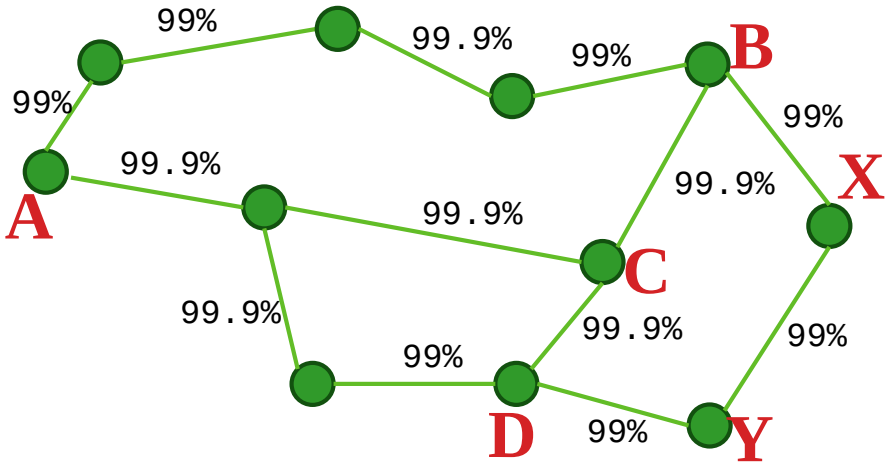
Figure 2.20: A Network needing extension



expensive links are used, the end-to-end availability of paths ending at the new nodes would be inadequate. So, more than two links must be used: will three links be sufficient?

The *only way* to achieve path diversity for the paths to the two new nodes with no more than three new links is to create a new ring which passes through the two new nodes, connecting them to the original network at two separate points, for example, at B and D. Let us call the new nodes X and Y. A network of this sort is depicted in Figure 2.21.

Figure 2.21: A Network with an additional loop



If we use the least-cost links, our total outlay will be $150,000.00. It is apparently not possible to achieve the desired reliability standard at a lower cost.

Does this way of connecting the new nodes achieve the desired standard?

Yes. Consider the availability of the O-D pair (A,X). We have, in this case two obvious alternative disjoint paths – the upper path and the lower path. The availability of the upper path is $\approx 1-4\times 0.01 = 96\%$. The availability of the lower path is $\approx 1-2\times 0.01\%$. So, together these two paths provide an availability of $\approx 1-0.02\times 0.04 = 99.92\% \geq 99.9\%$. So, this design does the trick. □

Exercise 2.6. Design for Reliability

Suppose it has been determined that the O-D availability of all pairs in the network depicted in Figure 2.12 is to be better than 99.9%. The cost of each possible network component is listed in the table 2.1. Determine an appropriate (cheapest) network design.

Origin	Destination	Availability	Cost ($\$\times 100,000$)
A-E	A-E	0.95	1
A-E	A-E	0.99	1.5
A-E	F	0.95	2
A-E	F	0.99	3

Table 2.1: Costs of network components for Exercise 2.4

Components can be used in parallel (as in the link between C and D in Figure 2.3 for example).
You should tackle this problem twice, from the following two different points of view:

(i) As a desert study – i.e. on the assumption that no equipment is in place at the start of the project; and

(ii) as an upgrade – i.e., the network depicted in Figure 2.12 is installed at the start of the project, and the objective is to upgrade the network to the required availability standard.

□

Exercise 2.7. Design for Reliability – Part II

Now let us suppose that the costs of transmission equipment for a network to connect the nodes depicted in Figure 2.22 are as depicted in Table 2.2 (note that there are two cases) and the traffic incident at each node is as shown in Table 2.3.

Capacity	Availability	Installation Cost (case (a))	Installation Cost (case (b))
64kbps	0.99	$10,000	$10,000
2 Mbps	0.99	$11,000	$30,000
20 Mbps	0.99	$12,000	$300,000
200 Mbps	0.99	$13,000	$3,000,000
2 Gbps	0.99	$14,000	$30,000,000
20 Gbps	0.99	$15,000	$300,000,000
200 Gbps	0.99	$16,000	$3,000,000,000

Table 2.2: Costs of network components for Exercise 2.7

Node	Incident Traffic (incoming and outgoing)
A	1 Mbit/s
B	1.5 Mbit/s
C	2 Mbit/s
D	3 Mbit/s
E	5 Mbit/s

Table 2.3: Incident Traffic at each Node for Exercise 2.7

The costs of transmission equipment, as set out in Table 2.2 are assumed to be *distance independent*. This simplifying assumption is not so far from the truth in many situations.

There are two variations to this problem – Case (a), and Case (b) – as depicted in Table 2.2. In both cases, it is required that the network satisfy the reliability constraint that end-to-end availability (in the weak sense) should be at least 99.9% and the traffic constraint that the network should be able to carry all the traffic listed in Table 2.3 (assuming a water-like model for how traffic is carried on a network) when all network components are fully operational. □

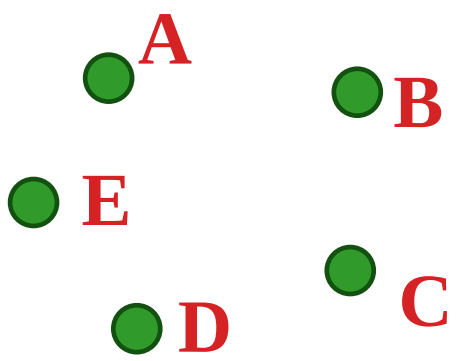


Figure 2.22: Nodes needing a network

Exercise 2.8. Design of SDH Ring Networks

Suppose that the links of an SDH network have measured availability 99.8% and it is desired to build a network out of SDH rings with end-to-end availability 99.95%.

1. How large should the rings be?
2. Check your estimate by recalculating the availability from A to B in a network like the one in Example 2.6 (making sure that the example network has rings no larger than the upper bound you calculated in (1)).

□

2.5 WDM Networks

Wave Division Multiplexing can be used to carry much larger capacities on optical fibers. Already, as mentioned in Subsection 2.3.4, systems carrying 160 different wavelengths are available commercially [11].

The existence of transmission systems with such a large number of wavelengths suggests a different approach to the design of networks, where, rather than converting the optical signal to an electrical signal, switching by SDH technology, and then converting back to an optical signal (in a different wavelength), the signal is transported through a sequence of purely optical switches, staying with the same wavelength over the entire path. In this way, we could imagine a network with purely optical switching elements which is capable of connecting a large number of nodes end-to-end.

An example of such a network is shown in Figure 2.23. The total amount of equipment in this network is not so dramatic, despite the fact that it achieves end-to-end connectivity at 10 Gbit/s. The cost of switching should be much lower than an equivalent network with SDH switches. Obviously, in order to be useful, we will need to be able to construct networks with many more than five switches. How many different wavelengths will be needed for such networks, as the number of nodes varies?

This question is addressed, for ring networks, in [12]. For ring networks of n nodes, the minumum number of wavelengths is at least $\lceil \frac{1}{2} \lfloor n^2/4 \rfloor \rceil$. For example, the minimum possible number of wavelengths required to achieve a full mesh of end-to-end paths in a ring of 5 nodes is, as in Figure 2.23, 3.

Example 2.8. Large WDM Rings

Consider the transmission technology which provides 160 wave lengths on each fibre, mentioned in Subsection 2.3.4. How large a ring could be built with this transmission technology, under the constraint that a complete graph of end-to-end paths can be constructed with optical-only switches?

We need to find the largest n such that

$$\lceil \frac{1}{2} \lfloor n^2/4 \rfloor \rceil \leq 160.$$

Ignoring the $\lceil\cdot\rceil$ and the $\lfloor\cdot\rfloor$, we can estimate $n^2 = 1280$, hence $n \approx 35$. Now, $\lfloor 34^2/4\rfloor = 289$, so for $n = 34$, $\lceil\frac{1}{2}\lfloor n^2/4\rfloor\rceil = 144$; if we try $n = 36$, we find $\lceil\frac{1}{2}\lfloor n^2/4\rfloor\rceil = 162$, and, finally, if $n = 35$, $\lceil\frac{1}{2}\lfloor n^2/4\rfloor\rceil = 153$. Hence, the largest number of nodes which can be accomodated on one ring, under the assumed conditions, is 35.

For reasons of reliability, we would not want to construct a network from one large ring, although it is comforting to know that we could do so if we wanted to. □

Exercise 2.9. Wavelength Assignment for a 6 Node Network

Find an assignment of wavelengths for a network with 6 nodes which provides end-to-end routing for all origin-destination pairs. Calculate the minimum number of wavelengths required, according to [12], and try to achieve this bound. □

2.5.1 Networks of WDM Rings

As mentioned in Example 2.8, we need a structure more complex than a single ring, something more like the network depicted in Figure 2.19. How many wavelengths will be required in this type of network?

We can obtain a lower bound on the minimum number of wavelengths required for a complete set of end-to-end paths, $wA(N)$, as follows. First of all, let us define the *load* on link ℓ, $\pi(\ell,R,N)$ due to a *routing*, R, (a collection of paths) on a network N as the number of paths in R which pass through this link. We are mainly interested in the situations where the paths in R includes precisely one for each origin-destination pair. Let us call such a routing *complete*. Let us now define the minimum link load of a network, N, as

$$\pi(N) = \min\left\{\max_{\ell}\pi(\ell,R,N) : R \text{ is a complete routing}\right\}$$

The minimum link load, $\pi(N)$, is a lower bound on $wA(N)$. An estimate of $\pi(N)$ can be obtained by adopting a shortest-path routing, which in many cases is the routing which minimises $\max_\ell \pi(\ell,R,N)$. Let us denote the maximum link load on any link in a shortest-path routing on a network N by $\pi_s(N)$. Note: there may be more than one shortest-path routing, in which case we shall assume that the one which minimises $\pi_s(N)$ has been chosen.

For example, in a ring of n nodes, the number of links terminating at node k, d_k, $\equiv 2$. It might be useful to check the following reasoning against Figure 2.23. If n is odd, the maximum length of a path which will be used in the shortest path routing is $m = \frac{n-1}{2}$. In this case, we can find the transit load in the shortest path routing (the number of paths passing through the node) at node k by summing over i, the number of hops between the source of the transit path and the node k, giving $\tau_k \equiv \sum_{i=1}^{m-1}(m-i) = \frac{m(m-1)}{2}$. Thus $\tau_k \equiv \frac{(n-3)(n-1)}{8}$, so that $\pi_s(N) = \frac{n-1}{2} + \frac{(n-3)(n-1)}{8} = \frac{n^2-1}{8}$.

If n is even, the shortest path routing is not unique, however, let us adopt a strategy of using alternately the left and right hand paths for the non-unique shortest paths as we progress around the ring (See Figure 2.24), The link load of this routing can be computed as follows.

First, let m denote the length of the longest path which is unique as a shortest path. In this case, $m = \frac{n-2}{2}$. The longest paths in use will be of length $m+1$. For convenience, let us refer to these as the *long paths*. Because of their length, and therefore a greater likelihood of long paths overlapping other long paths, we need more wavelengths for these paths than for any of the other lengths. If m is odd, the number of these long paths on each link will be $\frac{m+1}{2}$ whereas if m is even, the number of long paths on each link will alternate between $\frac{m+2}{2}$ and $\frac{m}{2}$. Now, adding up the contribution of *transit* paths of length 2,...,m, at a specific node, we find $\tau_k[\leq m] \equiv \frac{m(m-1)}{2}$, whether m is odd or even.

As for the transit paths of length $m+1$, there will be $\lceil\frac{m}{2}\rceil$ of them, in the worst (larger) case. Note that we are not counting the paths which originate (or terminate) at this node because they will be counted later. In the case where m is even, this is just $\frac{m}{2}$.

On the other hand, when m is odd the strategy of choosing alternately the left and the right hands for the long paths will produce a balanced load, with either the long path originating from the node on the left of the link passing on this link or the long path originating from the node on the right of the link passing on this link, but not both.

When calculating $\pi_s(N)$, we only need to consider the links with the larger load, so the result we obtain is

$$\pi_s(N) = \begin{cases} \frac{n^2-1}{8} & n \text{ odd}, \\ \frac{n}{2} + \frac{m}{2} + \frac{m(m-1)}{2} = \frac{n^2+4}{8} & n \text{ even}, \ m \text{ even}, \\ \frac{n}{2} + \frac{m-1}{2} + \frac{m(m-1)}{2} = \frac{n^2}{8} & n \text{ even}, \ m \text{ odd}, \end{cases}$$

which is again, consistent with formula given earlier for $wA(N)$ from [12].

It has been hypothesised that $\pi(N) = wA(N)$ for all networks. We can estimate $\pi(N)$ by $\pi_s(N)$ as just demonstrated for rings. This would seem to provide a plausible and practical method for estimating $wA(N)$ for an arbitrary network.

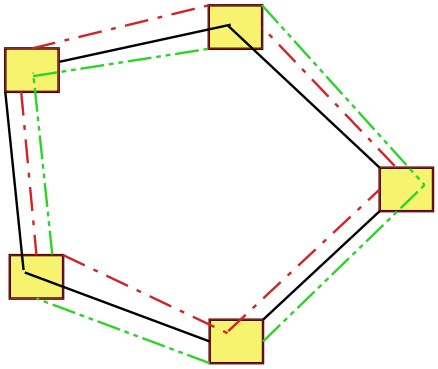


Figure 2.23: Three wavelengths used to connect end-to-end a ring of 5 nodes

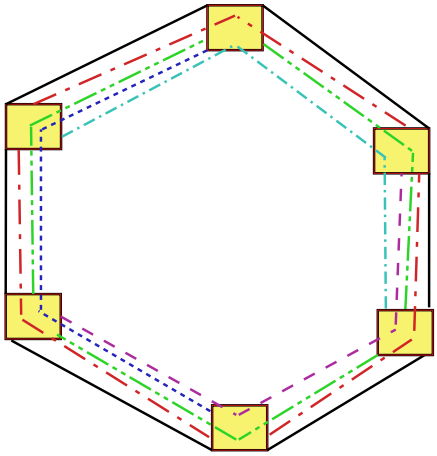


Figure 2.24: Five wavelengths used to connect end-to-end a ring of 6 nodes

2.5.2 Reliable Routing for WDM Networks

The paper [13] considers the logical extension of the preceding topic to the case where wavelength assignment and routing is undertaken with a view to ensuring the survivability of a network under failure conditions. The term survivable is taken, in this paper, to mean that the network will not become disconnected if one of the physical links from which it is built has become disconnected.

This is not the only way to introduce reliability considerations into the design of WDM networks. Another approach is to establish more than one disjoint path for each origin-destination pair to be routed. This is likely

to be a rather expensive approach, however. For example, if the physical network is a ring, this would more than double the number of wavelengths required. This strategy could be made a little more economical by formulating the second disjoint paths merely as plans for paths to be put in place, should they be necessary.

2.6 Further Issues

So far we haven't taken into account the cost of switching and routing. Switches and routers cost money also, and the use of diversity to enhance availability relies on the possibility that routers and/or switches can bring the alternative paths forward whenever they are required.

In networks with a large number of components which are failure prone, even when the probability of failure is low, we reach the point where it must be acknowledged that a *normal operational condition* will include one (or more) failures. An example of this sort is the *transmission network* of a large national telecommunication company, on which many other networks of this company, and its customers, will rely.

In this situation a different design criterion should be considered. Instead of making sure that the network maintains connectivity under all conditions except for a very unlikely collection of situations, it might be desirable to ensure that as well as maintaining connectivity, the network should have adequate capacity (on all links of any importance) even under the first rung of failure conditions – e.g. under all conditions of a *single* failure. This type of design problem will be considered in Chapter 9.

2.7 Closing Comments and Summary

Reliability of networks has been and is likely to remain a very important issue. It is important to be able to *plan*, reliable networks, to analyze the reliability of new or existing networks, and to be able to *design* to meet a reliability standard. In this chapter, almost all phases of the network analysis, planning, design, and maintenance process have been considered as far as they bear on the one performance issue of *reliability*. We have considered a variety of examples, with emphasis on the larger networks. The topic of design to meet a reliability standard will be taken up again in subsequent chapters, particularly Chapter 9, where the important additional issue of how to select the *size* of the links in a network will be addressed.

References

[1] AANSI T1 Committee. AANSIT1 standard for 24 channel pcm systems. Technical report, AANSI T1 Committee, 199x.

[2] ITU. G.xxy e1 transmission systems. Technical report, ITU, 199x.

[3] unknown author. The great at&t signalling failure of 80s. *IEEE Coms Magazine*, 198x.

[4] yy. An australian signaling system number 7 network. *Australian Telecommunications Research*, xx(y), 198x.

[5] R. G. Addie and R. Taylor. An algorithm for calculating the availability and mean time to repair for communication through a network. In *Proceedings of the ITC Specialist Seminar, 1989 [publication in the journal Computer Networks and ISDN Systems is currently being arranged]*. International Teletraffic Congress, 1989.

[6] Techfest. SONET / SDH technical summary. Internet Web Site. http://www.techfest.com/networking/wan/sonet.htm.

[7] ITU. G.774 (02/01) synchronous digital hierarchy (sdh) - management. Technical report, ITU, 2001.

[8] The International Engineering Consortium. `http://www.iec.org/tutorials/sdh/index.html`Synchronous Digital Hierarchy (SDH) Tutorial. Technical report, The International Engineering Consortium, 2001.

[9] sonet.com. The SONET home page. Internet Web Site. http://www.sonet.com/.

[10] Jeffrey Lynch. OIF electrical interfaces. Technical report, Optical Internetworking Forum, April 2001. http://www.oiforum.com/.

[11] Lucent Technologies. Lucent - product and services. Internet Web Site. http://www.lucent.com/products.

[12] J C Bermond, L. Gargano, S. Perennes, A. Rescigno, , and U. Vaccaro. Efficient collective communications in optical networks. In *Lecture Notes in Computer Science*, volume 1099, pages 574–585. Springer, 1996. http://citeseer.nj.nec.com/bermond96efficient.html.

[13] Eytan Modiano and Aradhana Narula-Tam. Survivable routing of logical topologies in WDM networks. In *INFOCOM*, pages 348–357, 2001. http://citeseer.nj.nec.com/modiano01survivable.html.

Chapter 3

Performance Analysis and Modeling

In this chapter we shall learn how to analyze the delay which will be experienced by data traveling through a network, how to be able to determine the proportion of lost packets over a communication path through a network, and we shall begin to understand the concept of *traffic* a little more deeply. We shall also investigate the way in which the end-to-end control protocols affect throughput and performance of TCP/IP networks.

In order to design networks, we need to know how they work: how the capacity of links affects performance, how buffers affect loss and delay, and so on. In order to do all this, we need to understand the concept of *traffic*.

So far, in Chapters 1 and 2, where necessary to talk about and model traffic, we have thought of it as a fluid, like water, and the paths through network we have thought of like pipes. This is a well-established and useful model of traffic, but it is not the only one, and it is not really adequate in order to study the behaviour of real network traffic [1, Chapter 6].

A more sophisticated model of traffic is going to require mathematics. Specifically, we will need the concept of a *stochastic process*. Before we introduce this concept, we will review the reasons for using mathematical models, and how they can be used to solve real world problems. In addition, in Section 3.1 we review all the mathematical concepts required in the remainder of this book. This section can be omitted by readers with a modest familiarity with the concepts of random variable and stochastic process.

3.1 Probability Theory and Stochastic Processes

3.1.1 Mathematical Models

Real world problems can always be solved by common sense. True of false?

True! Mathematical problems, theoretical problems, financial problems, and even psychological problems can be solved by common sense also. But it has to be the right common sense, and it has to be available in the head where the problem is being solved, which is very often not the case. Unfortunately, common sense is not so common.

Where does common sense come from? A lot of common sense is traditional, passed down the ages. However in a changing world we need new types of common sense. This new common sense also comes from a variety of sources: insight, experimentation, and, a lot of it, comes from science, which relies on careful collection of the facts, a balanced understanding of the competing issues and factors, and, in many, many cases, mathematical modelling.

Look around the room you are in (if you are in a room, otherwise look around at the buildings you can see), and you will probably see many devices which could not be designed or built without a detailed, scientifically valid understanding of electricity, materials, light, radio waves, and so on. In most cases the science underlying technology depends crucially on mathematical models – mathematical models of the atom, of molecules, of fluids, materials, magnetic and electrical fields, communication through wires, electronic processes in semiconductors, and so on.

In the case of communication systems and networks, the crucial mathematical models which are used every day by researchers, engineers, designers, and scientists in this field make use of probability theory, and in particular

the concepts of random variables and stochastic processes.

3.1.2 Probability Theory

At the heart of probability theory, random variables, and stochastic processes we have the concept of a *probability space*, typically denoted by Ω. Every outcome which can happen, in the ensuing experiment or experiments which we are currently contemplating, is an *element* of Ω and all the *events* which can possibly happen are *subsets* of Ω. Each subset, e.g. $A \subseteq \Omega$, is also assigned a *probability*, $P\{A\}$.

For example, if the experiment under consideration is the tossing of three dice until three sixes are thrown, Ω will be the set

$$\Omega = \left\{ \left(\begin{pmatrix} 6 \\ 6 \\ 6 \end{pmatrix} \right), \left(\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 6 \\ 6 \\ 6 \end{pmatrix} \right), \left(\begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 6 \\ 6 \\ 6 \end{pmatrix} \right), \left(\begin{pmatrix} 1 \\ 1 \\ 3 \end{pmatrix}, \begin{pmatrix} 6 \\ 6 \\ 6 \end{pmatrix} \right), \ldots, \left(\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 6 \\ 6 \\ 6 \end{pmatrix} \right), \ldots \right\} \tag{3.1}$$

(each element in this set is a series of 3-vectors and each 3-vector must contain only the numbers 1, ..., 6. The vector $\begin{pmatrix} 6 \\ 6 \\ 6 \end{pmatrix}$ can *only* occur in the final position).

An event in this case could be, for example, the set of outcomes in which the number of times the dice are thrown is even. This event can also be defined by listing all the outcomes where the number of throws of the dice is even. Each outcome has a probability and the probability of an event can be calculated by simply adding up all the probabilities of the individual outcomes, although this might not always be the easiest way to do it.

3.1.3 Random Variables

A random variable, eg X, is a real valued function defined on Ω. Thus, whatever happens in the experiment under consideration, the outcome will be an element ω in Ω and the value taken by X will then be $X(\omega)$. For example, the total number of 5s which were thrown in the entire experiment, or the sum of all the throws which included a 6.

We will usually deal with a random variable, e.g. X, by means of more specific characteristics such as its *mean*, *variance*, and its *distribution*. Before describing these in more detail, we need to recall a more fundamental concept in probability theory, the concept of *expectation*.

Definition 3.1 *The expectation of a random variable is its mean, or average, value over the possible outcomes in the probability space. If the probability space is finite, and X is the random variable,*

$$E\{X\} = \sum_{\omega \in \Omega} P\{\omega\} X(\omega).$$

In the case where Ω is not finite, we will need to define an integral over the set Ω and in this case,

$$E\{X\} = \int_{\Omega} P\{\omega\} X(d\omega).$$

It will not be important to review in detail the case where Ω is infinite. The finite case can be considered a satisfactory guide for how things work in the infinite case.

Now we can define the familiar parameters of a random variable; the mean of a random variable, X, is identical to its expectation, $E\{X\}$. Its variance is $\mathrm{Var}(X) = E\{(X - E\{X\})^2\}$, and its *distribution* is the function F_X where

$$F_X(x) = P\{\{\omega : X(\omega) \leq\}\}, \qquad x \in \mathbf{R}.$$

Here are a few basic laws concerning expectation:

$$\begin{aligned} E\{X+Y\} &= E\{X\} + E\{Y\} \\ E\{aX\} &= aE\{X\}, \end{aligned}$$

Together these properties state that the *expectation operator* is *linear*.

Now that we are dealing with two random variables at once, we should take time to recall the *covariance*, which is defined by

$$\mathrm{Cov}(X,Y) = E\{(X - E\{X\})(Y - E\{Y\})\},$$

and the correlation (or correlation coefficient) between two random variables,

$$\rho_{X,Y} = \frac{\mathrm{Cov}(X,Y)}{\sqrt{\mathrm{Var}(X)\mathrm{Var}(Y)}}.$$

The covariance operator, $\mathrm{Cov}(\cdot,\cdot)$ is also linear in both its arguments, e.g.

$$\mathrm{Cov}(aX + bY, Z) = a\mathrm{Cov}(X,Z) + b\mathrm{Cov}(Y,Z).$$

Notice that the *variance* can be expressed as the covariance of a random variable with itself:

$$\mathrm{Var}(X) = \mathrm{Cov}(X,X). \tag{3.2}$$

The properties for variances corresponding to those just listed for the expectation operator are

$$\mathrm{Var}(X+Y) \quad = \quad \mathrm{Var}(X) + \mathrm{Var}(Y) \tag{3.3}$$

so long as X and Y are uncorrelated, i.e. $\mathrm{Cov}(X,Y) = 0$, and

$$\mathrm{Var}(aX) \quad = \quad a^2\mathrm{Var}(X).$$

If the two random variables X and Y *are* correlated, (3.3) becomes:

$$\mathrm{Var}(X+Y) = \mathrm{Var}(X) + 2\mathrm{Cov}(X,Y) + \mathrm{Var}(Y), \tag{3.4}$$

which follows by using (3.2) in conjunction with the linearity properties of the covariance operator.

One last property of the covariance of two random variables will be needed in the sequel:

$$|\mathrm{Cov}(X,Y)| \leq \sqrt{\mathrm{Var}(X)\mathrm{Var}(Y)}. \tag{3.5}$$

Equality in (3.5) occurs only if $X = cY$ for some constant c. This property is not obvious but we shall not present a proof. This inequality is actually a special case of the Cauchy-Schwartz inequality.

3.1.4 Conditional Mean and Variance

We also need, from time to time, the concept of *conditional mean*, or *conditional expectation* (another name for the same thing), and *conditional variance* . If A and B are two random variables, the conditional expectation of A given B, denoted $E\{A|B\}$ can be envisaged as a random quantity which varies as the possible values of B vary, and for each possible value of B it is the mean value of A in the circumstance that B takes a particular value.

However, in order to make this definition more precise we are forced to express it a little more abstractly. In the event, the correct definition of $E\{A|B\}$ is that it is a random variable with the property that if C is any *other* random variable which is definable purely in terms of the random variable B, then

$$E\{AC\} = E\{CE\{A|B\}\}. \tag{3.6}$$

It is not obvious that this equation uniquely defines $E\{A|B\}$. However, this is indeed the case. It would be inappropriate to go into the details of this to any greater degree. For more discussion see any good book on probability theory, e.g [2, 3].

The conditional variance can readily be defined in terms of the conditional mean by the formula:

$$\mathrm{Var}(A|B) = E\{(A - E\{A\})^2|B\}.$$

This brings us to some useful formulae by means of which the mean of a random variable can be expressed in terms of its conditional mean, and the variance of a random variable can be expressed in terms of its conditional variance *and* its conditional mean. For the former, we have:

$$E\{A\} = E\{E\{A|B\}\}, \tag{3.7}$$

which may seem somewhat trivial, but it has its uses. On the RHS, the outer expectation is taken over the range of possible values for B while the inner, conditional expectation, is meant to be done separately for each possible value of B, and in each case the expectation is over the range of cases where B takes this particular value. Note that this formula follows from (3.6) by substituting 1 for C.

The variance formula is more interesting and has an immediate application below.

$$\mathrm{Var}(A) = E\{\mathrm{Var}(A|B)\} + \mathrm{Var}(E\{A|B\}). \tag{3.8}$$

The first term in this formula is the expected value (over values of B) of the *conditional* variance of A given that B takes a specific value and it takes into account the variance of A when B is fixed. The second term is the variance (over values of B) of the *conditional mean* value of A given B takes a specific value. It takes into account the variance due to the variation of B.

Example 3.1. Variance of a Product

Suppose $A = CB$ where C and B are independent random variables. What is the variance of A in terms of the mean and variance of C and B? Applying (3.8) to A and B we find:

$$\begin{aligned}
\mathrm{Var}(A) &= E\{\mathrm{Var}(A|B)\} + \mathrm{Var}(E\{A|B\}) \\
&= E\{B^2\mathrm{Var}(C)\} + \mathrm{Var}(E\{C\}B) \\
&= \left(\mathrm{Var}(B) + (E\{B\})^2\right)\mathrm{Var}(C) + (E\{C\})^2\mathrm{Var}(B) \\
&= (E\{B\})^2\mathrm{Var}(C) + \mathrm{Var}(B)\mathrm{Var}(C) + +(E\{C\})^2\mathrm{Var}(B)
\end{aligned} \tag{3.9}$$

□

Example 3.2. Calculation of a mean and variance

Suppose it has been estimated that the average speed of a certain yatch is $S = K \times W$, where W is the wind speed on the lake where this yatch is sailed, measured in metres per second and K has a uniform distribution on the interval $[0,1]$. The uniform distribution on the interval $[0,1]$ has mean 0.5, variance $\frac{1}{12}$, and therefore standard deviation $\frac{1}{2\sqrt{3}}$.

Now suppose that the wind speed, W, has mean 1 and standard deviation 0.5 metres per second. Let us now calculate the mean and standard deviation of the yatch speed.

Note that, given $W = w$, S is uniformly distributed on the interval $[0,w]$, and therefore has mean $0.5w$ and standard deviation $\frac{1}{2\sqrt{3}}w$. Using the concept of conditional expectation and conditional variance, we can re-express these observations in the form:

$$\begin{aligned}
E\{S|W\} &= 0.5W \\
\mathrm{Var}(S|W) &= \frac{1}{12}W^2
\end{aligned}$$

So, from (3.7), $E\{S\} = E\{E\{KW|W\}\} = E\{0.5W\} = 0.5 \times 1 = 0.5$ metres per second. As for the variance, from (3.8),

$$\begin{aligned}
\mathrm{Var}(S) &= E\{\mathrm{Var}\{S|W\}\} + \mathrm{Var}(E\{S|W\}) \\
&= E\{\frac{1}{12}W^2\} + \mathrm{Var}(0.5W) \\
&= \frac{1}{12}(1 + 0.25) + 0.25 \times 0.25 \\
&= \frac{1}{6}.
\end{aligned}$$

□

Exercise 3.1. Calculation of a variance

Suppose, instead of the speed of the yatch in the previous example following the formula $S = KW$, the yatch speed is given by

$$S = 0.8W + 0.2KW,$$

with W and K as before. Calculate the mean and variance of S. Hint: you might find it useful to re-use the result derived in Example 3.2. □

3.1.5 The Gaussian Distribution

The Gaussian distribution (normal distribution) of a single random variable, e.g. X, with mean μ and standard deviation σ takes the form:

$$F_{\mu,\sigma}(x) = P\{X \leq x\} = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{(y-\mu)^2}{2\sigma^2}} dy. \tag{3.10}$$

The normal distribution cannot be expressed in a simpler form than this. That is to say, the integral at (3.11) cannot be solved in terms of more elementary functions of mathematics.

The Gaussian distribution can be expressed in terms of the *error function*, erf, which is sometimes useful because the error function is often available in spreadsheets and other computer packages. The error function is very closely related to the Gaussian distribution, and is defined as follows [4]:

$$\text{erf}(x) = \frac{1}{\sqrt{\pi}} \int_{-x}^{x} e^{-y^2} dy. \tag{3.11}$$

In terms of the error function, the normal distribution can be expressed as

$$F_{\mu,\sigma}(x) = \frac{1}{2}\left(1 + \text{erf}\left(\frac{(x-\mu)}{\sigma\sqrt{2}}\right)\right), \qquad x \in (-\infty, \infty).$$

The *multivariate normal distribution* is also defined by an integral, in a manner similar to, but more complicated than (3.11). There is no need for us to ponder on this, however, since there will be no need to *evaluate* values of this distribution at any stage. For more information about the multivariate distribution see [4] or [2]

3.1.6 Stochastic Processes

So far the elaborate terminology of *probability space*, *outcomes*, and *events* has not achieved all that much. When dealing with one random variable we could express all the relevant facts and all the important calculations by means of the mean, variance and distribution, none of which really need the concept of a probability space.

But, a stochastic process is not so easy to define without a probability space, while *with* a probability space it is *easy* to define.

Definition 3.2 *A stochastic process over a continuous time parameter, $\{X_t\}_{t\in\mathbf{R}}$ is defined as a collection of random variables $\{X_t\}$, $t \in \mathbf{R}$.*

A stochastic process over a discrete time parameter, $\{X_n\}_{n\in\mathbf{Z}}$ is defined as a collection of random variables $\{X_n\}$, $n \in \mathbf{Z}$.

What makes these collections of random variables which we call a stochastic process interesting is the fact that the random variables are *related* to each other. We do not need to state this explicitly, or indicate specifically *how* they are related to each other, because this is all covered in their implied connection, the fact that they are all defined in terms of the same probability space, Ω.

For example, let us define a stochastic process in terms of the probability space, Ω, introduced in the previous subsection, as defined in (3.1) Set X_k as the sum of the numbers on the dice at throw k, if the experiment has k throws, or zero otherwise. For example, the sequence

$$12,\ 3,\ 33,\ 36,\ 0,\ 0,\ 0,\ \ldots$$

could occur as a value of the stochastic process X in one of these experiments. On the other hand, the sequence

$$12,\ 3,\ 33,\ 32,\ 0,\ 0,\ 0,\ \ldots$$

could not occur, because the zero values can only occur after the number 36.

The stochastic processes we will be interested in will all be related to *traffic*, i.e. the bits, bytes, packets, and messages that we communicate across networks. However, it is important to have a crystal clear framework within which to discuss this traffic.

3.1.7 Statistics of Stochastic Processes

When we reviewed *random variables*, in Subsection 3.1.3, we discussed their basic statistical features: their mean, variance (or standard deviation), and their distribution. In the case of stochastic processes, the collection of basic statistics needs to expand somewhat. A collection of stochastic process is just a collection of random variables, indexed by a parameter representing time, so it would appear that we should talk about the mean of each and every one of these random variables.

However, it is frequently useful to add some assumptions which reduce the range of possibilities. In particular, it is common to assume that all the values of the stochastic process have the same mean,

$$E\{X_t\} = E\{X_0\}, \qquad t > 0,$$

and the same variance, $\mathrm{Var}(X_t) = \mathrm{Var}(X_0)$, $t > 0$, although other possibilities will also be considered below.

As well as these parameters of a stochastic process, another useful parameter is the *autocovariance*:

$$C(s,t) = E\{(X_s - E\{X_s\})(X_t - E\{X_t\})\}, \qquad s,t \in \mathbf{R}.$$

The typical way in which the autocovariance becomes a little simpler occurs when $C(s,t)$ depends only on $t-s$, and so we write it as $C(t-s)$.

When all these simplifications occur at once, so the mean and variance do not depend upon time and the autocovariance is a function of $t-s$, we say that the process is *stationary* .

It is tempting to think that the distribution of a stochastic process is nothing more than all the distributions of the values of the stochastic process, i.e. the random variables. However, this is *not* usually appropriate. What we need instead is the *joint* distribution of all the values, or, what amounts to the same thing, the collection of the joint distributions of every finite collection of the values at times t_1, t_2, ..., t_n, for all possible choices of n, and t_1, ..., t_n.

These distributions are known as the finite dimensional distributions, or the *fidi* distributions as they are sometimes known. The collection of all the fidi distributions is sufficient to completely characterise a stochastic process. This is important to know, partly because, in effect, it demystifies the concept of a stochastic process. If we know the fidi distributions we know everything! For example, if the fidi distributions are jointly Gaussian (jointly normal – Gaussian is a synonym for normal when we talk about probability distributions), then we say the process is Gaussian. This is how the term Gaussian stochastic process is defined.

3.2 The Causes of Loss and Delay

3.2.1 The Causes of Delay

There are three major causes of delay experienced by data passing through communication networks, together with one minor cause which arises in the case of voice communication over packet networks:

Propagation Delay: This is the delay which is caused by distance. According to Einstein, who's views on the subject are still accepted today, communication across a distance cannot take place faster than the speed of light. Much of the communication we undertake today is carried by light. Electrical signals carried on wires also travel at approximately the speed of light. Fast as it is (300 million meters per second), the delay due to a pulse of light traveling from one side of the earth to the other is still significant.

Transmission Delay: Transmission delay is much more significant than propagation delay over short distances. It is the delay caused by the fact that it takes time to feed a signal onto the communication line. For example, suppose the line is transmitting at 10 Mbit/s, and the user wishes to send a file of 10 Mbytes. 10 Mbytes is 80 Mbits, so, all things being equal, the transmission will take 8 seconds. This is the transmission delay.

Queueing Delay: This, the most *interesting* delay is caused by storage and retransmission of bits, bytes, and packets, in equipment lying in the network between the origin and the destination of a transmission. Buffering can take place at the origin, and at any point along the way where retransmission takes place.

Packetization Delay: In packet networks, one more factor affects the end-to-end and round-trip delay for voice: the delay inherent in storing up a whole packet of digital audio data in a packet.

We shall now consider each of these contributers to delay in a little more detail.

Propagation Delay

Propagation delay can be a rather significant factor in network performance. The circumference of the earth is very close to 40,000 Km; in fact the meter was defined in such a way that this would be the case. Therefore, travelling half way across the earth might be expected to take approximately $20,000/c \approx 20,000/300,000 = 66$ milliseconds. This approximation presumes that the *path* of the transmission follows a *great circle*, i.e. a circle whose centre is approximately at the centre of the earth. In practice the path of a transmission link is likely to be much more circuitous than this, and this could easily increase the propagation delay by a further factor of two, or even three, so the delay in traveling half way around the earth might be more like 132 milliseconds. The return trip could therefore be more like 260 milliseconds.

Today's satellites are normally situated in geostationary orbit, 35,900 Km above the surface of the earth. A light (or radio) signal will take about 130 milliseconds to travel from the earth to a geostationary satellite. The trip from the earth to the satellite and back will therefore take 260 milliseconds – a quarter of a second. The *round trip time*, from one location on the earth to another *and back* will therefore be about a half a second.

When delays reach this level, i.e. half a second, interactive voice communication can become difficult. Before considering this issue, it is worth mentioning an approach to satellite communication which avoids this problem.

There is another type of satellite system, known as *Low Earth Orbit (LEO)* satellites . The first example of this system, the Iridium system, has already been planned, designed, deployed, declared financially unworkable, and was nearly de-commissioned! This appeared to be the end of the road, but then, after the whole system was sold to the *new* Iridium company [5], at a bargain basement price (by the creditors of the old company), this sytem is again up and running. The future of the new Iridium is uncertain at the moment.

The low orbit satellite concept has two major advantages which follow from the fact that the satellites are closer to the earth than the geostationary orbit satellites: (i) the power required to communicate with these satellites is lower, leading to cheaper earth stations, and the possibility of light mobile communication devices, which communicate via the satellite, and (ii) the delay experienced in transit from earth to satellite and back is greatly reduced. For this reason, the low earth orbit satellite concept is attractive and even if the Iridium system were to be unsuccessful, the concept would be likely to be revisited at some stage in the future.

Two problems associated with propagation delay need now to be mentioned: *lip synchronization (lip-synch)*, and *echo*. Whenever a video transmission is received it is important to ensure that the video and the audio signals are synchronized. A transmission technology which does not achieve this must be regarded as severely degraded, and not acceptable for widespread use.

Echo occurs whenever the audio signal from the speaker at the end of a transmission medium is allowed to leak into the microphone at the same location. This *leaking* of an audio signal is virtually unavoidable, because the last few centimeters of audio communication (think of a phone) take place in the air. The disturbing effect of the echo on communication varies from insignificant, when the delay is short, or the attenuation of the echo is high, to *severe*, when the delay is long and the level of attenuation is low.

If the round-trip delay exceeds about 200 milliseconds, it becomes essential to interpose a mechanism for attenuating (reducing the volume of) the echo. Two techniques are in widespread use:

- echo suppression, and
- echo cancellation.

Echo suppression is a fairly intrusive technique although it is simple and not difficult to implement. It works by forcing all communication to take place in only one direction at a time. When echo suppression is in use it is impossible for both parties involved in a conversation to speak at the same time and attempts to do so may prevent any communication from taking place. Despite this, people seem to be able to readily adapt to the use of echo suppressors, although not without realising that they are engaging in a somewhat *limited* standard of communication.

Echo cancellation is a much more complex and expensive technique. It works by first estimating the transfer function for the echo path, then interposing an additional electronic filter in this path which effectively cancels the echo signal.

Echo is only a problem for interactive voice communication, or any service, like interactive video, which incorporates interactive voice communication. Even if interactive voice communication is not always required, any service which might, from time to time, make use of interactive voice communication will potentially be affected by echo. Round-trip delays of longer than a second cause degradation of two-way communication even when echo has been suppressed or cancelled, and even a half second delay can be disturbing.

Packetization Delay

This type of delay has become more interesting in the light of the fact that there is at present a growing interest in using the Internet, or TCP/IP networks at any rate, for voice communication. In fact, voice over IP is already widespread and growing rapidly. This brings us to another contributor to delay in any approach where a voice signal is digitized and stored in a packet for the purpose of communication.

Suppose voice is digitized at the rate of 64 kbit/s, i.e. 8 bits sampled 8000 times per second (this is in fact the most common standard for digitization of voice). At this sampling rate, a 1000 byte packet is capable of storing 125 milliseconds of voice. The packet cannot be sent across a network until it is full, so there will be a delay of 125 milliseconds while the packet fills, in this case. If smaller packets are used, e.g. 100 bytes, the time taken to store a packet is reduced to 12.5 milliseconds, which is much more acceptable. It should be possible to ensure that this *packetization delay* will be incurred only once, at the location where the packets are filled.

Consider the following experiment. Two telephones are connected to a small, very fast, TCP/IP network. So, as the voice communication takes place, packets are filled with bits representing the voice signal of the speaker. The time it takes to fill a packet, 12.5 milliseconds, is incurred as the packets are assembled. Then the packets are sent to their destination in a very short (insignificant) time. The packet can then be replayed immediately. Although it takes 12.5 milliseconds to replay the packet, this is not really an additional delay. It just reflects the fact that every byte in the packet is delayed by the same amount.

Totally avoiding this delay is feasible but likely to be very difficult in a packet network, although a very simple way to reduce this delay, if an application demands particularly small delays, is to use shorter packets, or unfilled packets.

Another interpretation of packetization delay is that it is really just a special case of transmission delay, namely the transmission delay incurred in using the part of the transmission path which passes through the air, as an audio signal. This interpretation is a little difficult to get used to, but after a bit of thought it makes sense. Moreover, it suggests a very interesting possibility for increasing the speed of communication: bypass the audio!

Example 3.3. From Australia to Silicon Valley

Consider a path through a TCP/IP network as follows: the path starts at a home and passes through a modem connection to an ISP, in Brisbane, say, then to Sydey, then to San Francisco, and finally, to the web site of interest in Silicon Valley, California.

Let us consider and explain *all* the significant components of:

(i) delay,

(ii) delay variation,

(iii) loss.

Let us make sure to indicate *why* a particular component of delay should be significant and let us assume that the average packet length on the end-to-end path across the TCP/IP connection is 750 bytes.

In this example, an important component of delay is undoubtedly going to be *propagation delay*. This is because it is a long way from Sydney to San Francisco – not quite half way around the world – 100° out of 360°, in fact. The circumference of the earth is approximately 40000 kilometers, so, the distance from Sydney to San Francisco is approximately 11,111 kilometers.

Light travels at approximately 300,000,000 meters/s, so the delay in traveling from the origin to the destination should be $11111/300000 = 0.037$, i.e. 37 milliseconds. In fact, propagation delay over this path turns out to be (the last time I measured it) considerably more than this because the actual path is quite different from the great circle path which would have the minimum distance. We shall reconsider this issue a little later.

Another important component of delay in this case will be transmission delay. Transmission delay will occur at each point along the way where the packet has to be transmitted. Since there are likely to be about 15 hops in a journey like this, we can expect 15 transmission delays. In cases where the link speed is quite high, the transmission delay will be almost insignificant. However, quite a few of the links are likely to be of modest speed, e.g. 2 Mbit/s. The first and last links on the path, in particular, are often quite slow. However, let us leave out the modem link just for the moment, and suppose that among the other links there are 5 links as slow as 2 Mbit/s. Since the packet is 750 bytes in length (on average), the delay in each case will be $750/(2000000/8) = 0.003$, and so, these transmission delays add another 15 milliseconds of delay.

The modem link presents particular problems because it is so slow. Let us suppose, for example, that our modem is operating at 56 kbit/s. Then the transmission delay across this link will be $750/(56000/8) = 0.107$ seconds, i.e. 107 milliseconds. Note that experiments made using the `ping` command will not normally generate packets of this length, so that transmission delay of ping packets over the modem will be much less than 100 milliseconds. The ping command (under Linux at any rate) can be configured to send longer packets, which might be useful as a way to display the effect of transmission delay on network performance.

Last, but not necessarily least, let us consider queueing delay. This is the time a packet spends waiting in queues along the way. It is hard to know how long this might be, however we know that it is always a factor, because we can see that the time taken by packets to travel across a path such as the one under consideration, and back shows considerable variation, and out of all the contributions to delay, queueing delay is the *only* factor which exhibits significant variation. However, queueing delay is not the only way in which delay variation can occur. Although the other types of delay are all fixed, it is possible that successive packets might follow *different paths*, and the delay of each path will be significantly different from the delay on the other paths. In practice, we tend to observe that packets follow the same path virtually all the time, however.

To a modest approximation, we can expect the mean and the standard deviation of queueing delay to be similar. Also, if we conduct a series of experiments, there is a reasonable expectation that the minimum delay observed, over a sufficiently large number of experiments, will illustrate the case where queueing delay is approximately zero. In this way, we should be able to estimate the mean and standard deviation of queueing delay.

In this particular case, I would expect mean queueing delay to be approximately 40 milliseconds, and the standard deviation to be similar.

What about loss? It is hard to say what the loss might be, except by making some experiments, or perhaps by recalling experience of such experiments – whether formally undertaken for the purpose of gaining such experience or not.

Here is such an experiment, carried out early in the year 2001, from a computer at a university laboratory in Queensland, Australia (so that transmission delay over a modem did not occur in this case):

```
addie@decius : ~ ping www.apple.com
PING www.apple.com (17.254.0.91) from 139.86.137.50 : 56(84) bytes of data.
64 bytes from www.apple.com (17.254.0.91): icmp_seq=0 ttl=49 time=274.1 ms
64 bytes from www.apple.com (17.254.0.91): icmp_seq=1 ttl=49 time=272.1 ms
64 bytes from www.apple.com (17.254.0.91): icmp_seq=2 ttl=49 time=272.7 ms
64 bytes from www.apple.com (17.254.0.91): icmp_seq=3 ttl=49 time=272.7 ms
64 bytes from www.apple.com (17.254.0.91): icmp_seq=4 ttl=49 time=272.7 ms
64 bytes from www.apple.com (17.254.0.91): icmp_seq=5 ttl=49 time=272.6 ms
64 bytes from www.apple.com (17.254.0.91): icmp_seq=6 ttl=49 time=273.1 ms
64 bytes from www.apple.com (17.254.0.91): icmp_seq=7 ttl=49 time=273.1 ms
64 bytes from www.apple.com (17.254.0.91): icmp_seq=8 ttl=49 time=273.5 ms
64 bytes from www.apple.com (17.254.0.91): icmp_seq=9 ttl=49 time=273.5 ms
64 bytes from www.apple.com (17.254.0.91): icmp_seq=10 ttl=49 time=273.0 ms
64 bytes from www.apple.com (17.254.0.91): icmp_seq=11 ttl=49 time=275.4 ms
64 bytes from www.apple.com (17.254.0.91): icmp_seq=12 ttl=49 time=273.4 ms
64 bytes from www.apple.com (17.254.0.91): icmp_seq=13 ttl=49 time=283.9 ms
64 bytes from www.apple.com (17.254.0.91): icmp_seq=14 ttl=49 time=277.9 ms
64 bytes from www.apple.com (17.254.0.91): icmp_seq=15 ttl=49 time=273.8 ms
64 bytes from www.apple.com (17.254.0.91): icmp_seq=16 ttl=49 time=276.8 ms
64 bytes from www.apple.com (17.254.0.91): icmp_seq=17 ttl=49 time=277.8 ms
64 bytes from www.apple.com (17.254.0.91): icmp_seq=18 ttl=49 time=273.2 ms
64 bytes from www.apple.com (17.254.0.91): icmp_seq=19 ttl=49 time=279.7 ms
64 bytes from www.apple.com (17.254.0.91): icmp_seq=20 ttl=49 time=273.7 ms
64 bytes from www.apple.com (17.254.0.91): icmp_seq=21 ttl=49 time=273.6 ms
64 bytes from www.apple.com (17.254.0.91): icmp_seq=22 ttl=49 time=272.5 ms

--- www.apple.com ping statistics ---
23 packets transmitted, 23 packets received, 0% packet loss
round-trip min/avg/max = 272.1/274.5/283.9 ms
```

Here is another experiment, this time using `traceroute`. The traceroute command sends a series of packets to the specified destination, with `time-to-live` values starting at 1 and increasing in steps of `0, 0, 1, 0, 0, 1, ...`. The `time-to-live` value in the packet is decreased by one at each router along the path, so that these packets expire at a succession of the intermediate points along the way, when a router detects a `time-to-live` value of zero. When this happens, the router in question sends a packet back to the source, and these returned packets are used to prepare a report, for the user.

In this experiment, traceroute was used from a computer connected to a modem:

```
[root@lynx2 /root]# traceroute www.apple.com
traceroute to www.apple.com (17.254.0.91), 30 hops max, 38 byte packets
 1  139.86.23.1 (139.86.23.1)  129.090 ms  119.520 ms  129.724 ms
 2  139.86.24.1 (139.86.24.1)  139.511 ms  129.586 ms  129.728 ms
 3  usq-gw.usq.edu.au (139.86.128.1)  119.544 ms  129.424 ms  119.713 ms
 4  usq.questnet.net.au (203.22.86.33)  139.438 ms  119.712 ms  149.868 ms
 5  border.questnet.net.au (203.22.86.242)  140.538 ms  119.731 ms  109.960 ms
 6  ATM9-0-0-5.ia3.optus.net.au (192.65.88.209)  140.222 ms  139.931 ms  129.874 ms
 7  GigEth12-0-0.rr2.optus.net.au (202.139.191.22)  149.981 ms  129.760 ms  139.908 ms
 8  bcr2-serial6-1-0-0.Sydney.cw.net (166.63.225.165)  389.981 ms  379.920 ms  379.842 ms
 9  208.172.35.189 (208.172.35.189)  379.940 ms  379.890 ms  390.072 ms
10  acr2-loopback.SanFranciscosfd.cw.net (206.24.210.62)  389.759 ms  379.848 ms  399.937 ms
11  * internap-network-services.SanFranciscosfd.cw.net (206.24.209.138)  390.038 ms  379.872 ms
12  border10.ge3-0-bbnet2.sfj.pnap.net (216.52.0.78)  379.926 ms  389.856 ms  389.900 ms
13  apple-3.border10.sfj.pnap.net (63.251.231.170)  399.943 ms  399.877 ms  379.919 ms
14  tre.apple.com (205.180.175.29)  419.932 ms  399.878 ms  419.880 ms
```

The big jump in delay measurements occurs between the nodes `GigEth12-0-0.rr2.optus.net.au` and `bcr2-serial6-1-0-0.Sydney.cw.net`, so we can conclude that these nodes occur at either end of the cross-Pacific link. The increase in ping-times is about 240 milliseconds, which is consistent with the ping times in the earlier experiment.

Now, earlier, the time it should take for light to travel from Sydney to San Francisco (or Los Angeles) was calculated as 37 milliseconds, so that the round trip time should be 74 milliseconds – but our experiments find it to be, not 74 milliseconds but rather 240 milliseconds, a discrepancy of 3 $\times$. What can be the explanation?

A simple explanation which fits the facts is as follows. The path from Sydney to San Francisco, taken by this signal, is not a straight line, and not a great circle. The transmission medium, an optical fiber, sits on, or very nearly on, the sea floor. The sea floor is not flat. A factor of 3 between the great circle path and the sea bed path might seem rather higher than expected, however it is to be expected that the sea floor has a *fractal* shape and it can be shown that the factor of 3 is consistent with the path across the sea floor having a fractal dimension of near 1.5, which is not an unreasonable value.

The path from Australia to the United States may also deviate significantly from the great circle path not just in altitude (height above or below sea level), but also in the direction parallel to the surface to the sea. The path could well go via Japan and Antarctica (although that would seem to be an unlikely choice).

Another factor which will bear on the propagation delay across any large distance is the need for repeaters. Repeaters receive the signal, and retransmit it along the next segment of the path. The purpose of a repeater is to restore the signal shape to something close to the ideal value, and thereby avoid transmission errors interfering with the end-to-end transmission. Since a repeater must momentarily store the signal and then re-transmit it, there will be a delay induced by a repeater, however this is likely to be quite small. □

Exercise 3.2. Using Ping to display Transmission Delay

Use the ping command with a variety of packet sizes (e.g., 56 bytes up to 1500 bytes in increments of about 100 bytes) to display the effect of packet length on transmission delay. Ping the nearest Internet node which responds to a ping packet. The IP address of this node can be found by consulting the routing table for your computer (`route -n`), or by using a `traceroute` command.

Plot your results and draw appropriate conclusions, for example in relation to the impact of packet length on total delay, and the breakdown of delay into its various components. □

3.2.2 The Causes of Loss

When a packet arrives at a node where the outgoing route is busy, it must either be buffered or discarded. It is also possible, in some circumstances, that the processor which deals with routing is overloaded, in which case packets may be discarded as part of the load control strategy. However, the main cause of packets being lost is likely to be the situation where an outgoing link is busy and has been busy for sufficiently long that the buffer for incoming packets is full, and so some packets must be discarded. In such a situation, a variety of strategies are possible: new packets arriving could be discarded; the packets at the head of the buffer (the oldest packets in the buffer) could be discarded; the lowest priority packets in the buffer could be discarded; or, and this is an important approach used in ATM networks, and in the Internet under some current proposals (see Subsection 3.5.5), packets distinguished by the value of a *loss priority bit* (or a certain *Type of Service* bit) could be discarded.

Some classes of traffic are more sensitive to loss than others. For example, lost packets forming part of a file transfer will have to be recovered or the complete transfer will be useless. However, so long as loss is kept at modest levels, which can be ensured by regularly monitoring traffic levels and upgrading links at appropriate times, lost packets will be recovered by a higher-level protocol, and so, will not cause excessive difficulties for network users.

Packet losses at significant levels is currently considered normal in the Internet and packet losses are used to provide feedback to the source from the network concerning the current level of congestion. Packet losses cause sources in a TCP connection to back off from their current traffic levels. For the entire history of the Internet, up to and including the present, it is uncommon for an end-to-end connection to be able to support the highest possible rate of communication feasible for the source and destination. Hence, some feedback from the network to each host is necessary to indicate a limit on how quickly packets should be sent. This feedback is largely provided, at present, by fact that when a router or link along the path of communication becomes congested, packets will be lost, and the source will discover these losses indirectly because the TCP protocol includes acknowledgements for received packets. The losses cause missing acknowledgements and these are then interpreted, usually correctly, at the source, to be a sign of packet losses.

Because the TCP protocol interprets losses as a sign that it should slow up the rate of the source in a TCP connection, deliberate losses can be used as a sign from the network to a source that it should slow the rate of a source. This is one interpretation of the *Random Early packet Discard* technique, which has been introduced into some router buffer control algorithms [6].

3.3 Traffic Models

3.3.1 Randomness

In reality, traffic *fluctuates* year to year, day to day, hour to hour, minute to minute, second to second, millisecond to millisecond, microsecond to microsecond, and nanosecond to nanosecond. This is not a trite comment, though, it should be admitted, that nanosecond to nanosecond variations are not likely to cause too many problems for anyone. *When measurements of traffic (level of activity) in today's networks are made, it is found that significant random variation occurs at virtually every time scale (years, days, hours, etc) [7].*

Mathematical models of traffic which have been widely accepted as realistic in application to communication networks for decades do not have this property, but instead have the property that random variation becomes less and less significant as the time scale lengthens. New, different models of traffic are required to adequately capture this natural and completely genuine property of real traffic. However, before we consider these new models, let's review some of the old models, because it would be misleading to suggest that the old models are no longer relevant.

3.3.2 Poisson Traffic

The Poisson process, or traffic model, has been used since the first studies of traffic, which were originated by Erlang early in the twentieth century, at the time when telephone networks first began to grow. As the name implies, this process is also associated with the French mathematician Siméon Denis Poisson, who lived in the 18th and 19th centuries.

The Poisson process is a *point process* , by which it is meant that any *realization* of a Poisson process is a series of points, on a line. A *Poisson process* is not just a series of points, though, it is a *random* series of points.

An example might be, the successive times when a bird lands on a certain bench in my garden; the succession of times when a bird flies away from that bench; the succession of times when a child is born, or when a person dies, or when an photon is absorbed by an atom, or the succession of times when a volcano erupts on the earth. These examples illustrate that point processes can be set in a variety of contexts, and in particular, at a variety of *time scales*.

The point process we will be interested in are the arrivals of packets at a node, or the arrivals of new telephone calls at a telephone exchange. The equipment, or the building, to which all the telephones in a neighbourhood are connected is called a local telephone exchange. Also, the switching equipment which is used to connect calls from one local exchange to another, or from any exchange to any other type of exchange, is also known as a telephone exchange. These devices are nowadays also called *switches*.

The Poisson process can be characterized in several different ways, and it is useful to know these different ways and to understand their equivalence because this makes it easy to recognize a Poisson process when one arises.

First Characterization of the Poisson Process

The first characterization is, informally, that this is *the point process in which the points occur at random points in time in a manner completely homogeneous with respect to time*.

More formally, let us suppose that we are considering a process on the time interval $[0,T]$. Now let us divide this time interval into *small* intervals of time of length δ_t. In each interval, we conduct an experiment, in which we choose randomly whether a point occurs in this interval. The precise location of the point does not matter, however, in order to be precise, let us say that it lies in the middle of the interval. The probability that a point *is chosen to occur* in a specific interval is $p = \lambda\delta_t$ for every interval, where λ is a certain constant.

The process generated in this way is not exactly the same as a Poisson process, but it is very close to one, and it becomes more and more similar to a Poisson process as $\delta_t \to 0$. This way of generating a point process could be thought of as a method of *simulating* a Poisson process.

Second Characterization of the Poisson Process

The second characterization is more direct, although it doesn't help us to understand how Poisson processes arise naturally in so many situations. Again, we consider a Poisson process on the interval $[0,T]$.

> *The Poisson process is a point process in which the number of points, N, occurring in any interval, $[t_1,t_2]$, $0 \le t_1 \le t_2 \le T$, has the Poisson distribution:*
>
> $$P\{N=k\} = \frac{(\lambda(t_2-t_1))^k}{k!} e^{-\lambda(t_2-t_1)}, \qquad k = 0,1,\ldots. \tag{3.12}$$
>
> *Furthermore, the distribution of the number of points in one interval is independent from the distribution of the number of points occurring in any other interval, so long as the two intervals do not overlap.*

The condition concerning independence of the distributions of points in non-intersecting intervals can probably be derived from the first condition, that the distribution is Poisson with parameter $\lambda(t_2-t_1)$.

Third Characterization of the Poisson Process

The third characterization is a refinement of the first one, but this time there is no limit required. The Poisson process can be characterized as the only process which satisfies the following three conditions.

First we need to introduce (or refresh the memory of) an important mathematical notation: the "big O notation".

Definition 3.3 *Consider a sequence of numbers $a_1, a_2, \ldots$. We say a_k is $O(1/k)$ as $k \to \infty$ if there exists a constant $C > 0$ and an integer $K > 0$ such that for all $k > K$, $|a_k| \le C \times \frac{1}{k}$. The same notation can be used with any function in place of $\frac{1}{k}$. For example, to say that $a_1, a_2, \ldots$ is $O(k^{-1.5})$ means that there exists a constant $C > 0$ and an integer $K > 0$ such that for all $k > K$, $|a_k| < C \times k^{-1.5}$. Also, this notation can be used for a collection of numbers indexed by a continous real valued variable, such as time, or an interval of time, rather than by an integer, and the index variable may tend to values other than ∞. For example, if we say that $|f(t)| < O(t^2)$ as $t \to 0$, it means that there exists C and t_0 such that for all $t < t_0$, $|f(t)| < Ct^2$.*

Denote the number of points occurring in the interval $[s,t]$ by $N([s,t])$; then

(i) $P(N[t,t+\delta_t] = 1) = \lambda\delta_t + O(\delta_t^2)$ as $\delta_t \to 0$;

(ii) $P(N[t,t+\delta_t] > 1) \le O(\delta_t^2)$ as $\delta_t \to 0$;

(iii) $N([s_1,t_1])$ and $N([s_2,t_2])$ are independent whenever $[s_1,t_1]$ and $[s_2,t_2]$ do not intersect.

The first two conditions exclude the possibility that points should occur in clusters, at the same point in time, and they specify that the probability of a point occurring in an short interval is the same anywhere on the line, and is proportional to the length of the interval. The constant in this proportionality relationship is λ, which can be interpreted as the rate of occurrence of points.

Fourth Characterization of the Poisson Process

Choose a sequence of independent random numbers, $\{U_k\}_{k\in\mathbb{N}}$ drawn from a negative-exponential distribution with mean λ, that is to say

$$P(U_k \ge t) = e^{-\lambda t}, \qquad t \ge 0.$$

Now use these numbers as follows to select the positions of the points within the interval $[0,T]$. The first point is chosen to lie at U_1, the second at U_1+U_2, and so on, stopping when a selected point lies outside the interval $[0,T]$. This collection of points is also a Poisson process.

Proof of Equivalence*

Let us prove the equivalence of these four characterizations by showing that the first implies the second, the second implies the third, the third implies the fourth, and then the fourth implies the third, third the second, and, finally, the second implies the first.

Please note: This subsection should be omitted at a first reading is included for the interested reader, rather than as an essential peice of knowledge for all readers. The ideas pursued in this subsection are of historical interest, and are potentially important for the reader who values a thorough understanding of the subject.

Proof that the First Characterization Implies the Second

It is probably useful to start by discussing two classically important probability distributions: the Bernoulli distribution, and the Binomial distribution. Each of these distributions is associated with a certain *experiment*.

In the Bernoulli experiment (also known as a Bernoulli trial), a coin is tossed, and the experiment is declared to be a success if a head is the result. We suppose that the head of the coin comes up with probability p. Tosses in which a head is the result are also regarded as *successes* and we also associate a *value* with the outcome, 1 for a head and 0 for a tail. So, the successful experiment produces the *value* 1, while the other outcome produces a zero. This is the Bernoulli distribution.

The Binomial experiment is a series of n Bernoulli trials, each with probability of success 1. Each trial produces the value 1 or 0, and these are *added together*. The distribution of the resulting random variable, N say, is as follows:

$$P(N=k) = \binom{n}{k} p^k (1-p)^{N-k}.$$

Now we are ready to discuss the proof of the equivalence of the characterizations.

The first condition is stated as a limit, so to prove the equivalence we must show that the probability distributions discussed in the first characterization converge to the distributions described in the second.

Consider an interval $[s,t] \subseteq [0,T]$. In the first characterization, we subdivide this interval into sub-intervals of length δ_t, and choose, randomly, whether a point occurs in each interval or not, according to the probability $\lambda\delta_t$. The distribution of the number of points in the interval $[s,t]$ according to this approach must be the Binomial distribution with parameters $n = T/\delta_t$ and $p = \lambda\delta_t$. The distribution of $N([s,t])$ is therefore

$$P(N([s,t]) = k) = \binom{n}{k} p^k (1-p)^{N-k}, \tag{3.13}$$

where $p = \lambda\delta_t$.

Now let us consider what happens as we let $\delta_t \to 0$. We expect that (3.13) $\Longrightarrow$ (3.12), but how can we show this?

Now $\binom{n}{k} = \frac{n!}{(n-k)!k!} = \frac{n\times(n-1)\times\ldots(n-k+1)}{k!} \approx \frac{n^k}{k!}$ for large n. Since $n = T/\delta_t$, n is going to become very large as $\delta_t \to 0$. Taking into account that $p = \lambda\delta_t$, we find that, for δ_t sufficiently close to zero,

$$\begin{aligned}
\binom{n}{k} p^k (1-p)^{n-k} &\approx \frac{n^k}{k!}(\lambda\delta_t)^k (1-p)^{n-k} \\
&= \frac{(n\lambda\delta_t)^k}{k!}(1-\frac{\lambda T}{n})^{n-k} \\
&\to \frac{(\lambda T)^k}{k!} e^{-\lambda T},
\end{aligned}$$

as $\delta_t \to 0$.

This last limit comes from a well-known classical result for the exponential function:

$$e^{-a} = \lim_{n\to\infty}(1-\frac{a}{n})^n.$$

The independence of the distribution of points in non-overlapping intervals follows readily from the first characterization also. This completes the demonstration of First $\Longrightarrow$Second.

Proof that the Second Characterization Implies the Third

In the second characterization we are given the distribution of the points arriving in each interval whereas in the third characterization, we are just told the limiting behaviour, that for small intervals the probability of no arrivals is almost one, for one arrival it is λ times the length of the interval, and for more than one arrival, the probability is small, and reducing in proportion to δ_t^2 *at least*. To prove that the second characterization implies the third, we suppose that the second holds and prove that the third must also hold.

Now if the second characterization holds

$$P(N([t,t+\delta_t]=1)=\lambda\delta_t e^{-\lambda\delta_t}\rightarrow\lambda\delta_t$$

as $\delta_t\rightarrow 0$, which shows (i);

$$P(N([t,t+\delta_t]>1)=e^{-\lambda\delta_t}\sum_{k=2}^{\infty}\frac{(\lambda\delta_t)^k}{k!}\leq(\lambda\delta_t)^2\rightarrow 1$$

as $\delta_t\rightarrow 0$, which shows (ii), and, finally, (iii) is also assumed in the second characterization.

Proof that the Third Characterization Implies the Fourth

According to the third characterization, the number of points in successive intervals are independent and the probability of no points at all lying in an interval of length δ_t is $1-\lambda\delta_t+O\left(\delta_t^2\right)$, so the probability of no points at all in an interval of length τ must be

$$\lim_{\delta_t\rightarrow 0}(1-\lambda\delta_t)^{\tau/\delta_t}+O\left(\frac{\tau}{\delta_t}(\delta_t)^2\right)=e^{-\lambda\tau}. \tag{3.14}$$

Proof that the Fourth Characterization Implies the Third

This proof relies primarily on a famous and important property of the negative-exponential distribution, the so-called *memoryless* property of this distribution, which can be expressed thus. Suppose X is a negative-exponential distribution with mean $1/\lambda$. Then

$$P(X>t|X>\tau)=P(X-\tau>t)=e^{-\lambda(t-\tau)},\qquad t>\tau,\tau\in\mathbb{R}. \tag{3.15}$$

Now it might seem a long way from a knowledge of the distribution of the *gaps* between points to the distribution of the *number* of points in every interval, however, this is apparently not the case.

Let us divide the interval $[0,T]$ into sub-intervals of length δ_t. Now consider a specific such sub-interval, $[t,t+\delta_t]$. Now let us consider the possibility that a point, or more than one point, occurs in this sub-interval. Wherever the previous point occurs, or even if there is no previous point, by the memoryless property, (3.15), the distribution of distance from t of the *next* point after time t is the negative-exponential distribution, and therefore

$$P\left(N([t,t+\delta_t])\geq 1\right)=1-e^{-\lambda\delta_t}=\lambda\delta_t+O(\delta_t^2)$$

and

$$P\left(N([t,t+\delta_t])\geq 2\right)\leq\left(1-e^{-\lambda\delta_t}\right)^2=O(\delta_t^2).$$

The independence of the number of points in one sub-interval from the number in another disjoint sub-interval is clear, under the fourth characterization of the Poisson process. This shows that the fourth characterization implies the third.

Proof that the Third Characterization Implies the Second

We have already seen that when the third characterization applies to a process, the probability that an interval of length τ contains *no* points is $e^{-\lambda\tau}$. Independence of the count in one interval and another, for disjoint intervals, is assumed in both characterizations. So it only remains to show that the probability distributions of the count in an interval take the same values for counts greater than zero.

Consider an interval of length τ and further sub-divide this interval into sub-intervals of length δ_t, and let $n = T/\delta_t$. Since according to the third characterization, the probability that one point occurs in an interval of length δ_t is $\lambda\delta_t + O((\lambda\delta_t)^2)$ and the probability that more than one point occurs in this interval is $O((\lambda\delta_t)^2)$, the third characterization implies that

$$\begin{aligned} P(N([t,t+\tau]) = k) &= \binom{n}{k} (\lambda\delta_t)^k (1-\lambda\delta_t)^{n-k} + O\left((\lambda\delta_t)^{k+1}\right) \\ &\approx \frac{(n\lambda\delta_t)^k}{k!}(1-\lambda\delta_t)^{n-k} + O\left((\lambda\delta_t)^{k+1}\right) \\ &= \frac{(\lambda\tau)^k}{k!}(1-\lambda\tau/n)^{-k}(1-\lambda\tau/n)^n + O\left((\lambda\delta_t)^{k+1}\right) \\ &\to \frac{(\lambda\tau)^k}{k!}e^{-\lambda\tau} \end{aligned}$$

as $\delta_t \to 0$, which concludes the proof of this case.

Proof that the Second Characterization Implies the First

We have already shown that if a process is "simulated" by choosing whether points occur in the intervals of length δ_t into which the interval $[0,T]$ is sub-divided then the distribution of the number of points in *any* sub-interval, $[t,t+\tau]$ is Poisson with mean $\lambda\tau$. Now suppose, conversely, that the second characterization holds, with the mean number of points in any interval of length τ again $\lambda\tau$. Is it the case that this process is the same as one which was simulated as in the first characterization? Why, of course it is! Because if such a procedure was adopted, it would again, necessarily, produce a process in which the distribution of the number of points in any sub-interval of length τ would be Poisson with mean $\lambda\tau$.

This concludes the proof that all four characterizations are equivalent as regards the Poisson process on the interval $[0,T]$. A Poisson process on the whole real line can be envisioned as a collection of these $[0,T]$ Poisson processes pasted together, with each interval containing a Poisson dietributed number of points.

3.3.3 Telephone Traffic

The collection of *arrival times* of telephone calls at a telephone exchange (or elsewhere) are generally assumed, and found, to form a Poisson process. The increasing amount of use of the telephone for calls to Internet Service Providers (ISPs) has caused a change in characteristics of telephone usage to a degree, but the Poisson process is still the best model we have for the arrivals of telephone calls.

But, what about the *traffic generated* by these calls (i.e. the traffic in bits)? Each call typically lasts 120-180 seconds (depending on the proportion of those rather longer calls to ISPs). The duration of a telephone call is traditionally called its *holding time*. It is traditionally assumed that holding times are negative-exponentially distributed, however this assumption will not actually be necessary anywhere in this text and has become an increasingly dubious assumption over recent times (because of the calls to ISPs!).

If the *arrival rate* of calls is λ and the average holding time of these calls is h, the average number of active calls is given by the formula $a = \lambda h$. The unit in which telephone traffic (a) is measured is traditionally called an *Erlang*. Thus, when it is said that 8 Erlangs of telephone traffic are occupying a certain resource, it should be taken that *on average* there are 8 calls making use of that resource.

Typically, each call requires 64 kbit/s of bandwidth in a digital transmission system. Thus, we must multiply the Erlangs of traffic by 64 in order to measure the load induced on a transmission system in kbit/s. Note that there are systems in widespread use which convert a telephone call into a lower bit-rate also. For example, digital mobile phones use encodings at rates in the vicinity of 10-16 kbit/s and voice over IP systems often use 16 kbit/s during active periods.

When voice signals are transmitted over traditional *circuit-switched* networks, a clear channel with a consistent bandwidth is usually required, typically 64 kbit/s, in both directions. When voice is transmitted over a packet network, including an IP network, it is normal to suppress the silent periods. Since people do not usually talk at once over the telephone, the silent periods typically occupy more than half the duration of the call. Hence a further

reduction of approximately 55% in required capacity ensures in the case of voice over IP. It follows that a voice over IP system can be expected to require approximately one eigth of the bandwidth of a traditional circuit-switched voice network.

The number of calls active at any moment in time has a Poisson distribution with mean given by the traffic, in Erlangs. It is not immediately obvious why this should be the case, unless the holding times were all constant. If the holding times were all a fixed time, for example 3 minutes, the calls active at a certain time would be exactly those ones which arrived in a certain 3 minute period, which we know is Poisson distributed. For a more general distribution of holding times, the Poisson distribution must still apply, since we can view this case as a probabilistic mixture of the fixed holding times case.

Since the variance of a Poisson distribution is the same as its mean, the variance of the transmission rate of a bit-stream formed from a telephone stream of a Erlangs will be $64^2 \times a$ in $(kbit/s)^2$ and its standard deviation must therefore be $64 \times \sqrt{a}$ kbit/s.

In the case of a voice over IP system using a 16 kbit/s encoding rate, this formula needs to be modified on two accounts: First we need to replace the 64 by a 16 because of the different encoding rate. This is simple enough. Secondly, we need to take into account the fact that each call now generates a succession of *talk-spurts*. Assuming an activity rate of 0.45, the average number of active talk-spurts will be $a \times 0.45$. Because the distribution of calls is Poisson distributed, and each call is independently in a talk-spurt with probability 0.45, the distribution of the number of active talk-spurts will also be Poisson distributed, with mean $0.45a$. Hence the mean rate of a voice over IP bit stream of a Erlangs will be $16 \times 0.45a$ and the standard deviation of the rate will be $16 \times \sqrt{0.45a}$.

It is quite important to know both the mean and variance, or standard deviation, or all the traffics we deal with. The variance, V say, and the standard deviation, s say, of any random variable are related by the formula $V = s^2$. Standard deviations are in some respects a little more natural to deal with since they are measured in the same units as the mean and the same units as the quantity being measured, whereas variances are measured in this unit's square.

3.3.4 Gaussian Traffic

Now let us consider quite a different type of process. When the Poisson process is used, it is used to describe the arrival times, or departure times, of calls, packets, messages, and so on. When we use the Gaussian model, we use it instead to describe a *quantity of work* or a *number of bytes, or cells, or packets* which have arrived during a certain interval of time.

The most flexible framework is to consider the amount of work (in bytes, or cells, or packets) arriving in the interval $[0,t]$. Let us denote this quantity by Y_t, for $t > 0$. Both t and Y_t are allowed to take *real* values, positive or negative, although typically we would limit t to be positive or zero. Actually, it makes sense to limit Y_t to be positive or zero also, however there is a technical reason for not imposing this constraint.

In the Gaussian model, Y_t is assumed to have a Gaussian distribution with mean μ_t and variance $v(t)$, for all $t \in \mathbb{R}$. We also normally assume that the distribution of the work arriving in *any* interval is the same no matter where this interval occurs, e.g. if the interval is $[s,t]$ the quantity of work is $Y_t - Y_s$, and this has the same distribution as the quantity Y_{t-s}. It follows that, $\mu_t = \mu t$, for a certain constant μ. That is to say, the mean amount of work arriving in any interval of length t is $\mu \times t$.

Note: the process Y_t is not stationary. The mean of this process increases linearly with t and the variance of Y_t is also changing with t. However, there is an underlying process which *is* stationary. For example, the process $\{Y_t - Y_{t-1}\}_{t>0}$ is stationary.

The definition is not complete yet. We also require that for any collection of times, t_1, t_2, ..., the random variables Y_{t_1}, Y_{t_2}, ..., are *jointly* normally (Gaussian) distributed. It is not sufficient to require that each of the random variables Y_t has a Gaussian distribution individually (in the univariate sense) to imply that the they are *jointly* Gaussian.

How can we possibly ever verify such a condition? Fortunately, this will not be necessary. The main reason for adopting the Gaussian model of traffic is that we expect network traffic to become more and more Gaussian as traffic is aggregated, simply by networks becoming larger, and carrying a larger number of indpendent traffic streams. This is discussed further in Subsection 3.4.3.

Negative Traffic

One of the weaknesses of the Gaussian model which some people feel is quite disturbing is that it allows for the possibility of negative traffic. What can be the meaning of negative traffic? One possibility is that it represents an *ability to do work*, however it is difficult to see how an ability to do work can *arrive* on the same communication channel as the work which needs to be done!

The simplest way to rationalize this problem of negative traffic is to notice that for many realistic Gaussian models of traffic, although there is a non-zero probability that traffic arriving in a short interval could be negative, this probability is very small.

Characterizing Gaussian Traffic

The Poisson process is characterized by *one parameter*: λ, which is the average rate of arrival of points. In the case of the Gaussian process, there is a rich collection of parameters. First of all we have the mean amount of work arriving per unit time: μ, then there is the variance-time curve: $v(t) = \text{Var}(Y_t)$. These two, μ and $v(t)$ are sufficient to characterize the Gaussian process Y_t completely. It is this last "parameter", $v(t)$, which gives a great deal of flexibility to the Gaussian model. This function cannot be selected arbitrarily. For example, it cannot take negative values!

It might seem likely that $v(t)$ should always be increasing, however it is not necessarily the case; in fact we can even have $v(t) = 0$ for positive values of t. It is possible to define the restrictions on the function $v(t)$ precisely, however this would take us down a very technical line of argument without sufficient benefit in understanding or application to network analysis and design.

One simple observation which can be easily made, though, is that if there exists any process, $\{\widetilde{Y}_t\}$, not necessarily Gaussian, which has the variance time curve $v(t)$, then there is also a Gaussian process which has this same variance time curve.

Example 3.4. A Simple Gaussian traffic model

Consider the model in which $\mu = 1$ and

$$v(t) = t^2, \qquad t > 0.$$

Thus, the mean and standard deviation of Y_1, the amount of traffic arriving in the interval $[0,1]$, is 1, 1 respectively. Similarly, the mean and standard deviation of the traffic arriving in the interval $[0,2]$ is 2, 2 respectively. This pattern continues for intervals of any length.

Consider the two adjacent intervals $[0,1]$ and $[1,2]$. The traffic arriving in each interval has the same mean, 1, and the same standard error, 1, and the standard error of the *sum* of these two traffics is 2. Now, by (3.4),

$$4 = \text{Var}(Y_2) = \text{Var}(Y_1 + Y_{[1,2]}) = 1 + 2\text{Cov}(Y_1, Y_{[1,2]}) + 1,$$

so $\text{Cov}(Y_1, Y_{[1,2]}) = 1$. The maximum value that $\text{Cov}(X,Y)$ can take, for any random variables X, Y, is $\sqrt{\text{Var}(X)\text{Var}(Y)}$ (see (3.5)), and in the case of Gaussian random variables, this can only happen if $X = Y$, i.e. no matter what value X takes, Y always takes exactly the same value.

Thus, for this particular choice of mean and variance-time curve, the traffic process takes a random value, at the first time when it is observed, and henceforth the traffic continues to flow at exactly the same rate forever more. □

Smooth Gaussian Processes

The process Y_t does not necessarily have a first derivative, however there is a large class of processes where it does. We shall denote this process, typically, by X_t, so $X_t = \frac{d}{dt}Y_t$. It is called the *rate process* for the process Y_t. When it exists, this process must *also* be Gaussian.

Exercise 3.3. The Rate of a Gaussian Process

Consider Example 3.4. Does this Gaussian process have a rate? If so, what is the mean and variance of the rate process in this case? □

It is not necessarily obvious whether or not a Gaussian model for real traffic *should* be smooth. Here are some arguments for and against the smoothness of real traffic:

Argument against smoothness: Many studies have observed that the variance-time curve of real traffic takes the form

$$v(t) = Ct^{2H}, \qquad t > 0,$$

and this particular variance-time curve is known to lead to a Gaussian process which *does not* have a rate process.

Argument for smoothness: Real traffic flows through rate-limited communication facilities from the point of origin. Even the source of a packet which will eventually pass through the network under study is generated bit-by-bit, and then must pass through the network interface of the computer which houses this source. Every transmission facility has a defined and finite rate at which it is capable of transmitting, and therefore the traffic flowing through our networks also has a *maximum rate* at which traffic may flow. It follows that real traffic does have a rate.

This is a strong argument in favour of adopting the assumption that real traffic has a rate, however it should be kept in mind that as time passes the limitations on the maximum rate at which sources of traffic can communicate is increasing.

3.3.5 Long-range Dependence

Traffic measurements reported in [7] and in many other papers have shown that significant statistical correlation exists between traffic levels separated by long intervals of time, no matter how long the separating interval. In fact, we can be more precise.

Let us denote the traffic *rate* process by $X(t)$, where t is a positive real number. Denote the correlation of $X(0)$ and $X(t)$ by $\rho(t)$, that is to say, $\rho(t) = \frac{E((X(0)-\mu)(X(t)-\mu))}{E((X(0)-\mu)^2)}$, where $\mu = E(X(0))$. Then, real traffic has the somewhat surprising property that

$$\int_0^\infty |\rho(t)|dt = \infty.$$

Traffic with this property is referred to as *long range dependent*.

But, before we get too involved in this issue, important as it is, let's just think about how we might go about measuring and quantifying traffic. Since traffic is random, we must use ideas from statistics. What are the basic statistics we use for any random quantity? They are the *mean* (average), usually denoted by m, or μ, and the *standard deviation*, usually denoted by σ. A closely related quantity is the *variance*, $V = \sigma^2$.

Before we can use these statistics to quantify traffic we need to select a *sampling interval* – otherwise it is not clear how these quantities are defined. Suppose we select 10 milliseconds as the sampling interval, and we are measuring the quantity of traffic offered on a transmission link with speed 10 Mbit/s. If the transmission link was completely occupied by genuine data (not just a synchronization pattern being sent end-to-end), the quantity of data in a 10 millisecond interval would be $0.01 \times 10,000,000 = 100,000$ bits, or 12,500 bytes.

So, if the transmission link was *half full*, there would be 6,250 bytes being carried on the transmission link, on average, every 10 milliseconds. If we change our sampling interval to 100 milliseconds, it is clear that the average number of bytes transmitted will also increase by a factor of 10, to 62,500 bytes. If we use a sampling interval of 1 second, the number of bytes will increase by 10 again, to 625,000 bytes. It makes sense, since we have this simple linear relationship between the sampling interval and the mean bytes carried in that sampling interval, to speak, instead, of the mean *bytes per second*, to quantify the rate at which bits are being transmitted rather than the quantity transmitted in a certain interval of time. In the present case, the mean traffic being carried would be 625,000 bytes per second.

Now lets consider the second important statistic: the standard deviation. Again, let us start by using a sampling interval of 10 milliseconds. If the number of bytes in this interval was Poisson distributed, the variance of the number of bytes would be 6,250, and so the standard deviation would therefore be about 80 bytes. However, we know that there are upper and lower limits of 0 bytes and 12,500 bytes on the possible values for the bytes carried on this transmission system. So the standard deviation could be as high as 6,250 (in the extreme case where the number of bytes is always either 0 or 12,500). A value somewhere in-between these extremes of 80 and 6,250 is more likely, e.g. 1000 bytes.

Let us suppose that the standard deviation is about 1000 bytes, for the number of bytes arriving in a 10 millisecond interval. What is likely to happen in a longer sampling interval, for example, an interval of 100 milliseconds? One possibility is the case where successive intervals are statistically independent. In this case, the variance of

the longer interval can be obtained as the sum of the variances of the smaller intervals, and hence the standard deviation of the bytes carried in the longer interval would be $1000 \times \sqrt{10}$. At the other extreme, the successive intervals might be almost perfectly correlated. In this case, the standard deviation would be $1000 \times 10 = 10,000$.

In the former case, the variance of the traffic arriving in a time interval of length t would follow the law

$$V(t) = \sigma_1^2 t$$

and in the latter case, it would follow the law

$$V(t) = \sigma_1^2 t^2.$$

Which is closer to the truth? Are successive time intervals heavily correlated or are they statistically independent? This question has been thoroughly investigated and the answer is that successive intervals are heavily correlated, but not quite so heavily that they are totally correlated. Real traffic seems to follow the law:

$$V(t) = \sigma_1^2 t^{2H} \tag{3.16}$$

where H is called the *Hurst parameter*, with typical values around 0.7-0.8.

The precise value of H varies from place to place and time to time, although values near 0.8 are quite common. The law (3.16) holds for a wide range of values of t. An open question here is whether this law holds for smaller and smaller t. If the process has a rate, for example, a different law must apply for small values of t:

$$V(t) = \sigma_0^2 t^2$$

that is to say, at very short time scales, we would expect traffic to become totally correlated. Whether something like this really occurs at short time intervals does not seem to have been verified by experimental observation.

Two models which exhibit long-range dependence stand out as particularly important [8]: the Poisson-Pareto Burst Process (PPBP) model (also known as the M/Pareto model), and the Gaussian traffic model with a heavy tailed autocorrelation function, notably the *Fractional Brownian Motion* (FBM) process (which is a special type of Gaussian process)..

3.3.6 Fractional Brownian Motion and Fractional Gaussian Noise

Fractional Brownian Motion (FBM) is the name given to the Gaussian process with the variance-time curve

$$V(t) = \sigma_1^2 t^{2H}$$

for some H, $0 < H < 1$. The case $H = 0.5$ is a special case of great importance, namely *Brownian Motion*.

For no value of H does FBM have smooth paths, so there is no rate process. Nevertheless, it is convenient to pretend that there is a rate process, and we give this process the name *Fractional Gaussian Noise* (FGN). In the special case where $H = 0.5$, this process is known as *white noise*. There is no special trick involved here: we just use the concept of FGN as a way of talking about FBM.

Fractional Brownian Motion has been studied extensively and has frequently been proposed as a model of network traffic. We can use FBM to model the amount of traffic which has arrived since some point in the past, whereas in the case of FGN, we think of the model as representing the traffic arriving at each point in time.

3.3.7 The Poisson-Pareto Burst Process

One of the nice features of the Poisson-Pareto Burst Process is that it is corresponds in a simple way with a simple and natural view of what is happening in a real network.

According to this model, the traffic load at any one time is formed from the superposition of a number of bursts. There is no limitation as to how many of these bursts might be active at the one time. All the bursts are completely independent, in the statistical sense. The starting points, in time, for all the bursts together form a Poisson process.

Next we need to consider the statistics of the *length* of the bursts. There seems to be good evidence that the distribution of the lengths of the bursts, in a wide variety of observed traffic, has a *Pareto* distribution:

$$P\{B > t\} = \begin{cases} \left(\frac{t}{\delta}\right)^{-\gamma}, & t \geq \delta \\ 1 & \text{otherwise,} \end{cases}$$

in which $1 < \gamma < 2$ and $\delta > 0$. The parameter γ here is related to the Hurst parameter, H, by the formula $H = \frac{3-\gamma}{2}$.

Thus, the parameters which are required to specify this traffic model are:

- λ: the rate of arrival of bursts;
- γ: the decay rate of the Pareto distribution of burst lengths;
- δ: the scale parameter of the Pareto distribution;
- r: the rate of the individual bursts.

Exercise 3.4. The distribution of file sizes

It is not difficult, on a Unix computer at any rate, to collect statistics concerning the sizes of files. Here is a script for collecting the sizes of files on a Unix system:

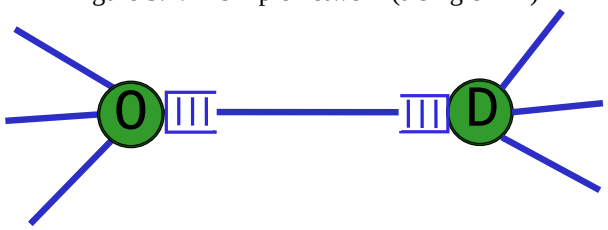
```
#!/bin/bash
# List all file sizes, in all subdirectories of the specified directory
# Store the file sizes in a file called sizelist
find $1 -type f -exec ls -s {} \; | awk '{print $1}' >> sizelist
```

Use this script to collect a list of all the file sizes on a computer and then form a histogram of the file sizes on this computer. Next, using a spreadsheet (or gnuplot or whatever plotting package happen to be familiar with), plot the log of the frequency vs log of file size for this histogram. Is the distribution possibly Pareto in form? Hint: if the curve appears as a straight line when plotted with log scales on both axes then it is Pareto distributed. Use this plot to form a rough estimate of the parameter of a Pareto distribution which could fit this histogram. □

3.4 Application of the Gaussian Traffic Model

In this section we shall apply the Gaussian traffic model to the behaviour of one of the simplest network components which exhibits interesting queueing behaviour: a single link, as depicted in Figure 3.1. If this seems unrealistically simple, keep in mind that real networks may be decomposed into just such elements.

Figure 3.1: A simple network (a single link)



3.4.1 A Simple Link

We shall assume that the traffic process is Gaussian and that there exists a rate process, which we denote by $\{X_t\}_{t\geq 0}$.

Thus, the amount of work arriving in the interval t_1 to t_2 would then be $\int_{t_1}^{t_2} X_t dt$. The cumulative traffic process will be denoted, as usual, by Y_t, so

$$Y_t = \int_0^t X_s ds.$$

As before: the mean of the process X_t, which we shall denote by μ; the variance of X_t: σ_0^2; and the variance-time curve of the process $\{Y_t\}_{t\geq 0}$ by

$$V(t) = \mathrm{Var}(Y_t).$$

As discussed above, there is a lot of evidence that the variance-time process tends to take the form

$$V(t) \approx \sigma_1^2 t^{2H},$$

for moderate and large t.

Thus, altogether, we require the following parameters to characterize a Gaussian process of the sort just discussed:

- μ: the mean rate of arrival of traffic;
- σ_0: the standard deviation of the traffic rate process;
- σ_1: the standard deviation of traffic in a time interval of length 1 (according to the medium and long time scale variance time curve);
- H: the Hurst parameter.

In discussing this example of a single link, in the remainder of this subsection, we shall now suppose that the number of packets the source attempts to deliver on this link in each 10 millisecond interval has a Gaussian distribution with mean and standard deviation as follows (we consider two cases):

(i) $m_p = 5$, $\sigma_p = 2$;

(ii) $m_p = 10$, $\sigma_p = 3.5$.

The packets on a network (an ethernet, the Internet, etc) also vary in length. TCP/IP places a maximum length limit on packets of $2^{16} = 65536$, but packets are often further limited by the underlying network. For example, if the underlying network is *ethernet*, packets cannot be longer than 1500 bytes.

A very simple way to estimate the average length of packets on an ethernet is as follows: there will be a lot of packets which are quite short, nearly zero bytes in length – for example, when it is necessary to *acknowledge* a packet sent from the other end of a TCP connection, but there is no data going in that direction on which the acknowledgement can be piggy-backed. There will also be a lot of packets which are quite long – for example, as generated by applications which have so much data to send that they can easily fill maximum length packets continuously. So, if we simply assume that half of all packets are of zero length, and the other half are full, we obtain an estimate of average packet length as 750 bytes.

What of the standard deviation of the packet length? Using the same idea, that half the packets are of length 0 and the other half are of length 1500, we obtain:

$$\sigma_L^2 = E\{(X-750)^2\} = \frac{1}{2}750^2 + \frac{1}{2}750^2 = 750^2,$$

so $\sigma_L = 750$.

Mean and Variance of Bytes per time interval

We do not need to estimate delay performance with great precision. Therefore, we can ignore issues such as packets which lie across the border between successive 10 millisecond intervals.

We could perhaps ignore completely the variation in traffic quantities engendered by the random variation in packet lengths. However, a better approach is to calculate losses in *bytes* rather than in *packets*. If we denote the mean number of bytes arriving in a 10 millisecond interval by m_b, the mean number of *packets* arriving in the same interval by m_P, and the mean *length* (in bytes) of packets by m_L, we find (by applying (3.7), or by common sense):

$$m_b = m_P * m_L = 5 * 750 = 3750$$

in Case (i) and 7500 in Case (ii).

The *variance* of the number of bytes offered in a 10 millisecond interval is a little more messy to calculate. It is affected by the variance in the number of packets arriving in each interval and also by the variance in the *length* of these packets. The formula for this variance follows from (3.9) and is:

$$\sigma_b^2 = (m_L^2 + \sigma_L^2)\sigma_P^2 + \sigma_L^2 m_P^2 \tag{3.17}$$

and in the present instance this evaluates to

$$\sigma_b^2 = \begin{cases} 2\times 750^2 \times 2^2 + 5^2 \times 750^2, & \text{in Case (i),} \\ 2\times 750^2 \times 3.5^2 + 10^2 \times 750^2, & \text{in Case (ii).} \end{cases}$$

so the standard deviation is $\approx 6\times 750 = 4500$, in Case (i) and $\approx 11\times 750 = 8250$, in Case (ii). Notice that most of the variation is packet lengths has quite a significant impact on the variance of the bits arriving in an interval.

Example 3.5. Statistics of Bytes per packet for Uniform packet lengths

Suppose, by contrast with the previous example, that the distribution of packet length was uniform on the interval $[0,1500]$ rather than all concentrated on 0 and 1500. What would the standard deviation of bytes per packet be then?

The mean of a uniform distribution on $[0,1]$ is 0.5 and the variance of a uniform distribution on $[0,1]$ is

$$\sigma_U^2 = \int_0^1 (x-0.5)^2 dx = \left[\frac{x^3}{3} - \frac{x^2}{2} + 0.25x\right]_0^1 = \frac{1}{12}.$$

Multiplying a uniform random variable on $[0,1]$ by 1500 gives us a random variable which is uniformly distributed on $[0,1500]$. Hence, the mean of such a random variable must be 1500/2=750 and its variance must be $1500^2 \times \frac{1}{12}$, by (3.4). Here, already, is an example where it is more meaningful to use the standard deviation. The standard deviation of the packet length in this case must be $\frac{750}{\sqrt{3}} = 433$, which is little larger than half the standard deviation in the more extreme case previously considered. □

3.4.2 The Normal Loss Function

Given that a quantity, X, is normally distributed, with mean μ_X and standard deviation σ_X, calculating the probability that it lies above a certain level, y, reduces to looking up a table of the normal distribution. We calculate the "Z score", $Z = \frac{y-\mu_X}{\sigma_X}$ and look it up in a table of the normal distribution, e.g. the one in Table 3.2.

However, that is not quite what is called for in the present situation. We have a quantity of *traffic* flowing along a limited pipe, and the *rate* at which this traffic flows varies. We don't want to know how likely it is that the rate exceeds the capacity of the link, we want to know how much traffic is lost *while* the traffic exceeds the capacity of the link.

If the mean of the traffic was zero (a little artificial, but nice for this example), the standard deviation was one, and the capacity of the link was y, the traffic lost would be given by the *normal loss function*, $\mathrm{NL}(y)$, a table of which is given in Table 3.1.

In the general case (which is of much more interest), The traffic lost will be

$$\text{Loss} = \sigma_X \mathrm{NL}\left(\frac{y-\mu_X}{\sigma_X}\right). \tag{3.18}$$

So, we can use a table for a *standard* normal loss function for all cases, by just doing a bit of arithmetic.

Derivation of a Formula for the Normal Loss Function*

It is possible to derive a formula for the normal loss function which is explicit as follows:

$$\begin{aligned} NL(y) &= \int_y^\infty \frac{1}{\sqrt{2\pi}}(x-y)e^{-\frac{x^2}{2}}dx \\ &= \int_y^\infty \frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}xdx - y(1-ND(y)) \\ &= \int_{y^2/2}^\infty \frac{1}{\sqrt{2\pi}}e^{-u}du - y(1-ND(y)) \end{aligned}$$

where $u = \frac{1}{2}x^2$

y	**Loss**
0.0	0.3989
0.1	0.3509
0.2	0.3069
0.3	0.2668
0.4	0.2304
0.5	0.1978
0.6	0.1687
0.7	0.1429
0.8	0.1202
0.9	0.1004
1.0	0.08332
1.5	0.02931
2	0.008491
2.5	0.002004
3	0.0003822
3.5	0.00005848
4	0.000007145
4.5	6.942×10^{-7}
5	5.346×10^{-8}
5.5	3.255×10^{-9}
6	1.56×10^{-10}

Table 3.1: The Normal Loss Function: $E\{Z-y; Z>y\}$

z	$P\{Z>z\}$
0.5	0.3085
1	0.1587
1.5	0.0668
2.0	0.0228
2.5	0.0062
3.0	0.00135
3.5	0.000233
4	0.0000317
4.5	3.398×10^{-6}
5	2.867×10^{-7}
5.5	1.899×10^{-8}
6	9.866×10^{-10}

Table 3.2: The Standard Normal Distribution: $P\{Z>z\}$

$$= \frac{1}{\sqrt{2\pi}}e^{-y^2/2} - y(1-ND(y)), \tag{3.19}$$

in which $ND(y)$ denotes the standard normal distribution.

Now if the normal distribution does not have mean zero and standard deviation one, we can nevertheless express the value of the normal loss function in terms of the *standard* normal loss function as follows:

$$\begin{aligned} NL(y;\mu,\sigma) &= \int_y^\infty \frac{1}{\sqrt{2\pi}\sigma}(x-y)e^{-\frac{(x-\mu)^2}{2\sigma^2}}dx \\ &= \frac{1}{\sqrt{2\pi}\sigma}\int_{(y-\mu)}^\infty (u-(y-\mu))e^{-\frac{u^2}{2\sigma^2}}du \end{aligned}$$

where $u = x-\mu$

$$\begin{aligned} &= \frac{\sigma}{\sqrt{2\pi}\sigma}\int_{\frac{(y-\mu)}{\sigma}}^\infty (v\sigma - y)e^{-\frac{v^2}{2}}dv \end{aligned}$$

where $v = u/\sigma$

$$= \sigma NL((y-\mu)/\sigma) \tag{3.20}$$

In this way, any normal loss function calculation can be reduced to a calculation in terms of the *standardized* normal loss function. Table 3.1 is a table of values of the standardized normal loss function which should prove adequate for most situations, and in particular will be adequate for all the exercises. More discussion of the normal loss function and its application in operations research is provided in [9].

Example 3.6. Loss Calculation

Suppose a communication link with capacity 10 Mbit/s is being offered 8 Mbit/s of traffic with standard deviation of the *bit rate* of 4 Mbit/s. Assume, furthermore, that the distribution of the *bit rate* is normal (Gaussian).

Calculate the amount of traffic which can actually be carried, assuming that the benefits of buffering are negligible, by using the Table 3.1 of the normal loss function. Note: $E\{Z-z; Z>z\}$ denotes the expected value of

$$\begin{cases} Z-z & \text{, if } Z>z, \text{ or} \\ 0 & \text{, if } Z \leq z \end{cases}$$

where Z is a standard normal random variable.

Even though the *average* bit-rate of the traffic lies below the capacity of the link on which it is being carried, since this bit-rate fluctuates, and for periods of time it is higher than 10 Mbit/s, there will be *loss* – packets will fail to pass successfully across this link. The amount of buffering which can be provided at the ends of the link will, moreover, probably be insufficient to significantly lower this loss.

This is not to ignore the fact that in some cases the packets which are lost will probably be transmitted later, due to high-level protocols which ensure that all offered traffic is eventually carried. Furthermore, there are also high level protocols (TCP for example), which will cause the rate at which traffic is offered to the link to adjust to a point where the loss rate is at an acceptable level. This is discussed in more detail in Subsection 3.5.3.

However, for the moment, let us ignore these high-level protocols and suppose that the traffic continues to be offered at the specified level and work out how much traffic will be lost, and how much will actually be carried.

If we apply (3.18), we find that the loss is

$$4NL(\frac{10-8}{4}) = 4NL(0.5) = 0.7912\text{Mbit/s}. \tag{3.21}$$

Since the traffic offererd to the network was 8 Mbit/s, the *loss rate* (i.e. the proportion of all packets which are lost) is is $0.7912/8 \approx 10\%$.

The calculations we have just described can be summarised by the formula:

$$\text{Loss Rate} = \frac{\sigma_0}{\mu} NL(\frac{S-\mu}{\sigma_0}), \tag{3.22}$$

in which S is the capacity of the link, μ is the mean offered traffic, and σ_0 is the standard deviation of the rate of the offered traffic.

□

Exercise 3.5. A Simple Network, and its Analysis

Suppose that the network in question is made up of a single link, with traffic flowing in one direction only, from left to right (as in Figure 3.1). (This may seem a bit unrealistic, but real networks can be broken down into components like this, so its a very important example).

Now to complete the definition of the problem, that you are to solve for this exercise, here is the question you are to answer:

> Assuming that whenever the number of bytes offered to the link, in a 10 millisecond interval, is greater than it can handle, the excess packets are lost, determine what proportion of packets will be lost. You will need to use one of the Tables 3.1 or 3.2. Assume that the link has capacity 10 Mbit/s, that the mean amount of traffic arriving for transmission on the link in a 10 millisecond interval is 3000 bytes and that the standard deviation of the number of bytes arriving in a 10 millisecond interval is 2000.

Next, solve the previous problems a second time, but this time using a 100 millisecond interval. You should assume that the statistics of packet length are the same, but the statistics of packets arriving in each interval are scaled up according to Hurst's law with $H = 0.8$, that is to say, the variance of the number of packets arriving in an interval of length t is given by the formula:

$$V(t) = Ct^{2H}$$

for some constant C. From this, and the known formulae for the variance in 10 millisecond intervals, work out a value for C. Then deduce what the variance of the number of packet arrivals must be in a 100 millisecond interval and repeat the loss estimate. □

3.4.3 The Central Limit Theorem

A strong argument for the use of Gaussian traffic models in the central core of a communication network is that the traffic in this part of a network will be formed as the aggregate, or sum, of many independent sources of traffic. The central limit theorem implies, as shown in [10], that the traffic in the core of our networks must be converging to a Gaussian process in distribution as networks get larger, and more traffic is aggregated together.

This means that for the large central links of a network it is not necessary to use the detailed Poisson-Pareto model, which has no simple queueing analysis. Instead, we can use a Gaussian model, which can be successfully analyzed [11].

The conclusion that real traffic is becoming more and more Gaussian is quite significant, so it is worth our while to check the assumptions on which this result is based. There is one critical assumption that deserves careful attention. This is the assumption that the upper limit to the rate at which sources of traffic can communicate will remain fixed or at least grow more slowly than the rate of the links which we use to carry this traffic. If this assumption fails, the central limit theorem will not be applicable, and we will need to use different models of traffic in networks. This issue is further addressed in Section 3.4.4.

In addition to convergence to a Gaussian traffic model, a second phenomenon occurs. If the individual traffic streams were Long Range Dependent, so will the aggregate traffic be. This feature is not mitigated by more and more aggregation of traffic. However, the impact of the *randomness* of the traffic will, nevertheless, gradually abate.

When n independent traffic streams with mean μ (of the *rate process*) and standard deviation σ_0 (of the rate process), the mean and standard deviation of the rate process of the aggregate traffic will be $n\mu$ and $\sqrt{n}\sigma_0$ respectively. As a consequence, it becomes gradually possible to carry an increasing aggregate of traffic with lower and lower levels of overhead for the purpose of reducing queueing delay.

3.4.4 Traffic with Infinite Variance*

The Poisson-Pareto burst process exhibits heavy tails in the lengths of the bursts. Moreover, a simple explanation for these long bursts is that the messages that network users want to deliver follow a Pareto distribution. So, if networks were faster than now, and growing faster all the time, users would want to send these Pareto distributed messages more and more quickly. Conceivably, this could lead to the situation where not only did the *bursts* have heavy tails, but even the distribution of the *traffic rates* might be heavy tailed.

The assumption that traffic in todays networks has a finite instantaneous variance is realistic but it is justified by the fact that traffic enters our networks via pipes with limited capacity. In effect, the infinite variance traffic which users would really like to submit is *shaped* so that it has finite variance.

In the discussion in Subsection 3.4.3, the Central Limit Theorem was used to show that if traffic has finite variance, traffic in the core will become more and more Gaussian. It was also noted that this depends critically on whether the variance to mean ratio of the traffic which makes up our networks remains much the same or, on the other hand, increases over time. If advances in networking technology allow the pipes by which users connect to our networks to expand sufficiently rapidly, the Central Limit theorem would not apply, and traffic would potentially tend to a different limiting form altogether. A good candidate for a model of traffic to use in place of Gaussian processes, under these circumstances, would be the stable stochastic processes [12].

However, even if networking *technology* allowed growth in access speed to our networks to be sufficient that the assumptions of the Central Limit Theorem did not apply, it is doubtful whether sensible *management* of our networks would allow this to happen. In this scenario, as networks grow, no matter how many users are connected, there would still be a small number with the potential to dominate the entire network by offering a particularly heavy burst of traffic. It seems unlikely that we would want to allow this scenario to come to pass.

3.5 Analysis of Loss and Delay

Design, and in particular dimensioning, can be viewed as the process of making decisions which minimize cost subject to constraints on performance, and in the present case, the performance we are concerned with can be quantified in terms of availability, loss, and delay.

This is the justification for expending considerable effort on the task of estimating the performance which a network might deliver on the basis of a description of the network and the traffic it attempts to carry.

3.5.1 Queueing Delay

A convenient and accurate for queueing delay is given in [11]:

$$\mathrm{P}(D > x) \approx \exp\left(-\frac{2(c-m)^2 v(t_x^*)}{v'(t_x^*)^2}\right) = \exp\left(-\frac{(x+(c-m)t_x^*)^2}{2v(t_x^*)}\right). \tag{3.23}$$

in which $X(t)$ is the net rate of arriving "bits" (or whatever unit seems appropriate) at time t, m is the mean of $X(t)$, c is the rate of the server,

$$v(t) = \mathrm{Var}(\int_0^t X(t)dt),$$

and, finally, $t_x^* \geq 0$ is the unique real number greater than zero which minimizes $(x+(c-m)|t|)^2/v(t)$. If v is differentiable outside the origin, t_x^* is the solution of the equation

$$\frac{2v(t)}{v'(t)} - t = x/(c-m). \tag{3.24}$$

In some cases (see [11], and (3.25) below), this formula can be solved explicitly. There are some important special cases which were solved much earlier, which should be mentioned. The Gaussian discrete time queue with short-range dependent input was solved in [13]; the queue with Fractional Gaussian Noise input (in continuous time) was solved in [14]; and the discrete-time Gaussian queue with Fractional Gaussian noise input was solved in [15].

Even in the case where (3.24) cannot be solved, so long as a method of computing $v(t)$ is available, the formula (3.23) is readily computable.

A closely related formula for queueing delay, also from [11], in the case where the work process has a well-defined rate, X_t, and the server has the rate S is that

$$P(D > 0) \approx 2P(X_t > S).$$

Example 3.7. Fractional Gaussian Noise

A special case of a Gaussian model which is particularly important, because it seems to fit real traffic quite well, is Fractional Gaussian Noise (FGN). This model was discussed previously, in Subsection 3.3.6. Now we want to apply (3.23) to find an explicit formula for the delay in a system where the input process is FGN.

In this case, $v(t) = \sigma_1 t^{2H}$, so (3.24) becomes

$$\frac{2t^{2H}}{2Ht^{2H-1}} - t = x/(c-m) \Longrightarrow t_x^* = \frac{Hx}{(1-H)(c-m)}.$$

It follows that [14]

$$\mathrm{P}(D > x) \approx \exp\left(-\frac{(x/(1-H))^2}{2\sigma_1\left(\frac{Hx}{(1-H)(c-m)}\right)^{2H}}\right) = \exp\left(-\frac{(c-m)^{2H}x^{2-2H}}{((1-H))^{2-2H}2\sigma_1 H^{2H}}\right) \tag{3.25}$$

□

Exercise 3.6. Gaussian Noise

Derive the delay formula for the special case of Gaussian Noise, i.e. the special case of Fractional Gaussian Noise where $H = 0.5$. □

Exercise 3.7. Delay Plots

Plot the complementary delay distributions for the model considered in Example 3.7, for the following parameter choices: $c-m = 1$, $H = 0.5$, $H = 0.6$, $H = 0.7$, $H = 0.8$, $H = 0.9$. □

Example 3.8. The Poisson-Pareto Burst Process

This example was discussed earlier in Subsection 3.3.7. This traffic model is fairly realistic in structure and it seems to fit real traffic quite well. Unfortunately it does not have a simple formula for delay in a stationary queueing system.

Some studies of the queueing behaviour of systems carrying this traffic have concluded that the behaviour of such queues must be quite different from queues carrying Gaussian traffic. On the other hand other studies, which have been confirmed by means of simulations, support the contention that for sufficiently large aggregation levels, which really means sufficiently high arrival rates of bursts, the queueing behaviour of this process will be quite similar to that of a system carrying FGN traffic. The simple explanation of this inconsistency is that the former results are making use of a method (large deviations) for deriving a stationary queuing formula which produces results which are accurate only in a region (of buffer levels) which is usually of no interest (because the buffer levels are too large).

The simple expedient of using an FGN model with appropriate parameters to replace a PPBP model has some appeal, but may be dangerous. The Poisson Pareto Burst Process queueing delay appears to be consistently *worse* than the delay in a corresponding Gaussian system. □

3.5.2 Loss Estimation

We have already estimated the total loss, in the situation where buffering is negligible, or correlation so high that buffering is ineffective. This is achieved by means of the normal loss function, and if the rate process has mean μ and standard deviation σ_0, and the server has rate S, the loss rate (in the same units as the offered traffic) is $\sigma_0 NL(\frac{S-\mu}{\sigma_0})$, so the *loss rate* measured as a percentage of the offered traffic will be

$$\frac{\sigma_0}{\mu} NL(\frac{S-\mu}{\sigma_0}). \tag{3.26}$$

3.5.3 End-to-end Control of Traffic in TCP/IP networks

The TCP end-to-end protocols have quite a significant effect on loss and delay in TCP/IP networks [16]. The diagnostic effect which the TCP protocol responds to is loss occurring somewhere along the path of the connection. When a significant amount of loss occurs, the protocol throttles back the transmission rate of this connection.

The TCP protocol is designed to ensure that the source and the destination find a satisfactory rate for transmission in either direction. In some cases it might be the destination that imposes the strictest limit on the possible speed of transmission, in another case it might be the source, and in yet another case it might be that one link in the path between the source and the destination imposes the tightest constraint on transmission because of the capacity of this link or the quantity of other traffic carried on it.

The interesting case, and not an uncommon one in today's Internet, is where it is a link in the middle which sets the maximum feasible rate for transmission. The TCP protocol is not able to completely eliminate loss in this situation because losses of packets is the mechanism used to signify to the protocol that there is a problem. However, many services are able to tolerate the degree of loss incurred in this way gracefully.

So what happens to the rest of the traffic demand? Suppose the source wants to transmit at 1 Mbit/s and the destination is capable of receiving packets at an even higher rate, but the path in-between can only sustain 200 kbit/s. In this situation, the source must be capable of queueing the unsent packets until the available transmission capacity is able to cater for it. In some cases this will provide a satisfactory outcome.

For example, if the task is to send a file, neither the source nor the destination will be too concerned if the file takes 2 minutes to receive rather than 1 minute. In effect, the packets to be sent can be queued, in this case, right back at their real source – the file. This consumes no additional resources and the actual service is still carried out in a time which is satisfactory for the end-user.

However, there are other services which could find this sort of behaviour of the end-to-end path quite unsatisfactory. An example of such a service is real-time audio, e.g. a concert broadcast. By means of compression, such a broadcast might be reduced to a rate close to 64 kbit/s without losing a great deal of quality. However, if the transmission path can only sustain 32 kbit/s, without some special consideration for the packets of the broadcast, a TCP connection will produce, at the far end, a quite unsatisfactory result. The packets which cannot fit within the

32 kbit/s rate cannot be queued because if they are delayed beyond a certain time they will not arrive soon enough to be played back at the appropriate time. Hence, the signal, when it is played back, will be missing half of the signal.

This leads us to the important topic of techniques for supporting special levels of performance on the Internet. This will be addressed in Subsection 3.5.5. Before we come to that we need to consider how we should decide on link capacities.

3.5.4 Dimensioning

Dimensioning is the process of deciding how many and what speed (or size) of equipment to install for a given traffic load. The assumption we will make is that our objective is to meet a specific performance standard with the least costly (and hence slowest, smallest, or least number of) available equipment.

The equipment under consideration could be a transmission system, a router or a switch, or any piece of equipment used to transport or process data on its way across a network.

Some time ago, transmission systems changed from an economic model in which n circuits cost n times as much as one circuit to a model in which 30 channels cost only 5 times as much as one circuit. When this happened, the rationale for dimensioning changed significantly. This process of increasing *modularization* has continued unabated till now the range of transmission systems to choose from might be one of the OC-1 (50 Mbit/s), OC-3 (150 Mbit/s), OC-12 (600 Mbit/s), OC-48 (2.4 Gbit/s), OC-192 (9.6 Gbit/s), OC-768 (40 Gbit/s) or perhaps a bundle of 160$\times$OC-192 (1.6 Tbit/s).

However, even if the dimensioning task has withdrawn from the central role in the management of telecommunication networks that it once held, it is too early to judge that we don't need to make this sort of judgement at all.

The performance standards we need to keep in mind when dimensioning networks are those which refer to *loss*, and *delay*. The approach we intend to take here will effectively merge these two performance issues together. This is not so unreasonable, because the delay and loss performance of a transmission system can be inter-traded readily by adjusting the size of the buffers. With large enough buffers, there will be no loss, but delay will be at a maximum. With no buffers at all, there will be no delay, but loss will be at a maximum. On the other hand, by increasing the capacity of the transmission system (or the speed of the processor, in the case of a router for example), both loss and delay performance of the system can be improved. The capacity of the system can be increased sufficiently that the system under consideration meets delay and loss standards for any choice of buffer size.

As discussed above, it is not unreasonable to suppose that the traffic process can be modeled as a fluid, with a *rate* process, as discussed above in Subsection 3.4.

With this view in mind, it is easy to see *when* traffic will be lost, or buffers fill: it will happen when the traffic rate process exceeds the rate at which the transmission system (or router, or switch, etc) can deliver traffic.

Since the traffic rate process is Gaussian, we can see quite readily how to keep the probability of this event occurring down to a very low level: we need to choose the capacity of the transmission system so that the probability of the rate exceeding this level is very low. For example, from Table 3.2, the probability of a Gaussian random variable exceeding the mean plus two standard deviations is 0.0228 and the probability of a Gaussian random variable exceeding the mean plus *three* standard deviations is 0.00135.

The probability of loss is best thought of as the proportion of bytes, or packets, which are lost. Keeping this in mind, we should really look up Table 3.1, not Table 3.2, because not all the bytes which arrive will be lost. The proportion of bytes exceeding 2 standard deviations according to Table 3.2 is 0.008491 and the proportion of bytes exceeding the mean plus *three* standard deviations is 0.0003822.

The dimensioning problem puts this around another way. The question is: what capacity do we require in a transmission system in order that the probability of loss should not exceed a given standard.

Exercise 3.8. Dimensioning a Simple Network

Suppose that the traffic to be carried on a link has mean rate 1 Mbit/s and standard deviation (of the rate, measured over an appropriate time interval) 1 Mbit/s. The available equipment for carrying this traffic has capacity one of: 1 Mbit/s, 2 Mbit/s, 10 Mbit/s, 20 Mbit/s, 50 Mbit/s, 150 Mbit/s, 600 Mbit/s, 2.4 Gbit/s, 10 Gbit/s, or 40 Gbit/s.

Select the appropriate transmission system to carry the traffic specified, an aggregate of 10 such traffic streams, 20 streams, 50 streams, and 100 streams. Justify your answer.

The loss/delay standard on the link should be met by ensuring that the probability of the traffic rate process exceeding the capacity of the link should not exceed 0.001 (0.1 %).

A handy table of probabilities of the Gaussian (normal) distribution which you can use for this exercise is given in Table 3.2. If you feel that the *Normal Loss Function* is the one which you need to use, you can gain access to it in Table 3.1. □

Exercise 3.9. Dimensioning a Simple Network

In the same context as the previous question, suppose, as well as the original traffic, traffic *A* way, with mean rate 1 Mbit/s, and standard deviation (of the rate) 1 Mbit/s, another independent type of traffic, traffic *B*, of mean 2 Mbit/s and standard deviation 4 Mbit/s must be carried by the link. What capacity, from those listed in Exercise 3.8 would you choose to carry traffic *A* together with traffic *B*? What about 5 independent streams of traffic *A* and 10 of traffic *B*? □

3.5.5 Differentiated Service

Almost for the entire history of communications and networking there have been calls for networks to provide different standards of service for different classes of customer. On the other hand, the technical difficulties facing network providers in differentiating between classes of traffic have prevented, up to now, any widespread provision of differentiated service. Railway networks have overcome these problems and in many places around the world it is possible to pay more for 1st class and less for 3rd class travel on trains and aeroplanes. On the other hand, the very high capacity rail networks which lie under the larger cities of the world usually target just one class of customer, presumably because even with the best effort to make a difference, all customers would really have much the same experience.

It is still unclear whether networks will successfully aspire to provide first class passenger service. However, quite a lot of work has gone into the techniques and technology of providing differentiated service [17, 18]. Some of the networking ideas which are linked to differentiated service, such as Asynchronous Transfer Mode (ATM) (see Section 7.3.2 and Chapter 8) and Multi-Protocol Label Switching (see Section 5.4.1), cite the provision of differentiated service standards as *one of the many* benefits of the proposed network architecture, without suggesting that it is *main* reason for its adoption.

On the other hand, the Internet service framework known as DiffServ [19] does firmly focus on differential service as the goal.

Nevertheless, the benefits and feasibility of differentiated service remain unproved. During a large part of the 20th century, networks capable of providing differentiated services to different classes of telephone customer have not,to this point, developed to a significant degree. Networks with special robustness, performance, and reliability constraints have occasionally been proposed or developed as a special separate entity, only to find that sharing resources with a large public network, such as the public telephone network or the public Internet is the best way to achieve high robustness and reliability.

One way to express the technical issue is this: is it possible that lowering the quality of service to a class, *B*, of users could be a *cheaper* way to improve the quality of service to a class, *A*, of users than simply providing more resources?

In the case of long distance flight services, because of the limited space in aeroplanes, and the willingness of a certain proportion of customers to pay high prices for their seats, the advantages of differentiated service are clear. In the case of the Paris Metro or the London Underground, on the other hand, since it is infeasible to provide any differentiation in the access to and from the stations, it is obvious that provision of different class carriages will not serve any useful purpose.

In the case of a modern integrated communication network, we can do some simple performance modelling to determine the potential gains of differentiated service. In order to do so, we need to define the problem more precisely, which we now do.

3.5.6 Estimation of Performance of Separate Streams

To keep things simple, let us concentrate on just two classes of traffic, a priority class and an ordinary class. To estimate the performance which will be experienced by these two classes in a *typical case* should not be unduly difficult. By a *typical case*, is meant one where the priority traffic is not an excessive proportion of the whole, e.g. no more than 10%.

Note: it is important that the priority class of traffic is limited to a certain proportion of the whole. If this was not the case, the low priority class of traffic could be completely frozen out for a period of time. To avoid this happening, it is necessary to treat the priority traffic a little differently than conventional, best-effort traffic at the point where it enters the network. Instead of allowing network users to submit as much traffic as they wish, it is expected that users will enter into a sort of *contract* with the network provider known as a *service level agreement* (SLA). The settings in this SLA will affect the charge this user incurs for use of the network.

In addition, at the point where a user of priority traffic transfers their load to the Internet at large, it will be necessary to *police* the traffic to ensure that the SLA has been honoured. If users were allowed to submit traffic without any form of checking, there would be no reason not to mark all submitted traffic as belonging to the priority class.

Under these circumstances, where the priority class of traffic cannot rise above a certain limited proportion of all traffic on any individual link, the performance experienced by the low priority traffic class will not be significantly different than if all traffic was treated as in one class. At the same time, the high priority traffic will receive very close to the performance which would be delivered if the network was carrying *only* high priority traffic.

This analysis may seem to suggest that we can get something for nothing. In fact, the low priority traffic will be affected to a degree and if the quantity of high priority traffic was excessive the low priority traffic would be severely affected.

In the present case, where the quantity of high priority traffic is kept to a minimum, to estimate the performance of the two classes of traffic we only need to carry out two separate calculations: one for the network carrying only high priority traffic, and a second calculation for the network carrying all the traffic. In the high priority calculation, we should use the parameters appropriate to the high priority traffic, and in the other calculation we should use the parameters appropriate for the low priority traffic.

It is not unreasonable to assume that the high priority traffic is mainly voice, or at least *like* voice. For example, this traffic might be video traffic which includes a voice component. Therefore, we should assume that it is aiming for loss levels less than 1% and that the appropriate time-scale is in the range 50-500 milliseconds.

Example 3.9. Estimation of Different Performance for Different Classes of Traffic

Let us now consider an example in which a link is required to carry 9 Mbit/s of data traffic and 1 Mbit/s of voice traffic on a 20 Mbit/s link.

Assuming that the 1 Mbit/s of voice traffic is encoded at 16 kbit/s, it must represent 60 Erlangs of telephone traffic, i.e. on average there are 60 active calls at once on this link. We know that the probability distribution of the number of calls will be Poisson and so the variance of the number of calls at one time must be 60, which implies that the standard deviation of the number of active calls is $\sqrt{60} \approx 8$. It follows that the standard deviation of the *rate* of the voice traffic is $\approx 16 \times 8 = 128$ kbit/s.

As for the data traffic, let us suppose that this has a Hurst parameter of 0.8, and a standard deviation equal to 2500 Mbit when measured over a 10 minute interval. The mean bytes arriving in a 10 minute interval would then be $10 \times 60 \times 9 = 5400$ Mbits.

So, if we now check the priority traffic, assuming that it is carried all by itself, we see that the capacity of 20 Mbit/s is $(20-1) \times 1000/128 \approx 100$ standard deviations of the voice traffic more than the mean value of the voice traffic. It follows that the voice traffic will experience no loss or delay, so long as it is given absolute priority over other traffic.

Now consider the low and high priority traffic together, over a ten minute interval. The mean total amount of traffic is 10 Mbit/s, which translates to 6 Gbits over a 10 minute interval. The variance of this traffic is dominated by the variance of the data traffic. The easiest way to estimate the variance of the total traffic is to simply ignore the variance of the voice traffic altogether. When two random variables are added together, the variance of the result is very similar to the larger of the two variances.

The correct formula for combining standard deviations of independent random variables is

$$\sigma_{\text{comb}} = \sqrt{\sigma_a^2 + \sigma_b^2},$$

however when $\sigma_a \gg \sigma_b$, $\sigma_{\text{comb}} \approx \sigma_a + \frac{\sigma_b^2}{2\sigma_a} \approx \sigma_a$.

It is more natural to quantify the random variation of the traffic by means of its standard deviation rather than by variances. The standard deviation of the data traffic measured over a 10 minute interval has been given to us as 2.5 Gbits.

The capacity of the link over a 10 minute interval is $10 \times 60 \times 20$ Mbits$= 12$ Gbits, so the excess capacity is 6 Gbits, which is a little over 2 standard deviations. If we consult Table 3.1, we find that the loss rate in this system will be

$$\approx \frac{2.5}{6}\text{NL}(2.25) \approx 0.025 = 2.5\%. \tag{3.27}$$

The conclusion is that this link might be a little underdimensioned to carry this traffic. If nominal losses of 2.5% were really implied by the traffic load supplied to this link on a regular basis, it would mean that users would really be deferring or forgoing their use of the data link to a significant degree. So, we conclude that 20 Mbit/s is not quite sufficient.

How much capacity would be sufficient? In equation (3.27) we would need to see the resulting calculations produce 1% or lower loss, so we should seek a standard score, or Z value, at least as big as 3. That is to say, the spare capacity on the link should be at least 3 standard deviations. One standard deviation of this traffic is 2.5 Gbits, so three is 7.5 Gbits, implying that the required capacity is 13.5 Gbits, in a ten minute interval. This is equivalent to a transmission rate of 22.5 Mbit/s.

Note that if the priority traffic really is kept to below 10% of link capacity, this example is likely to be typical in the respect that the priority traffic can be expected to experience virtually perfect service while not disrupting the remaining traffic to any significant degree. So long as the SLA's and the policing is working well, it appears that this approach to providing differentiated service is working well.

For a more sophisticated approach to estimating the performance of the lower priority traffic in a system such as the one discussed here, which should be reasonably accurate even if there is a lot more high priority traffic, see [20]. □

3.5.7 Benefits of Differentiation of Service

In this subsection we shall consider the benefits of differentiated service. We have in mind, primarily, the DiffServ architecture proposed for the Internet [19]. However, we will not go into more details of this architecture than are necessary for us to estimate performance of the individual traffic streams and thereby estimate efficiency. A more detailed consideration of the DiffServ architecture is undertaken in Subsection 5.3.3.

Example 3.10. Benefits of Differentiated Service

Now let us consider how we could potentially go about providing the two services described in the previous example *without* the differentiated service mechanism of priority bits and two separate queues for each traffic type in every router. In this non-differentiated situation, we will have to provide the top quality to all the traffic. Can we afford to do this?

The same principle that we have used in all other cases can be applied again. The required capacity is just the mean traffic plus a certain number of standard deviations. But now, we must treat all the traffic as voice. This means that we use a time-interval of 100 milliseconds, the variance of the data traffic will have to be translated to this new time scale, and then we will need to ensure that there is sufficient capacity to provide headroom of at least 3 standard deviations. This sounds easy enough, but keep in mind that we need to use the mean traffic arriving in a 100 millisecond interval, and the standard deviation of such traffic.

The standard deviation of the voice traffic has already been calculated, and is $8 \times 16 = 128$ kbit/s.

To calculate the variance of the data traffic at this time scale of 100 milliseconds, we can use the Hurst law, for standard deviations, i.e.

$$\sigma(100\text{ms}) = \left(\frac{100}{600000}\right)^H \sigma(10\text{min}), \tag{3.28}$$

We know the value of $\sigma(10\text{min})$ for the data traffic – it is 2.5 Gbits. This, with (3.28), produces the estimate

$$\sigma(100\text{ms}) = 2.37\text{Mbits}.$$

So the standard deviation of the two traffics together is $\sqrt{2370^2 + 128^2} = 2373$ kbit/s. Notice that the addition of the variance of the voice traffic has an insignificant effect. To provide 1% loss for this combined traffic stream at this time scale will therefore require additional capacity, above the mean, $3 \times 2.37 = 7.1$ Mbits every 100 millisecond. Since the base load is 1 Mbits every 100 millisecond, the total capacity required will be 80 Mbits/s.

This shows, in effect, that attempting to carry voice traffic as well as data traffic in this situation, without the DiffServ architecture, is very inefficient. It is unlikely that network providers could be persuaded to provide sufficient capacity to provide voice quality service end-to-end for both data traffic and voice traffic.

The accuracy of this estimate will be affected by a number of factors which are difficult to quantify. In particular, the assumption that the Hurst law for the variance of the data traffic arriving in an interval of time holds across the full range of time scales in this example must hold at least approximately in order for the conclusions of this example to hold good.

However, the broad conclusion that prioritising voice traffic is a very effective way to provide the required service for voice traffic without impinging significantly on data traffic can be expected to hold good under broad conditions. □

In order to be able to easily explore a range of situations, let us develop a formula for the amount of additional capacity required on a link when the level of service required by the priority traffic is given to all the traffic.

Assumptions:

- time scale for best-effort data traffic: Δ_d;
- time scale for priority traffic: Δ_p;
- mean of priority traffic: μ_p, in bits/s (or, as indicated);
- mean of data traffic: μ_d, in bits/s (or, as indicated);
- standard deviation of data traffic: $\sigma_d(\Delta_d)$;
- mean of priority traffic: μ_p;

The mean combined traffic is $\mu_d + \mu_p$.

The headroom required to ensure that performance standards are met is different in the two cases. In the priority case, the extra capacity required will be determined, as we saw in the preceding example, under the conditions imposed by the performance requirements of the data traffic. So the headroom required will be $3 \times \sigma_d(\Delta_d)$ expressed as bits arriving over the interval Δ_d.

In the non-differentiated case, the headroom required will be determined by the performance requirements of the voice traffic applied to the entire aggregate of traffic. Hence, the required headroom will be $3 \times \sigma_d(\Delta_p)$, expressed as bits to be carried over the interval Δ_p.

We need to translate these two headroom figures to a common time interval in order to be able to compare them. Let us use the time interval Δ_p as the common standard for purposes of comparison. The headroom for the non-differentiated case is already expressed in terms of quantities over this time interval. As for the other case, we need to multiply the headroom figure by $\frac{\Delta_p}{\Delta_d}$.

Thus, the saving derived from using priorities will be

$$3 \times \left(\sigma_d(\Delta_p) - \frac{\Delta_p \sigma_d(\Delta_d)}{\Delta_d} \right).$$

Using the Hurst law for the data traffic, we find that $\sigma_d(\Delta_p) = \left(\frac{\Delta_p}{\Delta_d}\right)^H \sigma_d(\Delta_d)$, hence the capacity saved by using priorities is

$$3 \times \sigma(\Delta_d) \left(\left(\frac{\Delta_p}{\Delta_d}\right)^H - \frac{\Delta_p}{\Delta_d} \right)$$

in each interval of length Δ_p. In order to express this gain as a transmission rate, we must conclude by dividing by Δ_p, giving a gain of

$$3 \times \sigma(\Delta_d) \left(\frac{\Delta_p^{H-1}}{\Delta_d^H} - \frac{1}{\Delta_d} \right) \tag{3.29}$$

Example 3.11. A More Secific Case

Now suppose that $\Delta_d = 5$ mins, $\Delta_p = 50$ milliseconds, $\mu_d = 4$ Mbit/s, $\mu_p = 1$ Mbit/s, $\sigma_d(\Delta_d) = 300$ Mbits.

Under these circumstances, the capacity required to carry the data and the priority traffic, using just the standard of the data traffic, will be, in Mbits in a 5 minute interval:

$$300 \times (\mu_d + \mu_p) + 3 \times 300 = 2.1\text{Gbits},$$

or, as a transmission rate, 7 Mbit/s.

If we attempt to carry all the traffic at the quality required for voice, the capacity will have to be, in Mbits in a 50 millisecond interval:

$$0.05 \times (\mu_d + \mu_p) + 3 \times \frac{300}{6000^{0.8}} = 1.1 \text{ Mbits}$$

which translates to a transmission rate of 22 Mbit/s. The saving from using priorities appears to be 15 Mbit/s.

The potential savings from the introduction of priorities can also be computed by using (3.29), which produces the estimate

$$3 \times \sigma_d(\Delta_d) \left(\frac{\Delta_p^{(H-1)}}{\Delta_d^H} - \frac{1}{\Delta_d} \right) \approx 3 \times 300 \times \left(\frac{0.05^{-0.2}}{300^{0.8}} - \frac{1}{300} \right) = 14.09 \text{ Mbit/s}.$$

□

Exercise 3.10. The Benefits of Differential Service

Suppose the priority traffic in a system carrying two classes of traffic has a mean of $\mu_p = 0.5$ Mbit/s, the data traffic has a mean of $\mu_d = 8$ Mbit/s, the time interval over which data traffic is buffered is $\Delta_d = 10$ minutes, the time interval over which the priority traffic is buffered is $\Delta_p = 200$ milliseconds, and the standard deviation of the data arriving in an interval of time of length Δ_d is $\sigma_d(\Delta_d) = 4$ Gbits.

Assume that the priority traffic is made up of voice calls encoded at 16 kbit/s. Assume that the performance required for this voice traffic is a loss rate of 5% when the traffic is buffered over a time interval of 200 milliseconds. The data traffic is expected to be carried in such a way that it experiences a nominal 1% loss when buffered over a time interval of 10 minutes.

How much capacity is saved by using a differential service architecture to carry these two classes of traffic rather than just providing premium service to both classes?

You should work out this saving in two different ways. First, work out the required capacity to carry all the traffic at the performance level expected for the voice. Then work out the capacity required when all the traffic is carried at the performance level expected for data. Subtracting the second figure from the former gives the saving gained by using a priority scheme for the priority traffic.

Then recompute the saving using the formula (3.29). The answers should be the similar, but not necessarily exactly the same. Why shouldn't the answers be exactly the same. □

For further discussion of differential service from the point of view of routing, see Subsection 5.3.3.

3.6 Security

The traditional performance issues which concern designers and managers of networks are loss and throughput (these two go together), delay, and reliability. However, in practice, there is another important issue: *security*.

Analysis of security does not readily present itself as a scientific subject, and design of secure networks at present appears to be achieved by a disparate collection of ad hoc techniques: firewalls, filtering, authentication, authentication servers and services, encryption, selection of IP address space allocation and routing plans which support various rules for traffic segregation, use of VLANs for the same purpose, and careful maintenance of

individual hosts (applying security patches as soon as they become available) for the purpose of avoiding security weaknesses.

If this grab-bag of techniques can be organised a little more scientifically, and broader classes of methods for achieving security identified, we will have achieved something.

Security will be *analysed* in this chapter and we will then return to the topic again in Section 4.3.2, where we discuss *measurements*, in Chapter 5.1, where we discuss routing, including security aspects, and then again in Chapter 9 where the subject of *design* for security will be addressed.

First of all, we need to *define* security.

3.6.1 Definition of Security

As with reliability, security is best defined in terms of its complement: lack of security. (Unfortunately, insecurity is not the opposite of security in the present context!)

Typically our networks are designed to provide certain services, S_1, S_2, etc, to certain users, U_1, U_2, ... Access to certain services may be restricted, e.g. S_1 is only to be available to users U_1 and U_2. A security *breach* has occurred whenever a user accesses a service to which they were not intended to have access. For example, we might declare that the legitimate users of a certain computer are just those in the group *A*. A security breach has occurred whenever a person outside this group gains access to the computer.

Security breaches can be broken down into certain types:

(i) access to information by an unauthorized party;

(ii) impersonation of an authorized party by an unauthorized party;

(iii) prevention of *authorized* activities by actions which do not in themselves represent a security breach (denial of service); and

(iv) misuse of services by legitimate users.

In broader terms, we can also talk of *security of action*. Each node in a communication network offers certain *services*. If a node offered no services at all there would be no reason for the node to be connected to the network. These services are offered to a limited range of *clients* and enforcement of the limitations on who can access a given service is achieved by *authentication*. A security breach has occurred whenever a service has been accessed by a party other than one of the authorized clients, or *whenever a service has been used in a manner which was specifically disallowed*. This is the last category of security breach in the above list, and it is this particular problem which, arguably, is the most difficult to address.

For example, a host might offer the telnet service to a group of clients (users) with the proviso that these users should only use the telnet service to access their own accounts. If a client uses the telnet service to gain access to someone else's account, e.g. the administration user of the host, a security breach has occurred.

Another example is the sending of SPAM, or sending an email message with an attached virus. The outward form of these actions is identical to a legitimate use of an offered service, but the details of the use which is being made of the service show that it is inappropriate – a breach of the usage rules.

3.6.2 Analysis of Security Issues

It is tempting to follow the model of loss or delay performance analysis when analysing security, however lack of security is quite different from loss and delay.

In particular, there are no well established quantitative measures for security failures – no way to measure the degree of severity, duration, or impact.

The critical events in the life of a network from a security point of view are:

(i) inappropriate access to data (data is observed by parties not intended to have access to it);

(ii) alteration of data by inappropriate parties;

(iii) denial of service; and

(iv) inappropriate use of a service.

The severity of any of these events may differ depending on how critical the data is, how long the event has gone undetected, and so on. Often security breaches cause additional costs to be incurred: systems have to be shut down, services disabled, and the daily routines of system administrator's may be severely disrupted. These additional flow-on costs can be as severe as the cost of the original event. Furthermore the impact of these flow-on events is easier to measure. The original incident in some cases leaves very few direct and measurable impacts upon the affected systems.

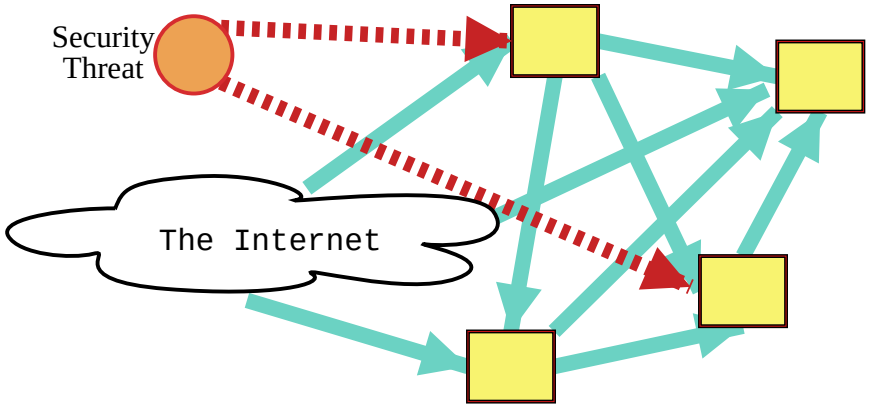
In order to establish some sort of a quantitative model for security, and its lack, we now seek to put in place a model. This model is not necessarily comprehensive, but it *is* quantitative.

3.6.3 A Simple Model of Security and Its Analysis

Suppose the network is as in Figure 3.2. There are a number of services to be offered, maintained, and protected against misuse. Each service attracts a number of traffic streams. In addition, there is a source generating security threats. The security threats are depicted in a form similar to traffic streams. They are depicted with striped arrows to make the distinction between the traffic and the security threats clear.

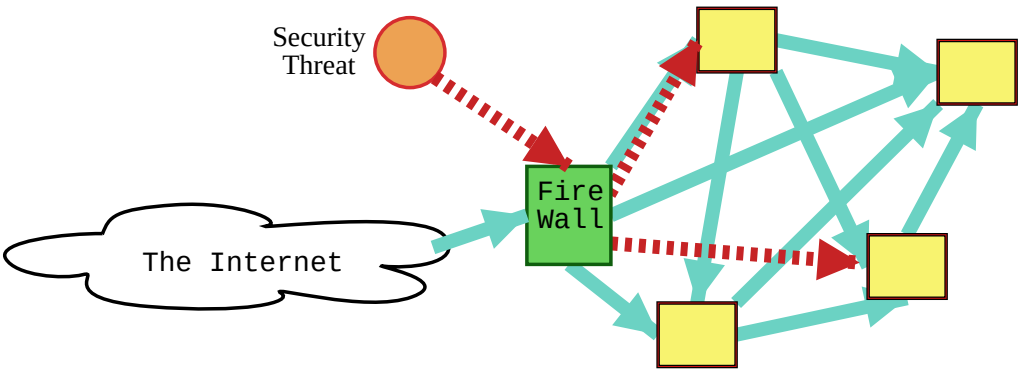
Whereas with traffic streams, our objective is to enable clients to gain maximum access to services, and to transport traffic with the best possible performance, in the case of security threats, our objective is to *inhibit* access as comprehensively and as early as possible along the network path.

Figure 3.2: A network with services and with security issues



One of the strategies for inhibiting security threats is to channel *all traffic* via a *firewall*, as in Figure 3.3.

Figure 3.3: A network with services and with security issues



A firewall can take various forms. In the simplest form, a firewall is a router with an appropriate collection of routing rules. Unfortunately, threats masquerade as genuine traffic, so it is not sufficient to use one strategy to protect a host or a subnetwork. Other strategies must also be used.

Security threats vary in *severity*, i.e. the amount of damage caused by any traffic which manages to get past the defences of the network or host, and *intensity*, i.e. the number of attempts which are made in a given period of time, and *location*.

The intensity of a security threat can be measured in the same way that we measure call or packet arrivals, i.e. events per second. The *severity* of a traffic threat is more difficult to measure, but an appropriate *quantification* of severity is by means of the quantity of "good" traffic which is affected. For example, if a server which normally delivers 3 Gigabytes to customers in a single day has to be taken down from service for a whole day, the severity of this incident is 2 Gigabytes. *Locality* of security threat is not a quantitative property. We can only speak qualitatively of whether a security threat is "fairly local" in its effects, or "widespread" in its effects.

Viruses and worms have an unusual mode of operation in that they are able to form a residual presence in a remote location from which they are then able to further migrate and interfere with normal operations. Such threats are *widespread*, in fact, one could say *global* in their impact.

At the other end of the scale, as far as locality is concerned, is the threat which is located at a specific site in the Internet: nefariously sampling traffic which passes by. This particular threat exists in other networks, for example telephone tapping is standard operational practice for certain categories of police all over the world. There is a strong perception that interception of communication is a real and significant risk when using the Internet.

The defence against this risk is somewhat different. Ecryption and authentication are now used in a widespread manner all over the Internet specifically to avoid this sort of problem, particularly when data such as passwords for access to financial records, and bank accounts, is being transmitted.

Exercise 3.11. Security Analysis

Consider the example described in §1.3.5, from the point of view of security. For the purpose of this example, let us assume (perhaps to make this exercise more interesting, although there are other reasons for making this choice), that the organisation makes maximum use of Internet communication facilities. Do not attempt to solve all the problems that you detect, just identify as many problems as you can and describe them in terms of intensity, severity, and locality. □

3.7 Examples

Let us now revisit the seven examples which were introduced in Chapter 1. In this chapter we shall discuss each of these examples and consider the performance issues which are particularly relevant, and discuss how these performance issues might affect our decisions concerning choice of equipment and the design of the networks. These examples will be considered again in subsequent chapters.

Example 3.12. A Home

This is the most cost-sensitive context we might want to consider. Nevertheless, performance issues are relevant. The level of aggregation of traffic in a home is as low as we are likely to find anywhere. It is likely that at most times there is precisely one dominant activity on the network, and so the network could, perhaps be designed around the objective of carrying one service at a time satisfactorily.

Delay within a home network is unlikely to be a problem. Packet loss is also unlikey to present a problem except in extreme situations. It follows that the selection of the speed of a LAN for a home is not a critical decision. If multiplayer gaming and file-sharing parties are a regular occurrence, it might be sensible to go to some extra trouble to ensure that the network operates at high speed, i.e 100 Mbit/sec or better.

Security, however, is a more important consideration. The key security issues relate to the interface between the home network and the Internet at large. An appropriate strategy to handle these security issues is to establish a gateway which prevents communication initiated outside the home from interracting with hosts inside the home except in quite specific ways which are chosen to be safe.

Email represents a particular problem because the security threat can only be identified, and dealt with, by investigating the *contents* of incoming messages. A service to protect home users from the dangerous aspects of

Internet email which lies outside the home could be cost effective and easier to establish than an email filter at a gateway. □

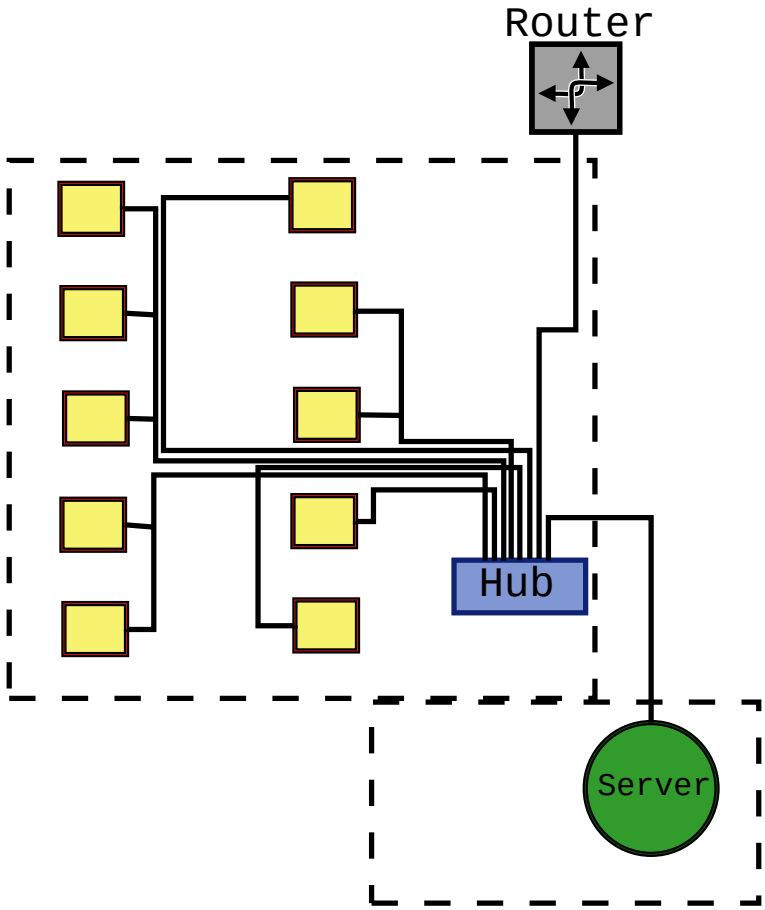
Example 3.13. A Laboratory

All four security issues – reliability, loss, delay, and security are relevant to the design of a laboratory network. Reliability was a topic of an earlier chapter.

Throughput (and hence loss and delay) is a critical issue in a computer laboratory because there are likely to be regular occurrences of roughly simultaneous access to certain high bandwidth services. For this reason, it makes sense to install a high speed network, 100 Mbit/sec or higher, in a laboratory LAN. The tighter distance restrictions associated with the higher speed LAN will not be a problem in a LAN.

A simple method for reducing LAN traffic and thereby improving LAN performance is to segregate the traffic in a LAN from the rest of the campus or school network. This suggests the use of a separate subnet for each Laboratory. This does not necessitate the installation of a separate router for each laboratory, just the careful configuration of whatever switch/router it is that the LAN hub is connected to. A diagram of a possible configuration is shown in Figure 3.4.

Figure 3.4: A laboratory



Note that the server will need to be in a physically separate place. This might be as simple as a locked cupboard inside the room.

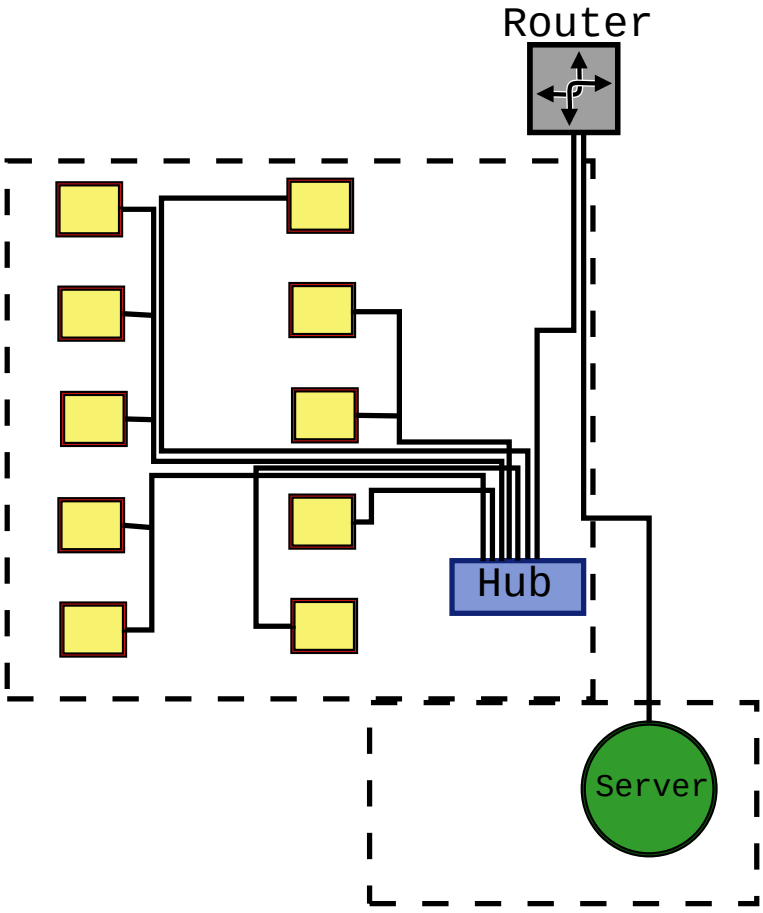
The Laboratory LAN needs to be treated both as a target for security threats from which it needs to be protected, and as a source of security threats which need to be controlled.

As usual, the first strategy to be considered is the use of a gateway which stands between the laboratory and the network at large. This is likely to be too expensive. Hence, a cheaper strategy which might be satisfactory should

be considered. This is to put appropriate filtering rules in the router by means of which this laboratory connects to the rest of the network. This strategy should be adequate in most cases.

A more secure arrangement might also separate the server from the workstations behind a router, as in Figure 3.5. There are two choices concerning this router: it could be a special router just for the purpose of securing access to the server. Such a router could be based on the Linux operating system for example, and therefore might be quite cheap to implement. It could be a general purpose router used for other purposes as well. In the latter case the fact that all the lab traffic passes through a central router would represent a significant design weakness because directing all the lab traffic through a central router is completely unnecessary.

Figure 3.5: A laboratory



If a switch/router is used, for this central router, it would be possible to carry this heavy load without too much difficulty, however there appears to be some fundamental weakness in this type of design for a laboratory in other respects. In particular, the *reliability* of this design will be much lower than a design in which the path between the server and the workstations passes through a minimum number of intermediate points.

A strategy which might be worth considering is to increase the capacity of the link, or links, by means of which the server, or servers, associated with a laboratory communicate with the workstations. For example, the server could have two, or more, network interfaces.

Another issue worth considering whether the servers are shared or allocated separately for each laboratory. The high bandwidth traffic (for booting, file-sharing, and printing) between a server and its clients can be confined to the smallest possible network by allocating a separate server for each laboratory.

It appears that some degree of compromise may be necessary in a satisfactory laboratory design. Traffic/performance considerations suggest strongly that the server(s) should be placed as close to the workstations as possible. Reliability considerations support this same arrangement. Security considerations, on the other hand,

suggest placing a barrier, a router, between the workstations and the server(s).

This balance between performance, reliability, and security, all mediated by the important consideration of cost, can be expected to arise in other examples as well.

In this particular case, we need to ask ourselves: what are the security weaknesses of a server which require that a server be protected from the workstations it serves which cannot be dealt with by careful configuration of the server? If no such issues can be found, it is clear that locating the server(s) in close proximity to the workstations, as far as networking is concerned, will be the right approach. □

Example 3.14. A School

A school is another rather cost-sensitive example, however efficiency and performance issues are nevertheless important.

Throughput on the link between a school and the Internet at large is likely to be an issue. It is probably inevitable that the external link from a school is overloaded, at least for certain periods of time.

An extremely important strategy for adoption is the use of a caching proxy server. This proxy server should cache internet access for the entire school, including staff as well as laboratories and individual students. The use of such a server has the benefits of reducing Internet costs (depending on the charging regime of the ISP), speeding up access to relevant parts of the Internet, which are being access by other staff and students, and, finally, improved Internet access even for staff, or students, with interests completely disjoint from those of other staff or students, because of the greatly reduced traffic.

In some schools, most staff and students have their own independent computer access to the school's network. In particular, staff and school students may be allowed to connect their own computers to the school network at a large number of different places. This gives rise to special security issues because:

(a) the physical location of the port to which the computer is connected is not a good guide to the class of user;

(b) it is inappropriate to rely to a significant degree on the specific configuration of the computers being attached to the school network.

A strategy which uses the ethernet address of the computer being attached to select which virtual LAN the computer should join might be appropriate.

Another important security issue in a school is the fact that the administrative and academic records of the school are of critical value to the school and access to these records must be very strictly controlled. Because of the critical importance and sensitivity of this data, multiple layers of protection will need to be put in place. In particular, authentication will be necessary whenever access is granted, but, in addition, access should be limited to computers on the basis of port location and/or MAC address (ethernet hardware address) as well.

The level of aggregation of traffic inside a school network is not high. As a consequence, single events – e.g. the backing up of a server over the network – have the potential to saturate the network. When choosing the speed of the network, the number of interfaces by which the servers are connected, and so on, certain scenarios should be kept in mind.

A good strategy for reducing network load and increasing the efficiency of the network as a whole is the segregation of certain classes of traffic from the main network. In particular, where possible, traffic between laboratory computers and the servers with which they are associated should be confined to the smallest possible subnetwork. □

Example 3.15. A University Campus

All aspects of performance – reliability, loss, delay, and security – are important in this Example, although there are networks with more stringent standards.

All the issues which were mentioned in the school example remain applicable, only more so. The security issues are similar but more intense. Administrative and academic data must be protected against loss, corruption, and access by unauthorised individuals.

A sensible strategy is to broadly distinguish three classes of user, service, and traffic: students, academic staff, and administrative staff, and their associated services and traffic. This subdivision of network activity into three classes is useful but imperfect because there are unavoidable intersections between these classes: academic staff

with administrative responsibilities, administrative staff who are also students, academic data which is stored in administrative computer systems, and so on.

Physical separation of computer resources and traffic of the three classes is not feasible, and institution of such a separation would be highly inefficient and inconvenient because of the economic cost of duplication of basically the same services and equipment and because of the prevention or inhibition of legitimate and necessary communication between the three sub-networks.

Logical separation of sub-networks is likely, however, to form a good starting point for a campus security plan. As a starting point, all users and hosts of the campus can be classified as belonging to one of the three classes: *student*, *academic*, or *administrative* and the routers of the campus network can then be configured to prevent, by default, communication between the three classes of user and host.

On top of this basic framework, many special cases where communication between one sub-network and another will then need to be allowed. In particular, broad classes of access to student networks should be granted to academic users; academic staff with administrative responsibilities will need access to administrative responsbilities. □

Example 3.16. A State-wide Retail Organization

The paramount performance issue for a State-wide organisation is security, and this is likely to remain the case for some time. The risk of security breaches is high, the temptation to mis-handle security is high, and the expertise and resources to handle security well are in short supply

On the other hand, this is an example where cost will remain an important consideration. For this reason, the use of the Internet to provide communication between sites is attractive.

Because of reliability considerations, it would be wise to have alternative communication facilities available at all sites. The alternative facility would not necessarily have to be available simultaneously with the primary facility, and might be available only after a delay, and even then only with somewhat reduced capacity and capabilities relative to the primary system.

The delay performance of the network might be an important issue, depending on the services which are carried on the network. It is unlikely that propogation delay will be sufficient to cause significant problems. On the other hand, if low speed lines are used from some sites, transmission delay could be a problem.

The level of aggregation of traffic within a retail organisation network is likely to be low. For this reaon, the capacity of the network is likely to be tested in single events, such as a state-wide sale, a stocktake, or a major reorganisation of staff. It would be wise to explicitly target a suitable scenario as the a way to determine appropriate choices of equipment and their capacities.

An opportunity will often exist to use the in-house network to carry in-house telephone communication as well as data. In the case where in-house telephone traffic is targetted as well as data, the Internet might not be able to provide adequate capacity for the primary communication services from which the network is composed, however in this case, the Internet might nevertheless be useful for providing backup communication services.

□

Example 3.17. A National Internet Service Provider

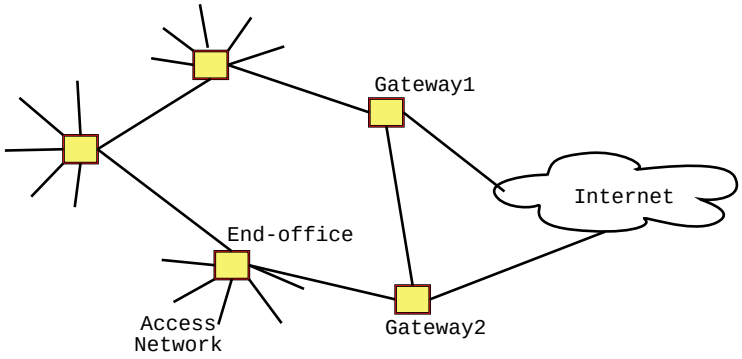
A national ISP can be anything between a re-badging of the ISP services provided by someone else, typically a large carrier, to a subsidiary service of a national carrier, fully provided with its own communication equipment. However, we do not need to analyse the extreme examples because in the former case, the rebadging, there is no actual network and so nothing for us to study, while we will deal with the latter case in the next example.

The case of interest is where the ISP has its own end-offices, to which its clients are connected by dial-up lines or fixed connections, and in which a good deal of the rest of the network is leased from another communication provider.

In a way, the ISP services of a national carrier also take this form, because it is likely that the ISP part of a carrier will have to obtain communication services from the rest of the organisation in much the same was as if they were being rented from another provider.

We can divide the network of an ISP into components as follows: there will be a local network at each of the end-offices, an access network, by means of which the ISP clients connect to the end-offices, an *inter-office* network by means of which the different offices of the ISP communicate with other, and, finally, a gateway network, by

Figure 3.6: A national ISP network



means of which this ISP is connected to the Internet at large. A network of this sort is depicted in Figure 3.6. Note that the intra-office networks are not shown in this diagram.

Each of these separate networks has its own performance constraints and issues. If one of these networks provides inadequate services in regard to any of reliability, loss, delay, or security, clients will be affected and may switch to a different provider.

The *capacity* of these networks is critical to both the perceived performance of the service as a whole, and also to the cost of running the service. If excess capacity is installed and has to be paid for, the financial viability of the business could be fatally affected. On the other hand if clients experience service which they feel is inadequate and choose to switch to another provider, the business may also be severely damaged. Hence, choosing the appropriate capacities for the access, inter-office, and gateway networks is of critical importance. On the other hand, the intra-office network should not, in principle, be easy to design in such a way that it has more than sufficient capacity for all the demands placed on it, without incurring significant costs. This is because this is the only one of the four sub-networks making up an ISP which can be located entirely within a single building, and therefore does not require leasing of any facilities from an external organisation. Nevertheless, a degree of care should be exercised to ensure that traffic local to the ISP end-office does not impinge upon client traffic in any way.

The level or aggregation of traffic in a national ISP is higher than in the preceding examples, but still not all that high. In most cases a Gaussian model can probably *not* be used to provide accurate estimates of loss and delay performance. However, the most important design problems in the case of an ISP can be adequately addressed by taking a pragmatic approach.

We will deal with all these issues in more detail in subsequent chapters, however let us quickly review the performance issues and the consequent design problems facing an ISP. These can be addressed most conveniently' by considering the four subnetworks one-by-one.

The Access Network

A significant part of the access network is often provided via the telephone network of the telecommunication company which operates in the area of the end-office. The ISP provides a terminal server and leases a number of telephone lines which are connected to this terminal server. One approach is to also connect a dedicated modem to each of the incoming lines. Nowadays it is more common that these modems are actually provided in a highly packaged form so that the connection between the incoming line and the receiving modem is not visible as a separate peice of physical equipment.

But there are other important options for the access network as well. Customers can install dedicated lines connecting their premises to the ISP, either by leasing such a line from the telecommunications company, or, if physical access is not a problem, by laying down an appropriate cable. Finally, depending on the regulatory framework, it may be possible for the ISP to place communications equipment in the local telephone exchange to enable customers to gain access to the ISP directly.

The Intra-office Network

The intra-office network is likely to be only a little larger than a home network. Because of the "mission critical" nature of this network, it is probably wise to install high speed (more than 100 Mbit/sec) equipment and to have backup hardware available on site. This network will have the role of connecting together the terminal server and all the service specific hosts which need to be provided at an end-office, typically a DNS server, a web server, a mail server, an authentication server an accounting and logging server, and a router. Some or all of these servers will need to be duplicated, for reliability reasons. The backup servers will need to be connected to the network at all times. It might not be necessary to provide *all* of these services on separate hosts. As an extreme example, the DNS, web, mail, authentication, and accounting servers could all be co-located. It is virtually mandatory, for reliabity reasons, to have a backup host for providing these services in the event of a single failure. It is not necessary for *every* site to have all of these services on-site. However, it should be kept in mind that the more centralised the system becomes the more prone it is likely to be to suffering widespread catastrophic failures.

The Inter-office Network

The inter-office network might be no more than a series of links connecting each end-office to the gateway. However, when the ISP grows beyone a certain size, it will become appropriate to provide multiple paths between sites and the gateway(s), and there will be cost advantages to connecting some sites to the gateway(s) via a series of hops.

Unless the ISP is very large, and this is unlikely to be the case for more than the largest few ISP's, the level of traffic across the inter-office network which is not either coming from or going to the gateway(s) will be insignificant.

It is in the interests of the ISP to maximise the degree to which traffic leaving the ISP can be diverted to stay within the inter-office network, because external traffic probably incurs a charge. The connection between the ISP and the Internet probably incurs a leasing cost and a usage related cost.

A very important technique for reducing the amount of external traffic and replacement by inter-office traffic is the use of one or more Proxy servers. Customers of the ISP can be requested, or even forced, to use the proxy server. It might be useful to have more than one proxy server, and to provide this proxy server with a considerable amount of memory and hard disk space in order to maximise its effectiveness.

The communication facilities of the inter-office network can be obtained by leasing lines from a carrier, installation of land-lines (under certain circumstances), installation of microwave towers, or leasing of satellite or other radio based communication services. A mixture of all the above might also be usefully employed.

There is unlikely to be any potential to make use of a different public ISP to provide communication services in the inter-office network, except possible for use as a backup service.

The Gateway Network

In many cases there will be only the one gateway host, although there could be considerable reliability advantages in having two gateways. It would be even better if these gateway hosts were connected to different ISP's although this could give rise to some difficult routing problems. In most cases nowadays, ISP's obtain their Internet address space from their "parent ISP's", and this relationship between junior ISP and parent ISP makes routing easy to configure. If a junior ISP is connected to two separate parents, special arrangements would need to be made to ensure that routing of packets was successful even when the primary ISP was off the air.

The choice for the capacity of the gateway link(s) is very important. Perfect performance is unattainable and nearly perfect performance is not necessary either. A practical middle-ground selection of a compromise between performance and cost seems to be necessary.

The larger the amount of traffic, the easier it will be to provide adequate performance for customers at reasonable performance levels. Although the level of aggregation of traffic is probably not sufficient to justify a Gaussian model for guiding the selection of gateway capacity, the Gaussian method should be adopted for selecting link speed. For example, a practical approach might be to decide that the link should be chosen with sufficient capacity that during the busy period no more than one five minute period in every hour experienced load levels which would cause traffic to be lost or suppressed. This is a pragmatic choice, reflecting the highly cost-sensitive nature of this

type of business. Better quality of service would be nice, but it comes at a price. The choice of a five minute sampling interval here reflects the fact that most of the traffic being carried is of the "best effort" variety. □

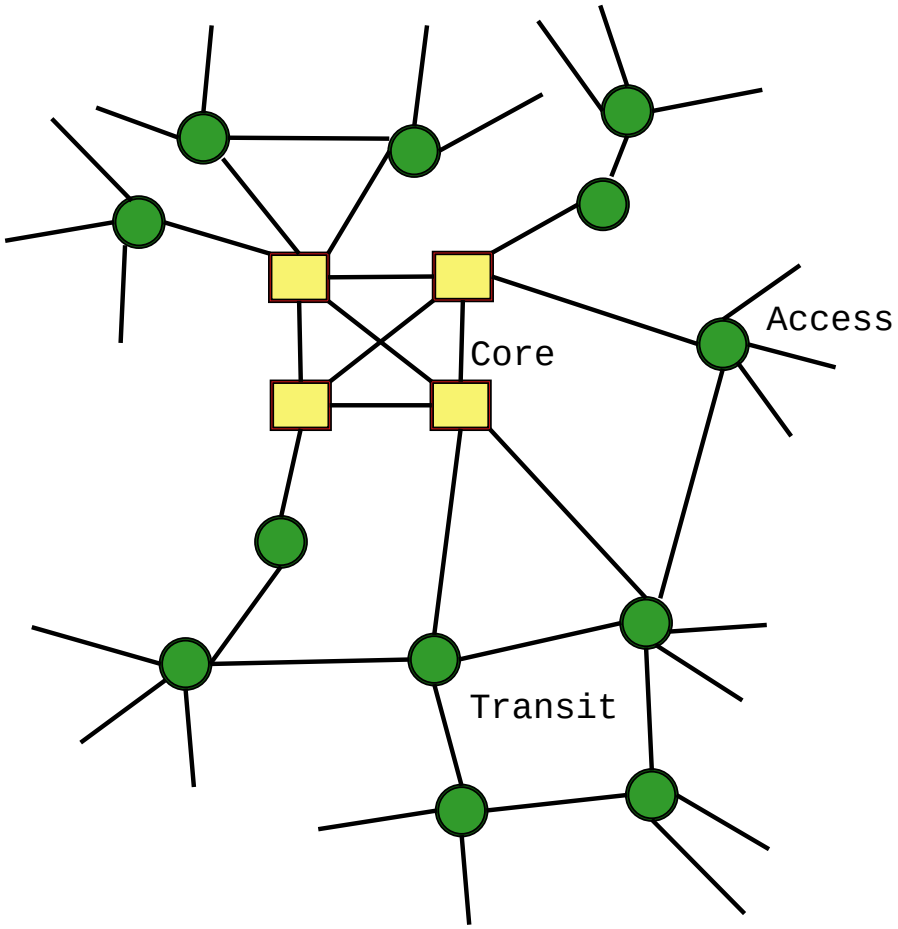
Example 3.18. A National Carrier

This is the example which includes all the other examples as special cases.

A national carrier will typically own and operate several networks which overlap and interract in a variety of ways. Some of these network may be built and maintained purely for use by other networks. The primary architectural principle governing the relationship between networks is the concept of *layering*. Two networks in a layered relationship, interract by the lower layer providing a service to the upper layer.

In addition, many large networks need to be subdivided into parts according to a measure of how the particular part is located relative to the central core or the periphery of the network. The customers (also known as subscribers or clients) of the services of a carrier are located on the outer edge of the network, whereas the services are located at the centre, or in the *core* of the network. Figure 3.7 depicts a network of this form.

Figure 3.7: A Carrier's network showing the different parts



This network is sub-divided into the access portion, the transit portion, and the core network. A structure like this applies to a variety of networks including telephone networks, some public TCP/IP networks and cable television networks.

The different portions (core, transit, and access) have different performance constraints, different cost structures, and different levels of aggregation. As a consequence the design principles for the different parts of such a network are quite different.

The Access Network

In the case of telephone networks, the access network is formed primarily by multipair cables which effectively connect each home to the local telephone exchange by an individual pair of wires. The pair of wires (the *local loop*) is usually completely unshared and simply connects the telephone or telephones of a house to an individual port at the telephone exchange.

However, the pair of wires goes through a series of transitions between the terminal and the exchange. The cables of the access network contain multiple pairs, anywhere from 16 to hundreds of pairs in the one cable and in the portion of the access network nearest to the telephone exchange, the access network makes use of higher density (more pairs) cables while in the more remote parts of the access network, lower density cables are used.

The access portion of the Internet effectively reuses the access portion of the telephone network. Most frequently this is achieved by means of modems. One of the advantages of modems is that by means of a modem a terminal at one point in a telephone network can connect to a terminal at a different location even if this necessitates a connection through one or more intermediate telephone exchanges. However, this flexibility has a cost, and the cost is that there is an upper limit to the speed of communication through a modem.

An alternative way in which the telephone network can be used to provide TCP/IP access to the Internet is by means of a digital connection between terminal equipment at the home or office connected to terminal equipment in the local exchange. The Integrated Services Digital Network (ISDN) is one example of this approach, and another approach based on the same idea is the Asynchronous Digital Subscriber Loop (ADSL). ISDN, as the name suggests, is more than just an access technology, whereas ADSL is just an access technology. ISDN, because of its older heritage, operates at a slower bit rate – 144 kbit/sec in both directions. ADSL operates at a variety of speeds depending on line quality and length, but a typical speed is 2 Mbit/sec in one direction and 64 kbit/sec in the other direction. [21]

The access portion of the networks managed by a telecommunications company tends to consume a very high proportion (in excess of half) of its capital equipment budget. The investment by telecommunication companies into access networks is very, very high. The technology used in these networks is rather slow-moving by comparison with the other parts of their networks. Replacement of cables or the equipment installed in ducts, pits, and manholes is often too expensive to be even considered as an option. Therefore, the access portion of a network is usually designed and installed many years before it is even used and not upgraded or replaced for decades.

The services carried on the access network are largely segregated one from another and one customers use tends to be completely separate from another's.

Transit and Core Networks

By contrast, the transit and the core components of a carrier's networks are more dynamic, as regards the changing of technology and the amount of work which is put into maintenance and upgrading. The equipment in these parts of the networks is often shared over many services and many customers. For this reason, the design and maintenance choices that need to be made in these components of a network are more difficult, interesting, and significant.

The nature of the traffic in the core and transit portion of a network naturally depends to a degree on the types of services which are being carried, however,.increasingly, carriers now have the option of converting all traffic to a common form (packets) and merging these together onto a common network.

Furthermore, the models discussed earlier in this chapter are sufficiently broad to be applicable, at least to an approximate degree, to any combination of traffics which might be carried in these parts of a network.

□

3.8 Closing Comments and Summary

In this chapter we have studied traffic, in much more detail than previously. We discussed three traffic models: the Poisson point process, the Gaussian process, in which work is measured as a real-valued quantity, and the Poisson-Pareto Burst Process, which is the model which attempts to mimic the true network as closely as possible.

Techniques for analyzing and designing simple networks based on these models were formulated. Much more could be said, however for the purposes of this book, it would be best to reserve some energy for some of the other

practical matters dealt with in the remaining chapters

References

[1] William Stallings. *High-Speed Networks: TCP/IP and ATM Design Principles*. Morgan Kauffman, 1999.

[2] W. Feller. *An introduction to probability theory and its applications*, volume 2. John Wiley and sons, second edition, 1971.

[3] Kai Lai Chung. *A course in probability theory*. Academic Press, 2001. 519.2 Chu.

[4] Milton Abramowitz and Irene A. Stegun. *Handbook of Mathematical Functions*. Dover, New York, 1970.

[5] Iridium Satellite Solutions. Iridium – home. Internet Web Site. http://www.iridium.com/.

[6] Sally Floyd and Van Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, 1993.

[7] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, 2:1–15, 1994.

[8] R. G. Addie, M. Zukerman, and T. Neame. Broadband traffic modeling: simple solutions to hard problems. *IEEE Communications Magazine*, 36(8):88–95, August 1998.

[9] Wayne L. Winston. *Operations Research, Applications and Algorithms*. PWS-Kent Publishing House, 1991.

[10] R. G. Addie. On weak convergence of long-range-dependent traffic processes. *Journal of Statistical Planning and Inference*, 80(1-2):155–171, August 1999.

[11] R. G. Addie, P. Mannersalo, and I. Norros. Most probable paths and performance formulae for buffers with Gaussian input traffic. *European Transactions on Telecommunications*, 13(3):183–196, May-June 2002.

[12] Gennady Samoridnitsky and Murad Taqqu. *Stable Non-Gaussian Random Processes*. Chapman and Hall, 1994.

[13] R. G. Addie and M. Zukerman. An approximation for performance evaluation of stationary single server queues. *IEEE Trans. on Communications*, 42(12), December 1994.

[14] Ilkka Norros. A storage model with self-similar input. *Queueing Systems – Theory and Applications*, 16:387–396, 1994.

[15] R. G. Addie, M. Zukerman, and T. M. Neame. Fractal traffic: Measurements, modelling and performance evaluation. In *Proceedings, IEEE Infocom 1995*. IEEE, April 1995.

[16] Ake Arvidson. The effect of end-to-end protocols on loss performance in tcp/ip networks. In P. Key and D. Smith, editors, *Teletraffic Engineering in a Competitive World*, volume 3B of *Teletraffic Science and Engineering*. 16th International Teletraffic Congress, Elsevier, 1999.

[17] Xipeng Xiao and Lionel M. Ni. Internet qos: A big picture. *IEEE Network Magazine*, 1999.

[18] R. Guerin and Ariel Orda. Qos-based routing in networks with inaccurate information: Theory and algorithms. *IEEE/ACM Transactions on Networking*, 7(3):350–364, June 1999.

[19] K. Nichols, V. Jacobson, and L. Zhang. A two-bit differentiated services architecture for the internet. Technical Report RFC 2638, IETF, 1999.

[20] Ilkka Norros and Petteri Mannersalo. A most probable path approach to queueing systems with general gaussian input. *Communication Networks*, 2002.

[21] Padmanand Warrier and Balaji Kumar. *XDSL Architecture*. McGraw-Hill, 2000.

Chapter 4

Measurements

In this chapter we shall identify the most important quantities to measure. We shall learn how to be able to measure these quantities on a Local Area Network with appropriate software. We will also consider how to measure traffic and performance on paths leaving a local network to join the Internet. We shall also study the statistical procedures which must be used to process measured data in order to estimate useful parameters which describe traffic in a manner which can readily be used for analysis and design.

The basic measurements we need to make may be divided into two categories: *performance measurements*, and *traffic measurements*. We need to make the former measurements in order to directly observe how well our networks perform. We need to make the latter measurements in order to understand better how our clients, the users of the networks we are analyzing and designing, want to use their networks, and also to understand why our networks deliver the performance that they do.

4.1 Traffic Measurements

We need to measure quantities of traffic – but what does that mean? We need to be more specific.

The traffic measurements of interest are:

- packet arrival rates (in packets/sec);
- packet lengths (in bytes);
- bytes of throughput (in bytes/sec).
- utilization
- call/connection arrival rates
- call/connection holding times

Utilization is a dimensionless quantity which measures the *proportion of time during which a resource is busy*. The term *occupancy* is also sometimes used in place of utilization. The resource in question could be a transmission link, a router, or a server. In the case of a transmission link, we could think of ourselves as measuring at a certain point on the link, and observing all the nanoseconds when the link was busy transmitting a packet, and all the nanoseconds when it was not busy transmitting a packet (although it might be transmitting a synchronization pattern instead). The number of busy nanoseconds divided by the *total number* of nanoseconds in the observation is the *utilization* of that link. Similar definitions apply to a router or a server.

All of these quantities can be measured using various different sampling intervals. However, so long as we use an interval-independent measure, such as packets/sec, the quantity we are estimating should be the same irrespective of the interval used, aside from the possibility that the estimation errors might be different for different rates of observation.

On the other hand, these quantities are random – we should apply some statistics to their measurement. In particular, we need to know not just the average value of packets/sec over a period of time, but we should also

estimate the *standard deviation* of this quantity. And when we do this it *is* important how long a time interval we use for its measurement.

One of the reasons that we like to estimate the standard deviation of a quantity like the packets arriving in an interval is that this allows us to determine over how long an interval we should make observations, in order to produce a satisfactory estimate.

A more important reason for measuring the standard deviation of traffic is that the higher the standard deviation, the more difficult it is to carry the traffic on a link or to deal with this traffic when it's passing through a router. So, the standard deviation is important in its own right, not just in its influence on measurements of the mean.

4.1.1 Measuring the variance or standard deviation of traffic

Now, as we observed before, the time interval over which we measure the mean, for example, packets/sec, does not affect the outcome. Suppose we choose to measure the packets arriving in successive 10 millisecond intervals. If we continue measuring for 10 seconds, we will have made measurements in 1000 intervals. The observed packet rate over this period can be obtained as the total number of the packets in each successive interval, divided by the total number of seconds of observation. The same answer will be obtained if the intervals are of length 100 milliseconds, so long as the start and finishing time of the observation period are the same.

But this is not the case when we measure the standard deviation of a quantity. The standard deviation of the number of packets arriving in an interval of length 10 milliseconds is not the same as the standard deviation of that quantity when measured over an interval of length 100 milliseconds.

If the quantity in question was statistically independent from one interval to the next, we do regain a natural way to measure standard deviation. In fact, it seems to be better to measure *variance* (the square of standard deviation). And the *variance* of a measurement over two intervals, in this case of statistical independence, turns out to be the *sum* of the variance over the two separate intervals, so it might make sense to talk about *variance per second*. This variance per second measure would then be independent of the time interval over which it was measured. Variance measured over an interval of length t follows, in this case, the law:

$$V(t) = \sigma_1^2 \times t \tag{4.1}$$

for some σ_1.

On the other hand, if the measurements in successive intervals were statistically *dependent* to the highest possible degree – totally correlated – then the *standard deviation* of the number of packets arriving in an interval would be proportional to the length of the measuring interval, and it would be appropriate to talk about *standard deviation per second*. Looking at the variance in this case we see that it obeys the law:

$$V(t) = \sigma_1^2 (t)^2 \tag{4.2}$$

for some σ_1.

However, in real networks, traffic in successive intervals is neither statistically independent, one interval from the next, nor totally correlated. Somewhat surprisingly, however, empirical studies show [1] that it does appear to follow a law a bit like the previous equations, (4.1) or (4.2). The law is:

$$V(t) \approx \sigma_1^2 t^{2H}$$

for some σ_1 and H, and for sufficiently large t. The parameter H is known as the *Hurst* parameter.

Thus, when we take our measurements, we do not necessarily have to make measurements over *every* time interval, but just over sufficiently many to be able to estimate the following parameters:

1. the mean;

2. the parameter σ_1, of the variance time curve;

3. the parameter H, of the variance time curve.

Example 4.1. Telephone Traffic

Telephone traffic is a traditional focus of attention and remains important because we still have large quantities of this traffic occupying our networks.

Telephone traffic is made up of *telephone calls* (*calls* for short) . We therefore model telephone traffic by a two stage process: first we model the process of arrivals of calls; then we model the duration of the calls.

Note that the concept that traffic is layered in this manner, *traffic made up of calls, calls made of packets, packets made of bytes, bytes made of bits*, applies to traffic other than telephone traffic. (Note: the packet layer does not usually exist here, although it will exist if the telephone traffic is carried over a packet network, including a TCP/IP network.)

Most Internet traffic follows a similar rule: *traffic is made of connections, connections are made of packets, packets are made of bytes*. The concept of a *connection* occurs in other traffic types as well and is really much the same thing as a call. Perhaps it would be more accurate to say that the term *connection* generalises the concept of *call* to other networks and to types of service other than telephony.

The call arrival process (and the connection arrival process also, in many cases) is well modelled by a Poisson process. The holding times of telephone calls are often modelled as following an exponential distribution:

$$P\{X > x\} = \begin{cases} e^{-x/h}, & x > 0 \\ 0 & \text{otherwise.} \end{cases}$$

The mean of this distribution is h, which is know as the *mean holding time*.

There is more than one way to measure telephone traffic. One can measure how many arrivals have occurred in a certain period of time. This provides an estimate of the *call arrival rate*, typically denoted by λ. Alternatively, one can measure the average number of active calls at any given time, typically denoted by a. The latter quantity is often referred to, simply, as the traffic, and the *unit* in which it is measured is, in effect, calls. However, in order to be a little more explicit, the term *Erlang* is usually used. Thus, if we say *the traffic is 5 Erlangs*, this means that on average there are 5 calls active.

The call arrival rate, λ, holding time, h, and the traffic, a, are related by the formula:

$$a = \lambda \times h. \tag{4.3}$$

The variance, σ_a^2, of the number of active calls at any one time is also of interest. The number of calls active at any time is Poisson distributed (this was shown in subsection 3.3.3), and so its variance is identical to its mean, i.e. $\sigma_a^2 = a$.

Let us now consider the mean and variance of the *bit-rate* of the traffic required to carry telephone traffic. The most conventional manner for encoding telephony as a digital signal requires a continuous 64 kbit/sec channel in both directions for a single telephone call. It follows that a Erlangs of telephone traffic will generate a bit stream with a mean rate of $64000a$ bits/sec and a variance of $64000^2 a$ (in bits^2/sec).

Measurements of variance tend to be much less intuitive than measurements of standard deviation, because the natural unit for a standard deviation is just the same as the unit in which the mean is measured. The standard deviation of the bit-stream associated with a Erlangs of traffic will be $64000\sqrt{a}$ bits/sec.

Because telephone traffic is relatively well understood and the standard model of telephone traffic is generally accepted as valid, the measurements required to quantify telephone traffic are not complex.

First of all, we must recognise the fact that the intensity of telephone traffic fluctuates during the day. Generally speaking there are two periods of approximately one hour in length, one in the morning and one in the afternoon, when telephone activity is at its highest. Traffic levels outside these times are much lower. However, it is traditional, and sensible, to design networks so that they provide good performance during these busy periods. Informally, we speak of dimensioning for the *busy hour*.

Given this background, the most important measurement to make is the average telephone activity level of staff making use of the telephone network during the busy hour. Activity levels may vary widely. Telephone operators probably exhibit activity levels close to 1. More conventional levels of activity are in the vicinity of 25%.

The total traffic generated by n telephone users with activity rate α will be $n\alpha$ Erlangs.

In many cases we will need to distinguish between different classes of traffic; notably *internal* traffic (traffic between members of an organisation) and *external* traffic (traffic to the rest of the world) are particularly important classes. If the proportion of telephone traffic generated by the n users which is external is p, total external traffic will be $p \times n \times \alpha$. □

Exercise 4.1. Packet Voice

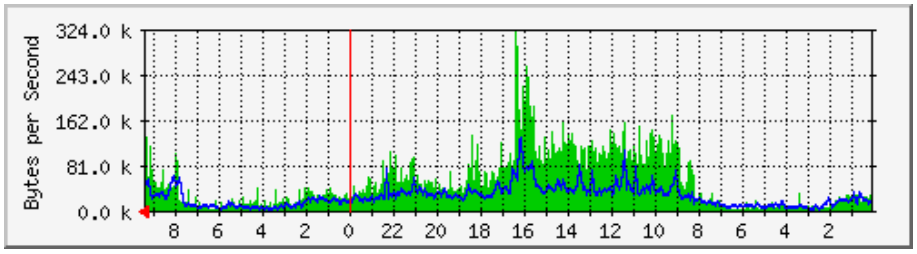
Suppose a voice signal is encoded in such a way that the voice signal is converted into a digital bit stream at the rate 16 kbit/s *during speech bursts* and no signal is generated during silence intervals. Suppose also that each speaker is speaking for 45% of the time. What is the mean and standard deviation of the bit-stream generated by 10 Erlangs (on average there are 10 simultaneous calls) of traffic? What is the mean and standard deviation of the bit-stream generated by 50 (50 calls on average) Erlangs of traffic?

Hint: because the total number of calls at any moment of time is Poisson distributed, and the *active* calls are found from the collection of all calls by selecting each call independently with probability 0.45, the number of *active* calls is also Poisson distributed. □

In principle, we should make measurements of each of the quantities of interest: connections, packets and bits or bytes and packet lengths.

In practice, it is difficult to take traffic measurements in this manner except at rather coarse time intervals. At present, the most precise measurements which are readily available tend to take a form similar to that shown in Figure 4.1.

Figure 4.1: A Traffic Plot from MRTG



The plot in Figure 4.1 was obtained by means of the freely available software MRTG (Multi-route Traffic Grapher. MRTG obtains these statistics from routers by inquiries to the router phrased in the Simple Network Management Protocol (SNMP).

4.1.2 Interrelationships

Consider now the following measurements made on a single link with capacity C measured in bits per second: Packet flow, in packets/sec, p, with mean μ_p and standard deviation σ_p; Bit flow, in bits/sec, b, with mean μ_b and standard deviation σ_b; Packet length, in bytes, L, with mean μ_L and standard deviation σ_L; and finally Utilization, U, with mean μ_U and standard deviation σ_U.

Each of these must be measured using a sampling interval Δt say, for the measure of packets, bytes and utilization.

Then the following interrelationships apply:

$$\begin{aligned} b &= p \times L \\ \mu_b &= \mu_p \times \mu_L \\ \mu_U &= \mu_b / C \\ \sigma_b^2 &= \mu_L^2 \sigma_p^2 + \sigma_L^2 \sigma_p^2 + \mu_p^2 \sigma_L^2 \end{aligned}$$

The last of these equations is a direct application of (3.9).

4.1.3 Connections and Bursts

The Poisson-Pareto Burst model of traffic proposed in Subsection 3.3.7 can readily be checked given the appropriate traffic measurements. According to this model, traffic takes the form of a collection of independent overlapping

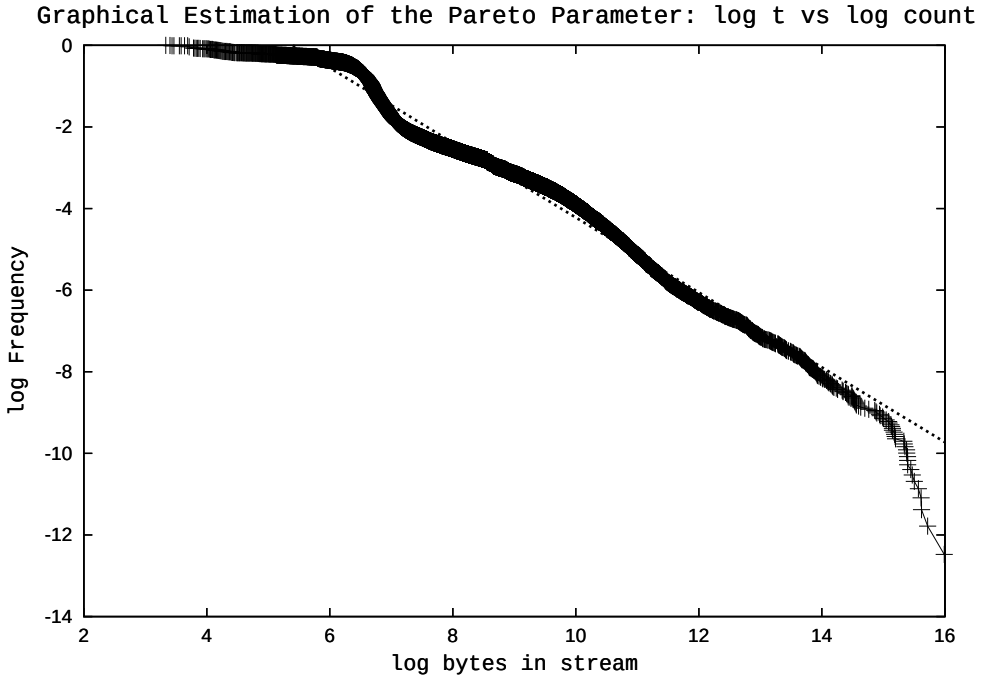
bursts. It is not difficult to recognise these “bursts” in TCP traffic. They are none-other than the TCP connections. It is also possible to identify *streams* of related data in related UDP packets, however these form a relatively small component of overall traffic at the moment, so we will not concern ourselves with this component of overall traffic.

On one day in April, 2001, the following traffic statistics were observed at the point where the traffic from a large educational institution joined the Internet:

Statistic	**Value**
Total bytes	943570079
Total TCP bytes	888714395
Total UDP bytes	50245452
Mean bytes per TCP stream	3840
Minimum bytes per TCP stream	35
Maximum bytes per TCP stream	38899054
Standard deviation of bytes per Stream	101423

These statistics support the view that the distribution of the number of bytes in a TCP connection has a very heavy tail. A plot of the empirical complementary distribution (i.e. instead of plotting the frequency with which values *less* than *x*, say, occur, we display the frequency with which values *larger* than *x* occur) of the number of bytes in a stream from this same data is shown in Figure 4.2 using log scale axes for both the x axis and the y axis.

Figure 4.2: Traffic Plot, Empirical Complementary Distribution for the Lengths of TCP Connections, in bytes, with linear fit



The plot also supports the view that the number of bytes in a stream is Pareto distributed. The slope of the complementary distribution, in the log-log space, is approximately -0.95, which suggests that the γ parameter of the Pareto distribution in this case takes approximately the value 0.95. This is a little surprising considering that we normally expect $1 < \gamma < 2$.

4.2 Estimation of Traffic Parameters

4.2.1 Estimation of mean

The obvious estimator for the mean traffic level is just the sample mean. For example, the mean arrival rate of packets can be estimated by counting the number of packets which arrived in an interval of length T, and dividing this number by T.

Similarly, the mean *bit-rate* is best estimated by counting the total bits transmitted over a period of time, T, and dividing this by T.

Measurement of *traffic* can similarly be acheived by sampling the traffic process and taking an average over the measured values. If the cost of sampling is significant, a compromise might be necessary in the choice of how many samples to take.

There is also the small matter of estimating the *accuracy* of these measurements. This immediately brings us to the question of how to estimate the variance of traffic.

4.2.2 Estimation of variance

In the case of telephone traffic, there is a fixed relationship between the mean and the variance of traffic. In the case of an arbitrary traffic flow, however, no such predetermined relationship exists. Also, in general, unlike in the case of the mean, measured variance varies with the measurement interval in a complex manner.

In effect, we cannot easily separate measurement of the variance of traffic at one time interval from measurement at another time interval. And if we are going to estimate the variance of traffic in a whole range of time intervals all at once, that brings us to the next topic, estimation of the Hurst parameter.

4.2.3 Estimation of the Hurst Parameter

The Hurst parameter is of considerable interest from a statistical point of view. It is rather more specific to the context of traffic than either the mean or the variance, although models of a similar form arise in other fields of study, e.g. in the study of flows of water in the Nile river a self-similar statistical model including a Hurst parameter has been used.

A variety of techniques for estimating the Hurst parameter have been proposed [1, 2, 3]. The first class of methods for consideration can best be described as *graphical*. It is not difficult to understand how these methods work. The simplest example is the method which computes estimates, $\widehat{\sigma}^2_{(m)}$ of the variance of the time series $\{X_k^{(m)}\}$ aggregated to the level m, for a sequence of values of m. The aggregated series $\{X_k^{(m)}\}$ derived from the series $\{X_k\}_{k=1}^N$ is defined by

$$X_k^{(m)} = \frac{1}{m} \sum_{j=(k-1)m+1}^{mk} X_j, \qquad k = 1, \ldots, \lfloor N/m \rfloor.$$

The estimator $\widehat{\sigma}^2_{(m)}$ is just the usual estimator of the variance of a time series, applied to the series $\{X_k^{(m)}\}$, i.e.

$$\widehat{\sigma}^2_{(m)} = \frac{1}{N/m} \sum_{k=1}^{N/m} \left(X_k^{(m)} - \overline{X} \right)^2 \tag{4.4}$$

These estimates of variance are then plotted against m, the level of aggregation, using a log scale for both the x-axis and the y-axis. If the series is self-similar or asymptotically self-similar we expect to see a straight line in this log-log plot and the *slope* of this log-log plot should be $2H - 2$. An illustration of this procedure is provided in Example 4.2.

Another important method for estimating H known as the Whittle estimator is actually a general method for estimating the parameters of a Gaussian time series. This method can be used for estimating H, in particular if we know (or make assumptions) for the form of the spectral density as a function H. The method works by solving

the following optimization problem, in which $I(\nu)$ denotes the periodogram of the time series:

$$\begin{aligned} \text{Maximize:} \quad & \int_{-\pi}^{\pi} \frac{I(\nu)}{f(\nu;H)d\nu)} \\ \text{Subject to:} \quad & f(\nu) \geq 0, \qquad \nu \in [-\pi,\pi] \\ \text{and} \quad & \int_{-\pi}^{\pi} \log f(\nu;H)d\nu = 0. \end{aligned}$$

The paper [4, pp177-217] explains and compares a number of graphical methods, the Whittle estimator, and some variations upon the Whittle estimator. In the examples studied in that paper, the Whittle estimator and appears to be accurate and reasonably robust by comparison with the other methods. On the other hand, the graphical methods have the advantage that if the data display an unusual feature which causes the estimation to fail, or to be less than perfect, this will probably show up in the graphs

Another significant method uses wavelets to estimate the Hurst parameter [3]. There is no reason to believe that this estimator should be more accurate than the Whittle estimator if the basic model is not too different from FGN, however there are some deviations from FGN which cause the Whittle estimator to perform quite poorly whereas the wavelet estimator of Abry and Veitch is able to provide a satisfactory estimator of H.

4.2.4 Estimation of variance (part II)

Now that we have discussed estimation of the Hurst parameter we can return to the issue of how to estimate the *variance* of the time series. If the data fits the FGN model exactly, the variance at one level of aggregation, together with the Hurst parameter, are enough to determine te Hurst parameter at all other levels of aggregation. This is a great advantage of the FGN model.

As illustrated in Figur 4.7, the assumption that variance does fit the model $V(t) = Ct^{2H}$ appears to be satisfactory.

Because a relationship of this form can be expected to hold, there are really only two parameters required to describe the entire variance-time curve. In a sense (for example, if we accept that a Gaussian model for the traffic is satisfactory), this means that we only need *three* parameters to describe the traffic: the mean, the variance of the bytes arriving in some time interval, eg an interval of 1 second, σ_1^2, and the Hurst parameter, H.

Of these parameters, the most interesting is H, but from the performance point of view the important paramters are actually the mean and the variance as measured in the time interval over which the traffic is typically buffered. For example, voice traffic cannot be buffered over a period of time longer than approximately 50 milliseconds without causing severe degradation, so the time interval interval should be 50 milliseconds, for voice. Best-effort traffic, such as a file transfer, or access to the web, can reaonably be queued for much longer periodn of time.

4.3 Performance Measurements

As well as measuring traffic, it is appropriate to directly measure network performance, such as delay, loss, reliability, and security flaws. The latter two are appropriately measured by keeping a log of events. The former, delay and loss, can be measured by means of some familiar tools.

4.3.1 Measurement of Loss and Delay

Exercise 4.2. Use Ping and Traceroute to estimate performance

Using `ping` and `traceroute` estimate loss, delay, throughput, the route, and the location of any bottlenecks, to the following locations:

`www.sci.usq.edu.au`,
`www.uq.edu.au`,
`www.altavista.com`,
`www.cam.ac.uk`,
`www.stat.ee.ethz.ch`,
and `www.spsselib.hiedu.cz`

Don't be surprised if the results are unexpected. Unexpected outcomes make experiments more interesting. If this happens, try to explain what has happened.

Where appropriate, estimate statistics of the observed parameters, e.g. mean, peak (in a certain sampling interval), and standard deviation. Also attempt to estimate the breakdown of delay into its different components. It may be difficult to estimate some of these quantities from the available information, so just make the best estimates that you are able to under the circumstances. □

Example 4.2. Internet Traffic Measurements

Quite a number of studies of traffic on the Internet have been undertaken [2] and fortunately the measurements from some of these studies are publicly available [5].

Plots of one of the traffic types (bytes in tcp connections) over a certain period of 2 hours aggregated at three different levels are shown in Figures 4.3, 4.4, and 4.5. These plots cannot be used to demonstrate any rigorous conclusions with regard to this data, let alone traffic data in general. However, they do illustrate a fact which has been observed *and* confirmed in some rather extensive studies of network traffic – namely, the fact that traffic in real networks exhibits variation at all time scales.

One of the graphical methods for estimating the Hust parameter is illustrated in Figure 4.6. The figure shows the log of the variance of the aggregated traffic data, aggregated over periods of length δ, as a function of the log of δ, for δ varying from 0.001 up to 200 seconds. The slope of this plot is an estimate of $2H-2$. The value of H suggested by Figure 4.6 is 0.83, which is typical of the values which have been observed in many studies.

In all of these figures, a consistent unit has been used for the y axis. For the plots of the data itself, the y axis is bytes/sec. In the case of Figure 4.6, the data plotted is the variance of the bytes/sec, as measured in intervals of a variety of lengths. So, in Figure 4.3, the data is aggregated into 1 second intervals and then the variance is calculated, by formula (4.4), in Figure 4.4, the aggregation is in intervals of length 10 seconds, and in Figure 4.5 the aggregation is in intervals of 100 seconds.

Another approach to estimating H is to estimate the variance, $V(t)$ of the total traffic in an interval of length t, as a function of t. This process is numerically very similar to the process used to generate Figure 4.6, however, although this approach doesn't produce new information, it has also been carried out and the results are shown in Figure 4.7. In this case, in addition, four different traffic traces, all from [5], are displayed.

The variance-time curves for all these traces are quite similar. The slopes of the curves vary somewhat, perhaps less than normal statistical variation, and the y-intercepts of the graphs vary somewhat more. □

Exercise 4.3. Set up MRTG

If you are already in the role of a network manager and you have access to a suitable server, set up MRTG (Multi-route Traffic Grapher) and use it to monitor the traffic levels on your network. □

One of the difficulties facing a network administrator is finding the tools for making appropriate measurements. The MRTG tool is convenient, and free, although it does not appear to be suitable for studying traffic in fine detail. Fortunately, this is not something we necessarily want to do all that often, although it is necessary in order to *study* the nature of traffic. Tools for making very accurate, fine measurements of arrival rates of large numbers of individual packets have been, on some occasions, specially developed.

However, there is also a free tool which can be used in some situations to make these sort of measurements: `tcpdump`. This tool is distributed with the Linux operating system, for example, and versions are available for most, if not all, variants of Unix. Versions are also available for windows.

`Tcpdump` can be used to observe *all* the traffic on an ethernet LAN (and variants are available for other sorts of LAN as well), so long as the network in question uses a broadcasts to deliver packets. This is the traditional approach in ethernet LAN's, however it is becoming increasingly common to install switched hubs, in which case `tcpdump` will not reveal much about the traffic on the LAN. It should be kept in mind, however, that even when switched hubs are used, because of the way TCP/IP runs over the top of an ethernet, there will be a considerable amount of broadcast traffic, and in fact this broadcast traffic is likely to be a significant influence on network performance.

`Tcpdump` cannot be used on a Unix host without root privileges. This is because access to the *promiscuous* mode of operation of an ethernet card is restricted, on Unix hosts.

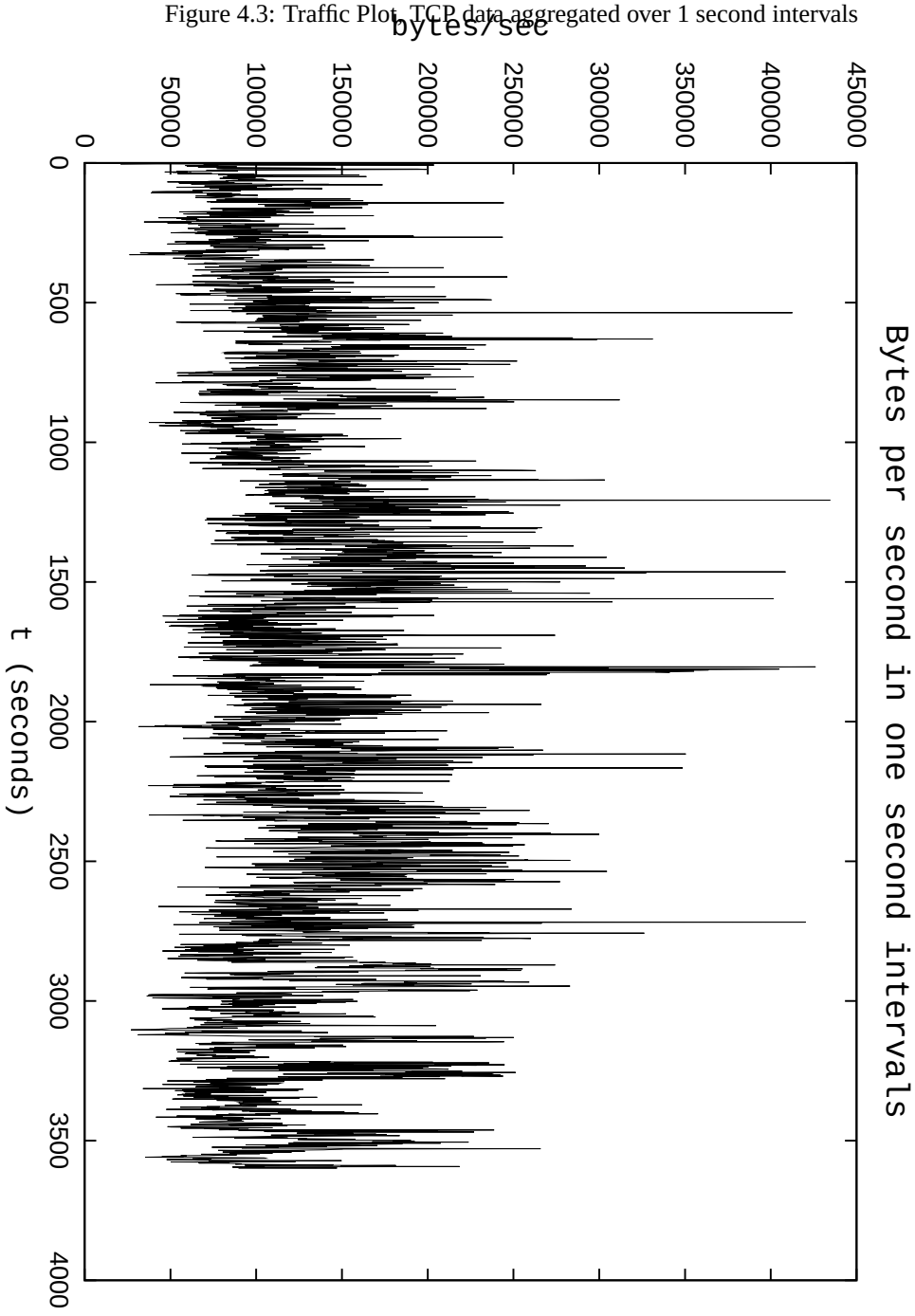
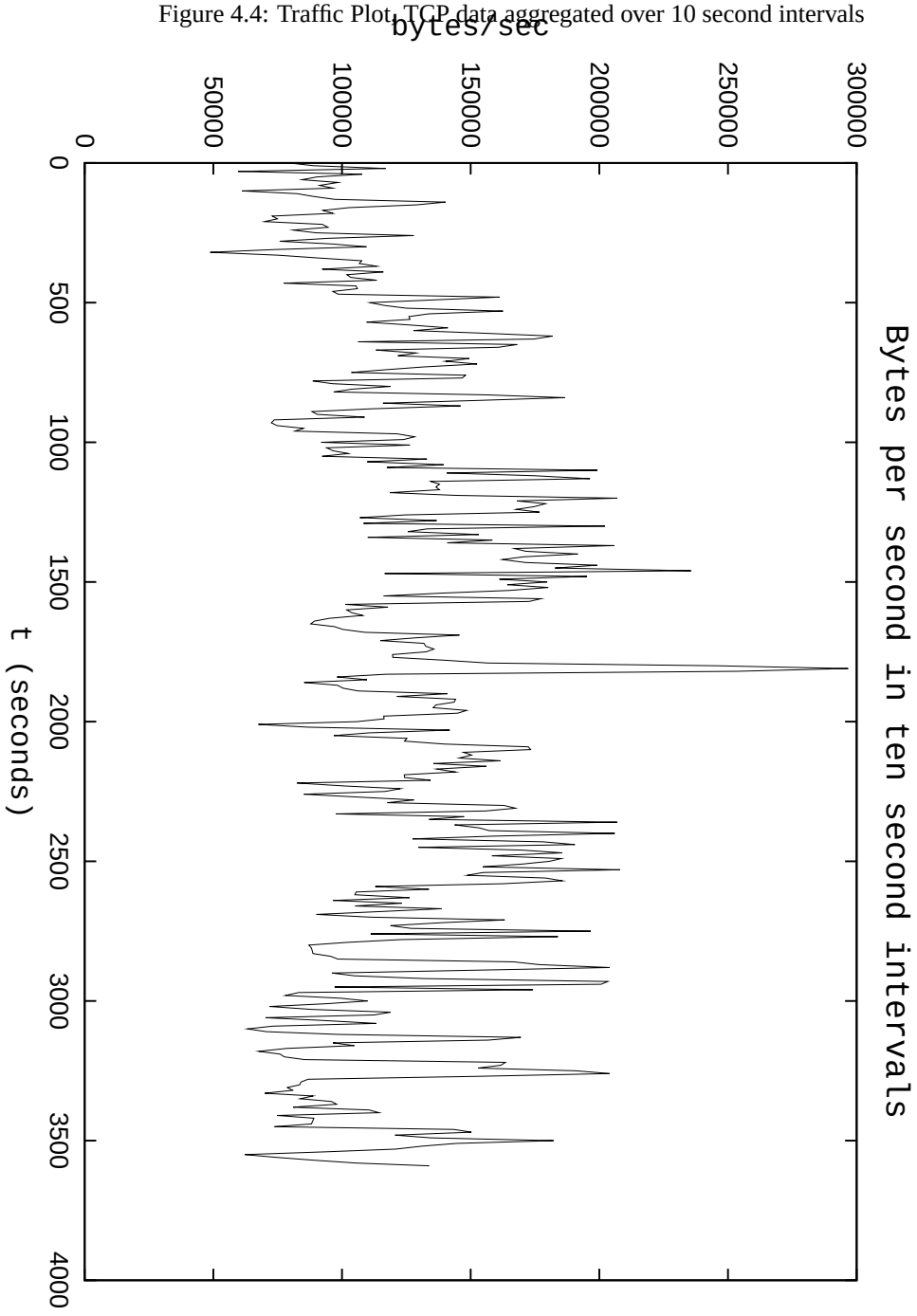


Figure 4.3: Traffic Plot, TCP data aggregated over 1 second intervals

Figure 4.4: Traffic Plot, TCP data aggregated over 10 second intervals



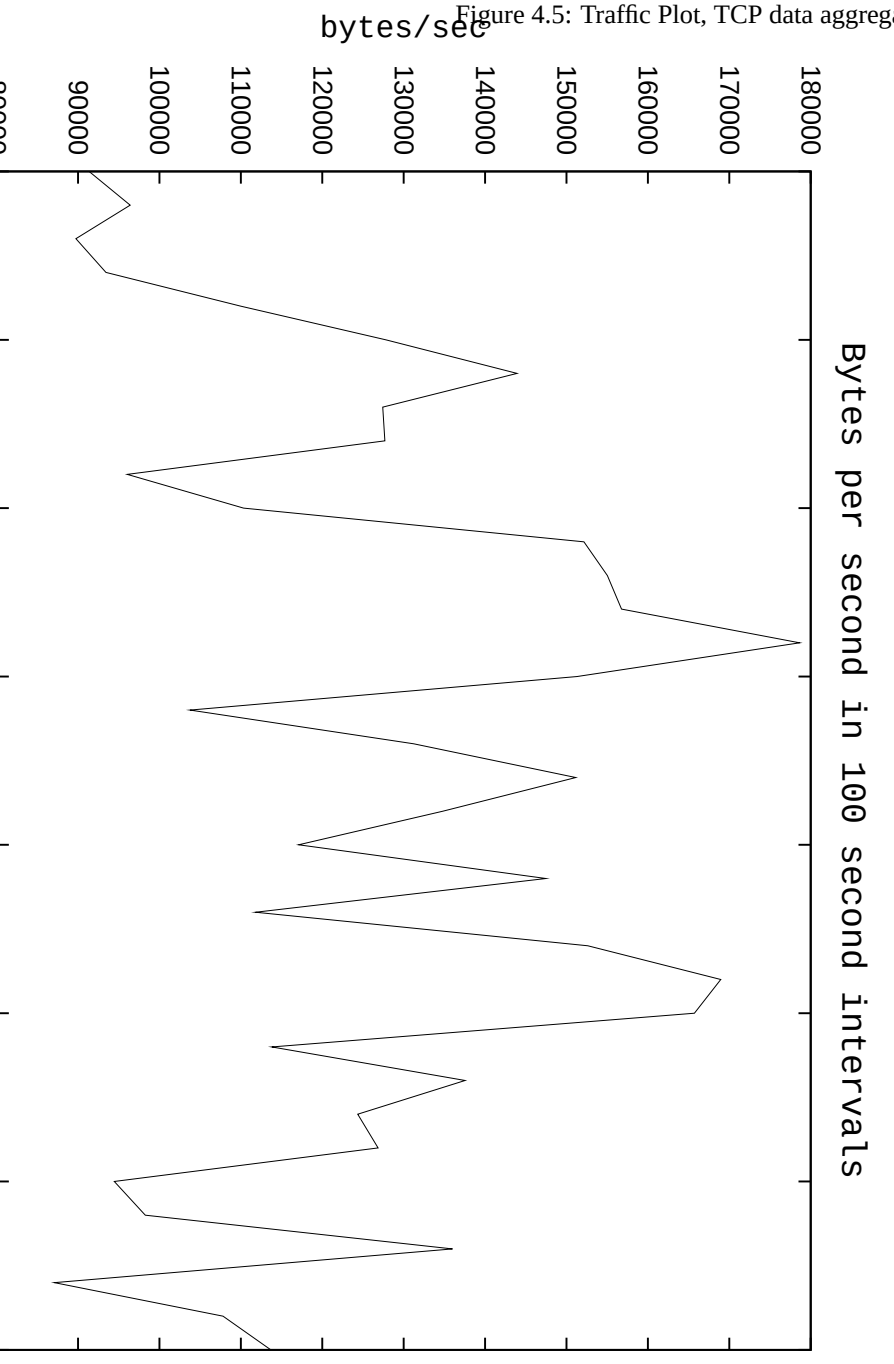


Figure 4.5: Traffic Plot, TCP data aggregated over 100 second intervals

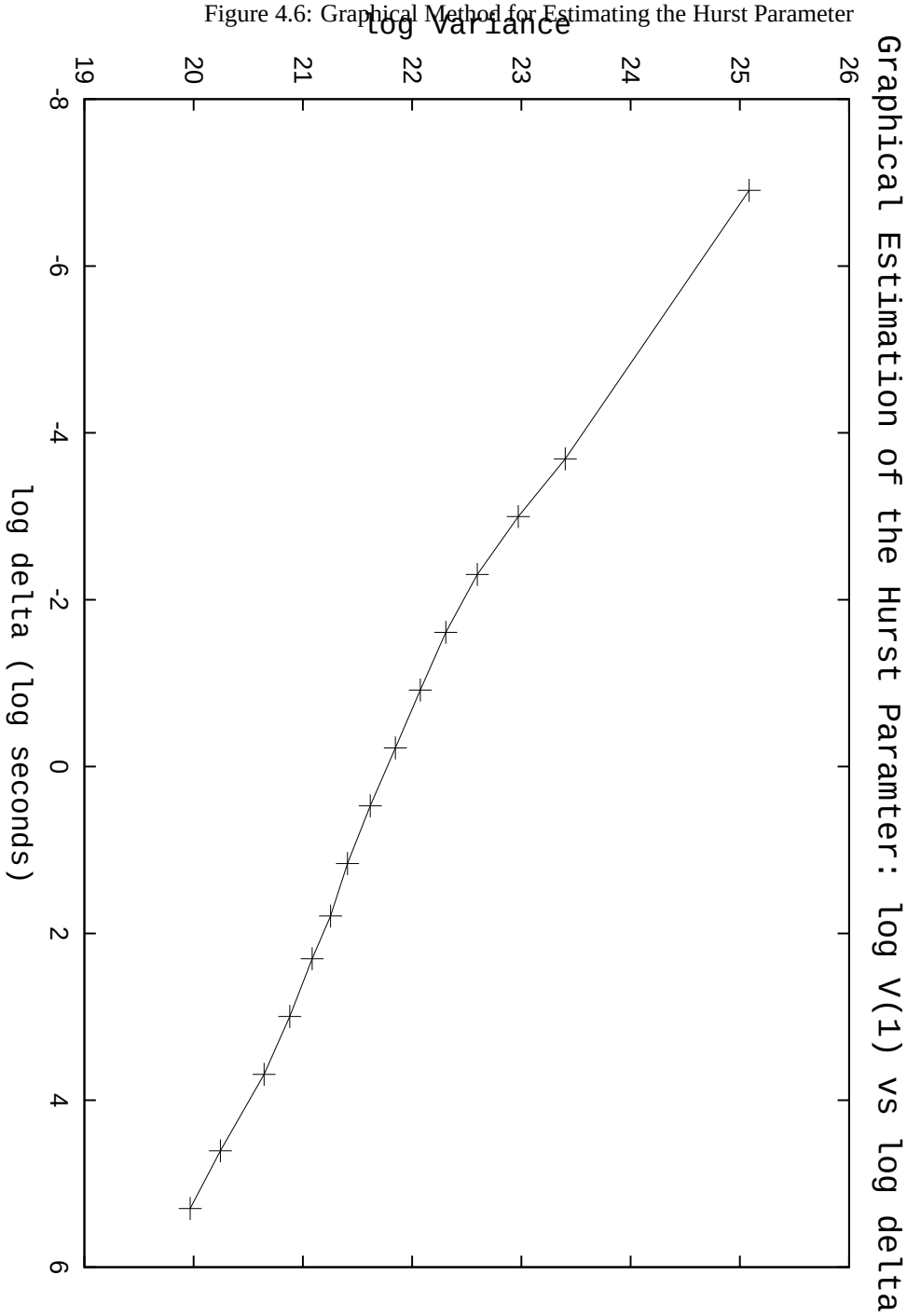
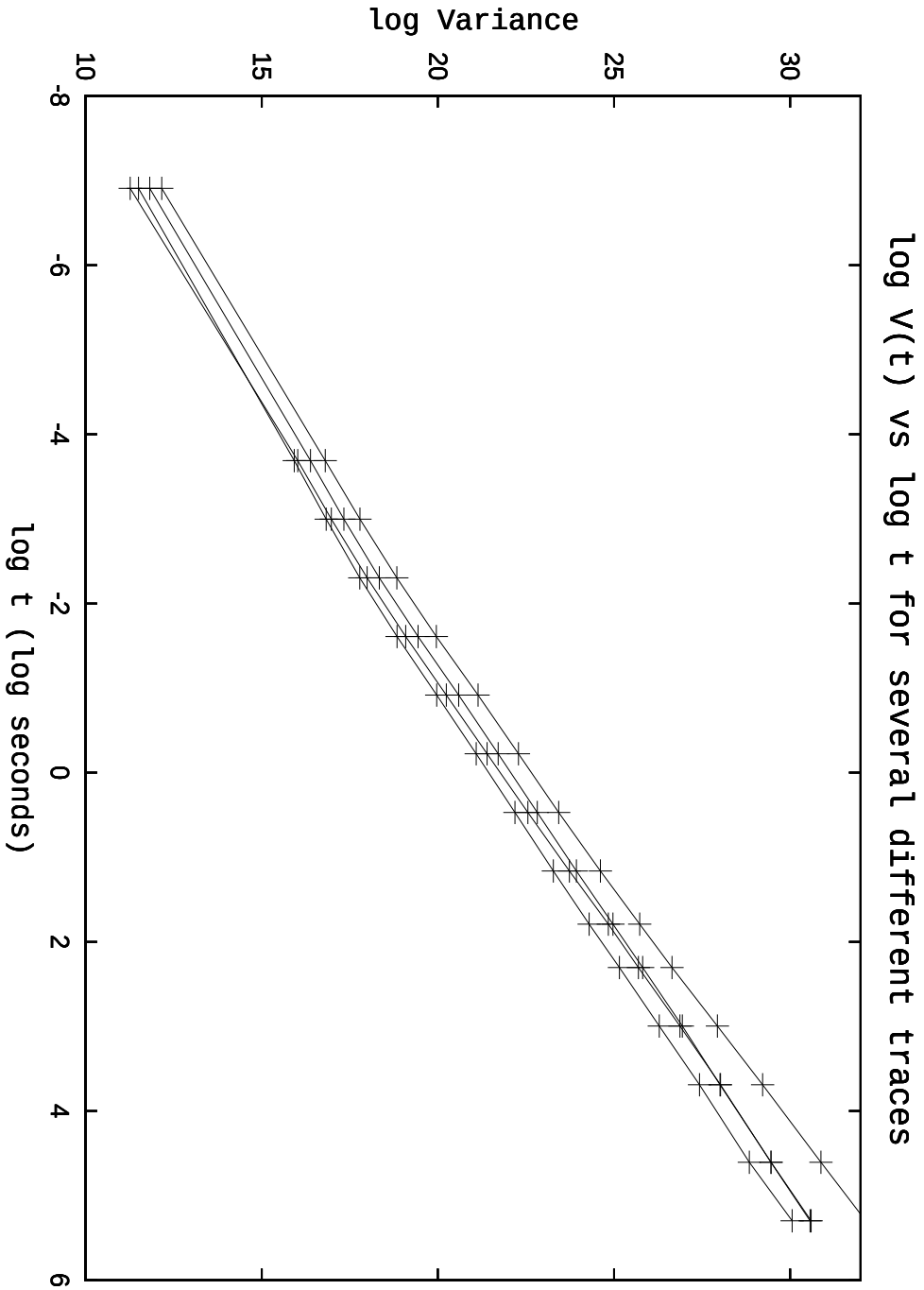


Figure 4.6: Graphical Method for Estimating the Hurst Parameter

Figure 4.7: Graphical Method for Estimating the Hurst Parameter: Variance of Aggregated Series



Exercise 4.4. Use `tcpdump` to observe network traffic

`Tcpdump` generates large quantities of data very quickly, so it would be wise to develop scripts which process this data on-the-fly. Then only the relevant data (packet arrival times, or packet counts in successive sampling intervals) will need to be stored.

Once you have this data, estimate the variance-time curve of the packet arrival count process from the data and plot this curve with logarithmic x and y axes. □

4.3.2 Reliability and Security Measurements

Both reliability and security share the feature that the parameter of the network or system under consideration must be observed over a long period of time and the state of the network can only be evaluated by measuring the frequency, duration, and *severity* of a series of relatively rare events.

In the case of reliability, the events we are concerned with are failures of equipment, and the occurrence of these failure events is a sign of a lack of reliability, whereas the events of concern in the case of security are failures of security policies and procedures (possibly prompted by the discovery of a previously unrevealed weakness by an attacker), and these events display a lack of *security*.

The *severity* of a failure event can be measured as the quantity of traffic (measured in bits/s, or as a proportion of all network traffic) which is disrupted and the duration can clearly be measured in hours, minutes, or seconds, or as a proportion.

For example, if only one event occurred in a year, this event lasted for 2 hours, and caused 10% of traffic to be affected, we would summarise this particular event as having intensity 10% and duration 0.02%.

As has already been discussed, we typically set standards for reliability which constrain the proportion of time during which a service is allowed to be *unavailable*. Events of low intensity are probably not of great concern, unless they occur quite often. Alternatively, we could form an aggregate measure of reliability by multiplying intensity by duration for all events and adding up the resulting terms, to produce an overall figure of *performability* (as this type of measure is called).

The main way in which reliability *and* security can be measured is simply by rigorously recording all significant events where reliability or security is compromised. Such a record forms a *log* of reliability and security events. At regular intervals, typically about once a year, this log should be reviewed (audited) to determine if performance targets are being met.

In the case of security breach events, it is possible that costs of a different nature may be incurred. Disruption of traffic quite often does occur, but, in addition, it is possible that users or organisations will lose data, data could be *altered* (without discovery for some period of time) or information of a sensitive nature might be revealed to parties to whom this information was not intended to be available. The *intensity* of such events can only be estimated in a fairly qualitative manner.

Nevertheless, it is useful to attempt to measure the intensity of security failures and the risk of such events because without making such estimates, there is a potential for an error in which the mechanisms selected to *avoid* certain security risks actually cause more disruption to users than the problems they are supposedly protecting against.

Example 4.3. Evaluation of a Security Log

A log of significant reliability and security problems in a middle size organisation has been kept for the month of July and is depicted in Figure 4.8.

How can we measure the impact of the events reported in this log? One event appears to be completely incomparable to another. However, the impact of each event can, in the last event, be traced back to the effect it has on individual users connected to a network. □

Exercise 4.5. Comparison of Security Logs

Consider the logs set out in Figures 4.8 and 4.9. Estimate the severity of each event and present a judgement as to which month experienced the better performance.

□

```
Reliability and Security Breach Log
Date    Time  Description                    Duration   Disruption
Jul 1   0931  Failure of internet gateway    10 min     Tot fail ext traff.
Jul 3   0032  Planned outage mail server     1 hour     Lost Access to mail
Jul 4   1236  Email Virus Infection          2 days     Heavy mail load *(1)
Jul 10  1530  Router configuration error     30 min     Building 1 access lost
Jul 12  1035  Internet Access Link down      35 min     Tot fail ext traff
Jul 12  1144  Email Virus Infection          8 hours    Heavy mail load *(2)
Jul 15  0235  Unix server break-in           4 hours    Server off-line *(3)
Jul 18  1945  Intranet access failure        6 hours    Intranet services fail
Jul 22  0630  Failure internet gateway       10 min     Tot fail ext traff.
Jul 26  1236  Email Virus Infection          4 days     Heavy mail load *(4)
Notes:
(1) Estimated total number of bogus messages sent averaged 10 per mail
client [organization has staff of 100 persons, each representing one
mail client]
(2) Variation on the previous virus. Average 5 mail messages per client.
(3) Server placed off-line for 4 hours during rebuilding of operating system
(4) New virus. Average 20 mail messages per client.
```

Figure 4.8: A security log

```
Reliability and Security Breach Log
Date    Time  Description                    Duration   Disruption
Aug 1   0930  Failure of internet gateway    10 min     Tot fail ext traff.
Aug 3   0032  Planned outage mail server     2 hour     Lost Access to mail
Aug 4   1236  Email Virus Infection          4 days     Heavy mail load *(1)
Aug 10  1530  Router configuration error     15 min     Building 1 access lost
Aug 12  1035  Internet Access Link down      15 min     Tot fail ext traff
Aug 12  1144  Email Virus Infection          16 hours   Heavy mail load *(2)
Aug 15  0235  Unix server break-in           1 day      Server off-line *(3)
Aug 18  1945  Intranet access failure        2 hours    Intranet services fail
Aug 22  0630  Failure internet gateway       5 min      Tot fail ext traff.
Aug 26  1236  Email Virus Infection          8 days     Heavy mail load *(4)
Notes:
(1) Estimated total number of bogus messages sent averaged 20 per mail
client [organization has staff of 100 persons, each representing one
mail client]
(2) Variation on the previous virus. Average 10 mail messages per client.
(3) Server placed off-line for 24 hours during rebuilding of operating system
(4) New virus. Average 40 mail messages per client.
```

Figure 4.9: Another Security Log

References

[1] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, 2:1–15, February 1994.

[2] V. Paxson and S. Floyd. Wide-area traffic: The failure of poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, June 1995.

[3] P. Abry and D. Veitch. Wavelet analysis of long range dependent traffic. *Trans. Info. Theory*, 44:2–15, 1998.

[4] Robert J. Adler, Raisa E. Feldman, and Murad Taqqu. *A Practical Guide to Heavy Tails*. Birkhäuser, 1998.

[5] Peter Danzig, Jeff Mogul, Vern Paxson, and Mike Schwartz. The internet traffic archive. Internet Web Site. http://ita.ee.lbl.gov/html/traces.html.

Chapter 5

Routing and Control

In this chapter we shall study routing in networks, specific routing schemes in TCP/IP, X.25, and telephone networks. Particular attention is paid to shortest-path algorithms because of the widespread use of shortest-path routing in the Internet. Finally, we shall study the role of Connection Admission Control, and its interrelationship with routing.

In order for any communication to traverse a network, it is necessary to find the appropriate path through the network for this communication to follow. This process of finding the appropriate path through a network is known as *routing*.

The simplest approach to use to routing is to allocate a cost to each path, probably as the sum of a cost assigned to each link, and then always (as far as possible) to choose the path with the least cost. This is the approach usually used, for example, in the Internet. This is considered in detail in Section 5.1.

However, this approach does have its limitations, and is *not* used in telephone networks, and in general is not used in networks with *Connection Admission Control* (CAC). It is not difficult to see some obvious problems associated with shortest path routing which can be addressed by the routing algorithms used in networks with CAC, as discussed in Section 5.2.

In recent times the Internet has begun to adopt, or consider adopting, some routing ideas with their historical roots in the world of telephony. Notably, connection admission control and the use of explicit routes have begun to appear at least in the draft standards of the Internet. These developments are discussed in Section 5.3.

Yet another type of routing, termed *layered routing* in this book, and known best under the name Multi Protocol Label Switching (MPLS), has also evolved within the last few years within the Internet. This approach has developed in some respects as a way to combine the strengths of TCP/IP networking and Asynchronous Transfer Mode (ATM).

Finally, in Section 5.5, some of the seven standard examples are reconsidered in the light of the knowledge provided in this chapter about routing.

5.1 Routing and control in the Internet

We shall start with a discussion of routing. Control will be first addressed in Subsection 5.1.10.

Routing in the Internet, and TCP/IP networks in general, should be viewed at two levels. The base level is *route selection*, which happens according to the current routing tables; and the upper level is a process which alters the contents of these routing tables. Not surprisingly, all the significant aspects of routing take place inside and in the communication between *routers*. Routers contain, at all times, a table of routes, the *routing table*, which is used to make routing decisions. An example of such a table is depicted in Table 5.1.

Such tables can be very long, in important routers. The appropriate entry to use is the one for which the IP address of the routed packet matches the *Destination IP* entry in the table in the bit positions where the mask is non-zero, and for which the number of non-zero bits in the mask is greatest. If there are two entries which both match, the *more specific* of the two entries is chosen, i.e. the one with the mask with greater number of 1s. Normally there will be only one entry which matches with the maximum number of matching bits in the mask

Destination IP	Mask	Interface	HW Address	Cost
139.86.139.96	255.255.255.224	eth0	34:a6:70:9c:b2:30	1
139.86.139.0	255.255.255.0	eth0	1a:56:7b:8c:b2:30	1
0.0.0.0	0.0.0.0	eth0	a4:36:7b:8c:b2:44	1

Table 5.1: A routing table

although it is also possible (when OSPF is the router protocol in use) to set up multiple routes with exactly the same cost which share the traffic to certain destinations.

Load sharing over routes with the same cost does not appear to be a widespread phenomenon in the Internet at present, except in situations where the need for load sharing is blatantly obvious, such as several servers sharing the load of providing a certain service with very high demand.

This routing algorithm is, in itself, deterministic. That is to say, it will choose the same path for a given packet *every time*, except perhaps in cases where there is a degree of ambiguity – for example, where two entries match an incoming IP address equally well. However, even though routing in accordance with the routing table is essentially *static* the entries in the routing table are up-dated *dynamically*. The entries in a routing table may change over time, without operator intervention. This is what makes routing in TCP/IP networks dynamic and allows TCP/IP networks to adapt to changing situations.

The procedure used to update routing tables is that routers communicate information about distances (we could call these costs, although that might be misleading) between themselves and other routers. In this way, all routers obtain accurate information concerning the distance between themselves and all other routers. They then select the *shortest path* between themselves and any other router (or destination network) as the path to use in the routing table.

5.1.1 Finding the shortest path

Finding shortest paths through networks is one of the easier analysis tasks, in a network. One of the best known algorithms for this purpose is *Dijkstra's algorithm*, which calculates the shortest path to every node in the network from a certain fixed starting or finishing point. We assume that the network is made up of nodes and links, and that the links are assigned non-negative costs, as in Figure 5.2.

Dijkstra's algorithm is a *labelling algorithm*. At each step, the state of the algorithm is captured by the set of labels attached to the nodes of the network. These labels refer to the distance from the starting node (or to the finishing node). Each label is also nominated as *permanent* or *temporary*. Permanent labels, as the name suggests, stay the same till the end of the algorithm. The algorithm is described in Figure 5.1.

Since one node has its label declared permanent at each stage, the algorithm will relabel all nodes after n stages, where n is the number of nodes in the network.

The reason this algorithm works is as follows:

At the first stage, when only the target node has a permanent label, the collection, $\mathcal{P}$, of permanent labels has the following properties:

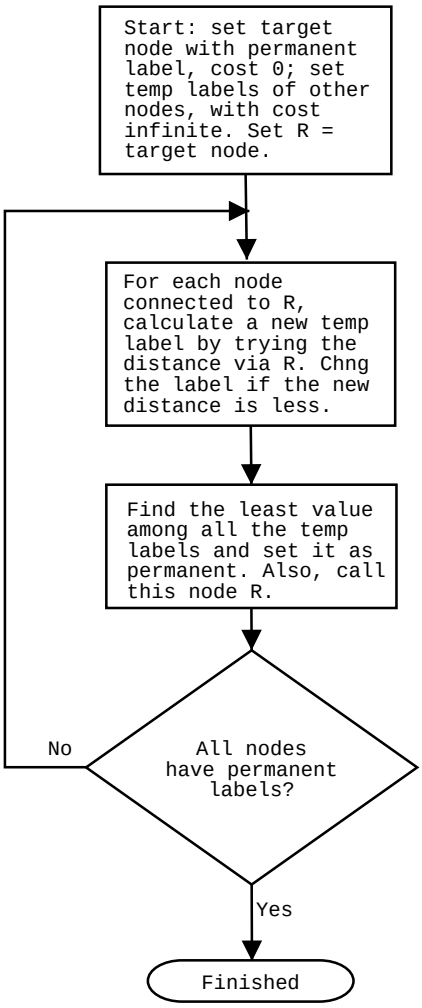
(i) these labels provide the least cost of a path from the labelled node to the target node, and,

(ii) the nodes with permanent labels are also *at least as close* to the target node as all the nodes with temporary labels.

Suppose that all this is true at a certain stage. Let us now show that these properties of $\mathcal{P}$ remain true from step to step through the algorithm. Let us recompute the temporary labels for nodes directly connected to nodes in $\mathcal{P}$, as in the algorithm, and consider the node, B, with the cheapest temporary label. The cheapest path to B must pass directly from a node in P to B because any other path will have to cost at least this much. So this node can be added to $\mathcal{P}$, preserving the asserted properties of this set of nodes. In this way, $\mathcal{P}$ can be expanded, node by node, to include the whole network.

Example 5.1. Finding the shortest paths

Figure 5.1: Dijkstra's Algorithm



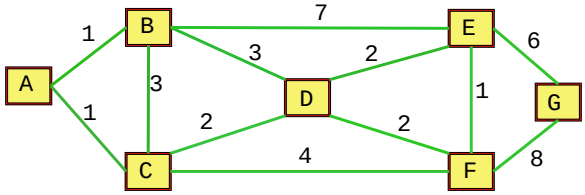
Consider the network depicted in Figure 5.2. Let us use Dijkstra's algorithm to work out the shortest paths through this network.

In applying this algorithm, we assign labels to the nodes of the network stage by stage, starting with a permanent label of zero at node A.

Each label is considered temporary at first and only declared permanent when this particular label is found to be the smallest of the temporary labels which are currently not permanent. When a label is declared to be permanent, all the nodes with temporary labels connected to this node are checked to see if a lower temporary label may be assigned by taking into account the path from this node with a newly permanent label. If a lower label is produced in this way, the temporary label is changed to permanent.

The labellings of this network at each stage, from first to last, are shown in Table 5.3. The labels are only shown in the table at the stage when they change.

Figure 5.2: A simple network



Stage	Node	Label	Permanent?
0	A	0	Y
1	A	0	Y
1	B	1	N
1	C	1	N
2	B	1	Y
2	D	4	N
2	E	8	N
3	C	1	Y
3	D	3	N
3	F	5	N
4	D	3	Y
4	E	5	N
4	F	5	N
5	E	5	Y
5	G	11	N
6	F	5	Y
7	G	11	Y

Figure 5.3: Calculations of Labels, Stage by Stage, for Example 5.1

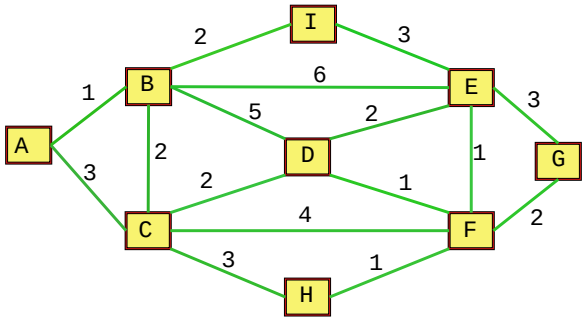
□

Exercise 5.1. Shortest Paths though a Simple Network

Consider the network depicted in Figure 5.4. Use Dijkstra's algorithm to find all the shortest paths to node G, and their cost, in this network. Make sure to be explicit about *how* the algorithm is used to find the solution. A table in the same format as Table 5.3 might be the best way to explain how your calculations progressed.

Make sure to calculate paths *to node G*, not from node A. In this respect this exercise is different from Example 5.1, however the difference is not fundamental. □

Figure 5.4: A simple network



Exercise 5.2. Routing Tables for a Simple Network

Calculate the shortest paths to a second node, node D, in the network of Figure 5.4.

Having worked out the appropriate shortest paths through the network of Figure 5.4 to nodes D and G, determine routing tables for all nodes in the network with a view to ensuring that packets to nodes D or G will always follow the shortest path. Each routing table should contain a list of *entries*. Each entry refers to a single destination and provides the following information: *what node should be next on the path to G*. An example routing table (for the node A) is depicted in Figure 5.5. This is not necessarily the correct routing table.

Figure 5.5: A Routing Table for Node A in Figure 5.4

Destination	Next Hop
D	B
G	C

How would you go about calculating entries for these routing tables for *all* nodes in the network?
In a network of n nodes, how many routing table entries are required in each routing table? □

5.1.2 Subnetworks

Although the principle of shortest path routing is firmly established in the Internet, in order to make shortest path routing workable even in very large networks (such as the Internet as a whole), several techniques for breaking the routing function down into a hierarchy of different stages of routing have to be used. At the bottom of this hierarchy we have the techniqe of *subnetting*.

Except at the final stage or two of routing, routers choose the next hop for a packet on the basis of only *part* of the address of the packet – the network part. The way we divide the IP address of a packet into a network part and a host part varies depending on *where* we are in the Internet. This is the key concept underlying subnetting.

Example 5.2. Subnetworks

Suppose my host is in the subnetwork used by the Department of Measurements and Routing in the University of Certain Routing. The University as a whole has, say, all the IP addresses in the Internet which start with 133.48 (in the first 16 bits of the IP address). The Department on the other hand makes use of all the IP addresses which start with 133.48.113 (in the first 24 bits of the IP adddress).

The advantage of this framework is that routers outside the university do not need to distinguish between packets going to different addresses at the University. Any packet going to the university will be routed in the same way. When the packet arrives at the university, this changes. Inside the university, the packet will be routed according to its 24 bit subnetwork address, instead of in accordance with its University network address.

This trick can be re-used multiple times – we can have subnetworks within subnetworks to any level. And there are no constraints on the number of bits which are used to identify a subnetwork. Nor are there are any

requirements that the subnetwork sizes are consistent in different places in the same organisation. A university can, if it wishes, use subnetworks with network addresses of length 24 bits in one place and of length 26 bits in other places. This doesn't seem to happen very often though – probably because it could be confusing.

The defining principle, which defines how routers work and how subnetworks are defined, is that all routers make use of the entry in their routing table which provides the *longest match*. The address in a routing table must exactly match the packet for each bit position where the network mask in this routing table entry is set. If the length of the network mask is longer for entry A (i.e. has more bits set) than for entry B, say, then entry.A will be chosen ahead of entry B. □

Subnetworks are only the bottom level of a hierarchical subdivision of the routing problem. However, at the levels above the network - subnetwork layer, the reason for the hierarchical subdivision of routing is a little different. Instead of attempting to simplify and reduce the number of entries in routing tables, our objective is to reduce the amount of *traffic between routers*.

(There are also other techniques for simplifying and subdividing the routing problem such as *layered routing* – this is described in Subsections 5.4.1 and 7.3.5, and *Network Address Translation (NAT)* – see Subsection 5.1.5.)

5.1.3 Routing Domains in the Internet

It is very important to know that routing in the Internet is completely independent from naming of nodes and domains. Names are handled by *domain name servers*. Routing is handled by routers, and it is not on the basis of the *name* of the destination, but on the basis of its *number* that a packet to a certain destination is routed.

However, because the Internet is too large to allow the protocols by means of which routers communicate routing information to keep every router in the entire Internet informed about the routes from one side of the Internet to the other, for the purpose of routing, the Internet *is also* divided up into *routing domains*. Actually, the terminology we should use is a little more complex than this. We need to define the terms *Autonomous System*, *Routing Domain*, and *Administrative Domain*.

These concepts have evolved with the development of the Internet and are still evolving, however, the definition of these terms at present is as follows: An *administrative domain* is a collection of hosts, routers, and any other networking equipment which is under the administration of a single body or organisation. This concept can become somewhat unclear if a collection of bodies choose to administer their networks jointly, by appointing a committee to coordinate their administration.

The concept of *routing domain* is not specific to the Internet. A routing domain is any collection of hosts, routers, and any other networking equipment within which a consistent routing policy is used. The following definition of *routing domain* was also given in [1] and quoted in [2]:

> "A Routing Domain is a collection of routers which coordinate their routing knowledge using a single (instance of) a routing protocol."

An *Autonomous System* is a routing domain under a single (or at least unified) administration that has been assigned an *Autonomous System Number* by the Internet Engineering Task Force (IETF). Currently there are approximately 11,000 Autonomous System Numbers in use worldwide. A definition of *autonomous system* has also been given in [3], which was repeated in [2]:

> "The use of the term Autonomous System here stresses the fact that, even when multiple IGPs and metrics are used, the administration of an AS appears to other ASs to have a single coherent interior routing plan and presents a consistent picture of what networks are reachable through it. From the standpoint of exterior routing, an AS can be viewed as monolithic: reachability to networks directly connected to the AS must be equivalent from all border gateways of the AS."

In principle, in the Internet, it is conceivable that a routing domain could be further sub-divided into other routing domains, and those could be further sub-divided, and so on. In practise, this is not necessary at the moment.

5.1.4 Router Protocols

The other element in the architecture of routing in the Internet that we need to consider is the protocols that are used to interchange information between routers. At the outer edge of the Internet, some networks do not need router communication protocols at all, because the rules which ensure shortest path routing are sufficiently simple that the routers can be managed manually.

When the number of hosts and sub-networks in an organisation rises above a certain level, however, it becomes necessary to use router protocols which allow routers to create their own routing tables dynamically. The router communication protocols all have the function of finding out the shortest paths from this router to other hosts in the same routing domain, or the best choice of a gateway router when the packet being routed is destined for a host outside the routing domain.

Different router communication protocols are used *within* autonomous systems from those that are used *between* autonomous systems. In addition, there are a variety of different router communication protocols designed for use within an autonomous system, and, likewise, a variety of router communication protocols designed for use between gateways of autonomous systems to establish the shortest paths for packets to follow from one autonomous system to another.

The notable router communication protocols in the Internet are RIP [4] and OSPF [5] for use within autonomous systems and BGP [6] for use between autonomous systems.

Although the basic goal of router communication protocols is quite clear – to provide to each router sufficient information in order that it can identify the best hop, for shortest path routing, for every packet it has to handle – there are significant alternatives for the way this information can be phrased and transferred from router to router. Since the volumes of information generated and consumed by routers is quite significant, it is important to design these protocols so that they are reasonably efficient and yet are able to maintain the information in routing tables as accurately and as up-to-date as possible. A compromise is clearly necessary because if the routers communicate with each very often, the load of all this routing information will be a burden on the network whereas if they communicate too seldom, the information in routing tables will be at significant risk of being out of date.

One way to understand the way router communication protocols work is that they enable the network of routers as a whole to implement a distributed version of Dijkstra's algorithm. (It would be more accurate to call this distibuted algorithm the Bellman-Ford algorithm, an algorithm which is not as efficient as Dijkstra's algorithm but fits the distributed implementation more naturally).

Whereas Dijkstra's algorithm proceeds in stages, the routers of a TCP/IP network repeat the algorithm for selecting shortest routes over and over again, so that if the shortest paths have been identified, or nearly identified, and a change occurs in the distance between certain routers, e.g. a link fails or a link becomes available, the routers will be able to discover that their routes should be changed, and make the changes, in a reasonable period of time.

In fact, sometimes Dijkstra's algorithm is used explicitly within the routers. The routers are able to communicate with each other to ensure that all routers have complete accurate information concerning connectivity and costs of links throughout the network and then an application of Dijkstra's algorithm is used to determine the shortest paths. This approach is fast and quite straightforward.

The information that is passed between routers sometimes includes the *shortest distance to any destination* from the router sending router information. This is known as the distance vector. This is the approach used in the oldest of the well known router communication protocols, RIP [4]. The other alternative is for the router communication protocol to send the *distance to neighbouring routers*. The latter approach is preferable in large networks because although there is a little more work for the receiving router to do to work out the shortest paths (at worst an application of Dijkstra's algorithm), and thereby decide what routes to put into its own routing table, the amount of data which has to be sent from one router to another is reduced. In large networks the reducing the quantity of information which has to be transferred is the priority because the traffic generated by routers could become, otherwise, a burdensome load. An even more important advantage of this approach is that information about changes in network state can be taken into account in routers all over a routing domain as soon as information about the change has been received, which will be shortly after the change occured, rather than after waiting for this information to propogate through a series or routers. This is the approach used in OSPF [5].

In the case when the router protocol is operating *within* an autonomous system, this path information is only provided for routers in the same autonomous system. In the other case, communication between routers acting as gateways *between* autonomous systems, the information passed from one router to another says which other

gateways to autonomous systems this gateway is connected to and how far away these gateways are.

5.1.5 Network Address Translation

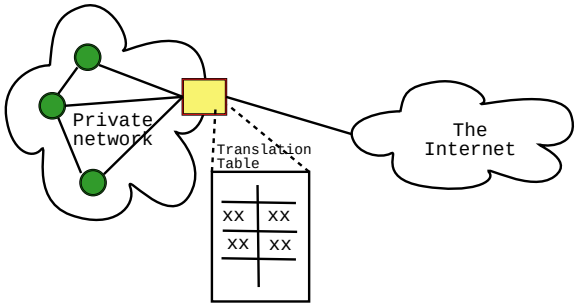
Another routing technique which is very often used in the leaf networks (networks not used for transit to anywhere else by any *other* network) of the Internet, is *Network Address Translation (NAT)* (also known as IP masquerading). This technique causes packets to have their destination IP addresses altered at the gateway between this network and the Internet at large. On the leaf network side of this gateway, IP addresses in the private ranges (10.0.0.0 to 10.255.255.255, 192.168.0.0 to 192.168.255.255 and 172.16.0.0 to 172.31.255.255) are used. On the Internet side, "real IP addresses" are used.

Network address translation violates one of the philosophical principles of IP routing, namely the idea that each packet is independently routed and that routing information should not be stored in the network. However, network address translation is a practical, sensible exploitation of an opportunity so if it seems to be at variance with the original philosophy of TCP/IP routing, so what? The fact that routing information is not *normally* stored in routers doesn't prevent us from doing precisely this in certain situations where this approach to routing has so many advantages that they can't be ignored.

How Network Address Translation Works

Network Address Translation (NAT), also known as IP masquerading, works by storing a table of translations between the private and public addresses in the router, as depicted in Figure 5.6. Since we only use NAT for hosts which are not providing services, all interractions are initiated by the hosts on the private side of the gateway.

Figure 5.6: Network Address Translation



When a host initiates a communication with any other host or server on the private side of the gateway which uses a private IP address, the gateway plays no role and communication between the two hosts takes place in a completely conventional manner. When a host initiates communication with any host which uses a public IP address, whether it is on the private side or the public side of the gateway, the router places an entry in its translation table for this interraction, changes the IP address of the source in the packet to its own IP address, and changes the *source port* entry in the packet to a dynamically selected port number, which is noted in the translation table. If the packet which initiated the interraction was a SYN packet of a TCP connection, the entry will remain active for the duration of the connection (or until a timer expires due to inactivity). If the packet was a UDP packet, the entry remains active until a reply is recieved from the server or host with a public IP address to which the original packet was sent.

It is the dynamically allocated port which helps the router to identify which interraction is relevant to packets returning from the server.

When a packet is translated on the way from the private network to the public network, its private IP address is replaced by the IP address (or *an* IP address, in case there is a choice) of the router, and the port is replaced by the dynamically assigned port. The original private IP address and the original port of the packet are both stored in the translation table so that packets returning through the gateway can have the original values for these settings restored to their original values.

These ports can be allocated and de-allocated dynamically. Although ports are addressed by a 16 bit address field, which means that there are only 65536 of them, this should be sufficiently many for quite a large network. With this collection of dynamically allocated addresses, a router should be able to manage 60000 simultaneous interractions passing through the network address translation mechanism. All be done with just *one* real IP address.

Why it is attractive

A simple explanation of why it is attractive to use network address translation is that this technique enables us to better exploit the limited IP address range. Within the IP version 4 framework, IP addresses have to be less than 32 bits in length. This means we have only 4 billion distinct IP addresses. Network address translation allows hosts to be connected to the Internet without having their own IP address – they use an address in the private IP address range. These addresses can be reused an unlimited number of times.

In fact, conventional IP addresses are only really necessary for hosts which will act as servers. At present there are many situations where network address translation could be used to recover many IP addresses. The ultimate limit of this process would be to reserve conventional IP addresses only for servers, and use only private IP addresses for hosts which do not act as servers. In this way we could make do with the IPv4 address range for a lot longer. Whether this effort is warranted is unclear. However, this technique, and others, should have no trouble spinning out the limited IPv4 address space for quite some time to come, well and truly long engough to enable the Internet, its hosts and routers, to be ready for IPv6, which has 128 bit host addresses.

Network address translation has some other advantages, aside from saving IP address space. Another advantage is that because the IP addresses used on the private side of the gateway are private, the *only* way a packet can reach one of the private hosts is by passing through the network address translation mechanism. This means, in particular, that hosts on the private side of the network can only participate in interractions with hosts on the public side if they are initiated from the private side. Thus, the router where the network address translation is implemented operates as a basic firewall. A proper firewall will probably implement other security measures as well, but this one mechanism goes a long way to ensuring the integrity of a private network.

5.1.6 Router Configuration

It is important to realise that all the automatic routing configuration carried out by routers relies on the specific information provided by individual network managers when they configure routers. If this information is incorrect, routers will make use of, and propogate to other routers, this incorrect information. A significant proportion of routers in the Internet *do* contain incorrect or inappropriate data. It is not all that easy to get it right! A study of routing errors and end-to-end routing in general in the Internet is presented in [7].

The basic information which routers need to be equipped with accurately is as follows:

(i) routes to all the other routers to which this router is directly connected, including the cost / distance / hop-count to this router, and, if a router communication protocol is *not* in use:

(ii) routes for any networks that are indirectly connected to this network, and,

(iii) routes for any hosts that are indirectly connected to this network, and, for which tbe best route is not the same as any network route, and

(iv) a default route which will be used for access to hosts and networks not explicitly dealt with already.

In many leaf networks the information just specified will be all that is required, in each router. For larger networks, however, specifying routes can become error prone and it is preferable to specify only item (i), leaving the rest for the router to work out for itself, by communication with other routers and the use of a shortest path algorithm.

However, if the information provided at item (i) is incorrect, the router communication protocols will not discover the error, they will merely propogate this error as widely as possible.

Exercise 5.3. Inspect Some Routing Tables

Use the *route* command which exists on most versions of Unix to determine the routing table in your current computer. If your computer is "Windows only", there is still a satisfactory way to conduct this test. The command in question follows a slightly different terminology, but in other respects it is almost exactly the same. □

5.1.7 Virtual LANs (vLANs)

The vLAN concept

The *traditional* relationship between a layer 2 protocol, such as ethernet, and a layer 3 protocol, such as IP, is that the layer 2 protocol operates between hosts which are directly connected whereas the layer 3 protocol acts across networks of many connected links and enables end-to-end communication.

Background

In addition, the interworking of these protocols is facilitated by the mechanisms of the Address Resolution Protocol (ARP) and the Reverse Address Resolution Protocol (RARP) which make use of broadcasts throughout an ethernet to establish a fixed relationship between the ethernet addresses (Media Access - MAC addresses) and IP addresses. The confinement of this broadcast traffic to a single LAN is important, because, for example, if such traffic was broadcast throughout the Internet, we would have very little in the way of communication resources left for any other purpose.

The original ethernet concept used to broadcast *all* packets over the entire LAN. In fact, originally, there was no hub – the hosts simply broadcast their packets over a shared medium. However, it soon became apparant that a more economical approach was to use the cheaper medium of pairs of twisted pairs configured to connect each host to a central hub. One pair in each pair of pairs is used for communication *to* the hub, the other for communication *from* the hub. This hub rebroadcasts all incoming packets from any incoming pair to all outgoing pairs.

However, in this framework, broadcasting all packets is unnecessary, and when traffic in LANs became sufficiently intense to cause congestion, the idea that the hub could transmit outgoing packets only to the host to which this packet was directed arose naturally. The extra "intelligence" to do this in the hub could be incorporated in the central hub without increasing the cost of manufacture unduly. A device of this sort should no longer be called a hub, but instead a *switch*. Assuming that the switch is capable of transferring packets between more than one pair of ports simultaneously, switching instead of rebroadcasting significantly increases the carrying capacity of a LAN. There is also a significant security advantage in confining the transmission to paths connected to the communicating hosts.

However, as already mentioned, the ARP and RARP protocols *require* broadcasts over the entire LAN for their operation. There are other types of local broadcast traffic which also, generally, are confined to the immediate LAN. Broadcast traffic of this sort has the potential to limit the total capacity of a LAN, even when switches are used instead of hubs, so we may need to consider carefully how it should be managed.

Although this subdivision of labour between Layer 2 and Layer 3 hardware is quite neat and makes reasonable sense, there are also good reasons for relaxing this arrangement and allowing layer 2 connectivity to spread over a somewhat broader region of a network, and freeing up the concept of a broadcast region so that it does not necessarily correspond to a single LAN.

It might not be *ethernet* that we use at layer 2 – it could be ATM for example, or the Token Ring protocol. In any case, the principle remains the same. We want to allow nodes in our network to join a broadcast domain which is suitable without this choice being determined, necessarily, by physical location or connectivity. Also, we would like these broadcast regions to be easily defined, easily adjusted, and to be able to spread over the hosts connected to more than one layer 2 switch.

Since ATM was designed to provide wide-area networking it is hardly surprising that the idea of putting more than one ATM switch together to provide wide-area connectivity underneath an IP layer should arise naturally. However, nowadays the same capabilities can be provided by collections of ethernet switches.

The vLAN Concept

The concept we are seeking to define here, is called a virtual LAN (vLAN). The concept is derived, to a degree, from the earlier *LANE* concept from CISCO **??**. In principle, in a network capable of providing vLANs, the following principles apply:

(i) Hosts may be in any vLAN, or more than one, and may easily be switched from one to anther.

(ii) vLANs may extend across several physical LANs;

(iii) communication *between* vLANs always takes place via a router, at least to the extent that the first packet of a *stream* (e.g. the SYN packet of a TCP connection) must pass via the router, *even when the hosts are on the same physical LAN*.

The vLAN concept serves a number of competing purposes:

(i) hosts on different LANs can communicate via switches without passing through a router, hence reducing communication delay and reducing load on routers;

(ii) hosts on the same LAN can be forced to communicate via a router, thereby enforcing security policies;

(iii) membership of vLANs can be centrally managed;

(iv) broadcast traffic can be confined to vLANs rather than spreading over the entire physical LAN.

Implementation

The vLAN concept can be implemented in many ways and with a variety of layer 2 protocols. The IEEE Standard 802.1Q [8] specifies a broad framework for vLAN operation. This standard specifies how certain existing or additional fields in layer 2 protocols should be used to specify the virtual LAN on which a packet is to be transmitted. In addition, this standard includes other fields - for example, for specifying a *priority* for a packet.

The way in which a packet from a host is assigned to a certain vLAN is not completely determined by the 802.1Q standard. Let us associate vLAN identifiers, VIDs, with each vLAN. So assignment to a vLAN is equivalent to assigning VIDs. The following approaches have been considered, and have been used at various times:

(i) VIDs associated with each port;

(ii) VIDs associated with each MAC address;

(iii) VIDs associated with the IP subnet;

(iv) a mixture of the above.

The choice between these different approaches must be based on security considerations, and manageability. Assignment by MAC address has some great advantages in regard to flexibility and convenience, and has some security advantages, but there is also at the heart a security weakness. For the most part, networking clients have no control over the MAC address assigned to their network interface card. However, the possibility that a sophisticated user *could* change the MAC address of their interface card remains present. This risk cannot be ignored.

Assignment of VIDs by port is probably the most natural approach. Adjacent ports on the same switch can be assigned different vLANs, for reasons of security, for example, while on the other hand ports on different switches can be assigned to the same vLAN, for reasons of efficiency.

Cut-through Switching

A further method for reducing load on routers is *cut-through switching*. This is fundamentally the same idea as used in MPLS (see 5.4.1 and 7.3.5), but in the context of a leaf network rather than the core or transit region of the Internet. When a TCP connection request packet is routed by the router with cut-through switching capability, it sets up a path through the switches of the network served by this router so that subsequent packets of the same TCP connection will not need to be routed – they can just follow the path that has already been set up.

How paths through the switches of a leaf network are set up is the next topic we need to consider. But first, how do the switches know when a packet is supposed to follow one of these paths?

This, surely, depends upon the specific hardware. In the ATM case, the logic is natural and well suited to the ATM architecture. The packets are all segmented and packed into ATM cells at the first point where ATM is used. At this point, a Virtual Circuit Identifier (VCI) and a Virtual Path Identifier (VPI) are also assigned to every cell. The VCI/VPI to be used has to be dynamically assiged. Once assigned to a flow, the same VCI/VPI will be used till the flow is compete. The VCI/VPI of the ATM cells is what is used to make sure that all cells, and therefore also packets, follow the same path.

If the layer 2 hardware is ethernet, on the other hand, it is not immediately clear how subsequent packets to the first of a flow are forced to follow the path which has been set up. This can be done in a variety of ways and is not standardised to any significant degree. Let us simply indicate here, a *possible* method.

Consider a TCP connection which is being set up between to vLANs with VIDs 3 and 5. For simplicity, we shall further assume that both the source and destination hosts are located on physical LANs which are directly connected to this switch. The case where they are connected directly to different switches will be addressed in a moment.

When the SYN packet, i.e. the packet which sets up the connection, is received by the switch-router it will need to be routed in the usual way. In addition to routing this packet, however, the router will place an entry in a table in the hardware of the line processor to which the physical LAN of the source host is connected. This entry has the effect of providing a short-cut for packets with the same socket-pair as the SYN packet of the TCP connection. By a *socket-pair* we mean the following data: incoming IP address, incoming port, outgoing IP address, outgoing port. All the subsequent packets with identical values for these parameters will be part of the same connection, and so there is really no need for these packets to be re-routed. This entry in the line unit must be removed when the FIN packet for the connection is received, or, alternatively, after a period of time during which no further packets in this TCP connection are received.

Signaling – PNNI

If packets are to be switched at layer 2 along a path including more than one switch, some communication between the switches to prepare this path in advance will be necessary. This is a well established procedure in telephone and ATM networks, where it is known as signaling. Signaling of this sort is not necessary in TCP/IP networks because the philosophy of TCP/IP networks deliberately avoids storing routing state information anywhere in the network. As a consequence, all packets in TCP/IP networks have to be routed on the basis of the IP address alone.

However, the stresses of exponential growth in traffic levels can with reason be expected to soften the Internet philosophy. So the concept that storing state information in routers and/or switches is a “bad thing” does need to be reconsidered and revised. A breif reconsideration of this idea reveals that storage of state information in this manner can speed up routing quite considerably. It really amounts to caching of routing decisions. Caching is standard method for speeding up all sorts of computational processes. Why not do this for routing as well?

Once it is admitted that pre-allocation of paths is not necessarily a bad thing, however, we are faced with the need for signaling. Signaling is just the process of communication between switches which stores state information in these switches for the purpose of switching packets of a particular flow along a preallocated paths.

The ATM signaling protocols can be divided into two components: User Network Interface (UNI) signalling, which is the signalling used between terminal devices and the ATM network, and Public Network to Network Interface (PNNI) signalling, which is used between two public ATM networks. The signaling protocol used *inside* a public or private ATM network is a proprietary issue and is not standardised, however it can safely be assumed that it must be similar to PNNI signaling.

The two key functions of the PNNI protocol are as follows:

(i) to communicate sufficient routing information from one ATM node to another that each node has sufficient information to know how to route its new connections;

(ii) to allow ATM nodes to communicate with other ATM nodes in order to set up an ATM connection.

The complexity of signaling is a very significant barrier to the development of signaling protocols for a variety of different switch architectures. A complete signaling architecture and family of protocols has been established for ATM networks. No such protocol exists for interconnected ethernet LANs, for example. Specialised proprietary protocols for communicating information about flows through a network of interconnected ethernet LANs have been developed. However, it seems unlikely that these protocols would be used for purposes outside the small extension of enabling cut-through switching to pass ethernet frames in an existing flow from one switch to another without the need for routing any but the first frame.

Another reason why it is unlikely that a global switched-ethernet signaling protocol would ever be likely to undergo development is that the MPLS concept already provides a scheme for transferring the benefits and concepts of cut-through switching to a global, Internet-wide context, and it does this in a manner which does not require the definition and development of signaling protocol such as PNNI. In the case of MPLS, the objective

is to make use of the best features of connection-oriented, explicit path routing, as in ATM networks, without significantly increasing the complexity of the current protocols used for of

5.1.8 IP Version 6

The main limitation of the current implementation of the TCP/IP protocols is known as IP Version 4. A new version of the TCP/IP family of protocolshas been defined. It is known as IP Version 6 (IPv6).

The main problem with IP Version 4 is the limited address space for hosts. The IP Version 4 header allows for a host address of 32 bits whereas in IP Version 6 hosts have 128 bit addresses. At the time when IP Version 6 was formulated and adopted there was a perception that the limited address space in IP Version 4 could become a problem in the near future, however, since then the address space problem has been addressed in a number of ways and does not appear to be so pressing after all. IP Version 6 can be phased in without the entire Internet needing the upgrade all at once. A graceful transition mechanism for introducing IPv6 has been defined [9].

The fact that the address space problem is no longer so pressing seems to have lead to the introduction of IP Version 6 being deferred for the the time being. In the mean time, IPv6 capability is being introduced into all new routers and new versions of operating systems and new features introduced into the Internet are generally defined to be compatible with both IPv4 and IPv6.

5.1.9 Load Distribution

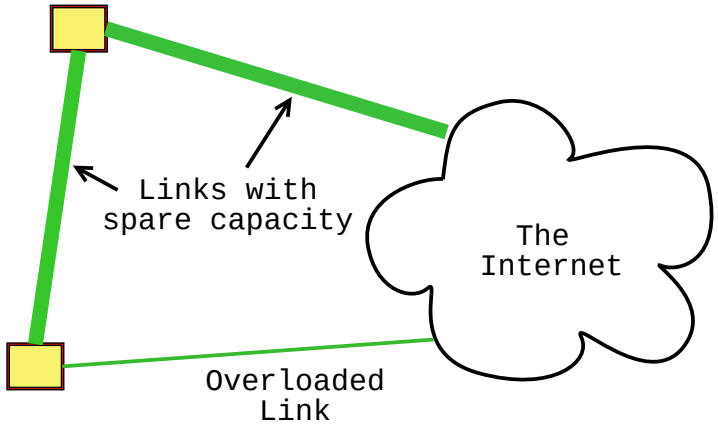
Load sharing of an individual point-to-point traffic flow cannot occur easily under shortest path routing. The OSPF router communication protocol and its associated routing algorithm is able to distribute traffic load over a collection of different paths *if* these paths all have the same cost. However, this does not allow very much in the way of load distribution, which might be needed to be applied over paths with *different* costs to be really useful.

Fortunately, it is quite likely that a reasonably balanced distribution of traffic will occur nevertheless, because of the very great diversity of destinations to which Internet traffic is typically directed. The amount of traffic going to any one destination is likely to be insignificant, or at least a relatively small percentage of the whole of the out-going traffic, in the whole scheme of things, except in unusual situations where one particular destination has special importance.

Also, it can reasonably be argued that if there is a tendency for all traffic to congregate on a certain link, and thereby cause congestion, it could be that what is really required is an *upgrade of that link*, rather than a routing algorithm which imposes greater routing diversity.

Consider the example depicted in Figure 5.7.

Figure 5.7: A network with unbalanced traffic



In this network, almost all traffic between A and the “The Net”, can pass through either of the routes shown, i.e.through B or not through B. The simple interpretation of the topology of this network is that the nominal cost

of the path though B is 1 greater than the cost of going to the same place on the direct link to "The Net". For the moment, let us assume that this is the case.

In a network with Connection Admission Control (CAC), if the traffic between A and "The Net" on the direct path becomes congested, the other path will start carrying all new traffic. In the Internet, however, the path along the direct link will remain the shortest path even when it is overloaded, so rather than traffic being diverted, all the connections on this path will adapt to the congestion and each will carry lower and lower rates of bytes as the congestion increases.

In an extreme situation, we could find that the link to B remains significantly underloaded even though the direct link is overloaded over long periods of time.

5.1.10 End-to-end Congestion Control

The end-to-end congestion control provided by means of the TCP protocol is one of the great successes of the Internet, and one which is perhaps responsible for the remarkable flexibility of the Internet to adequately provide all sorts of service.

However, any good idea has the potential to become a weakness when pushed to far. End-to-end traffic congestion is an essential – life-saving – feature of the network protocols in the situation which has predominated in the Internet up to now, where there is widespread congestion.

But perhaps a time may come when congestion is not widespread – when most connections will be able to communicate at speeds which suit the computers each end of the path. Certainly, advances in the traffic capacity of an individual fiber and the capacity of routers would suggest that this might come to pass. If this happens, the end-to-end flow control of TCP will become largely irrelevant.

5.2 Routing in Telephone and ATM Networks

Routing in telephone networks is fundamentally different from routing in the Internet. In telephone networks, also, a different approach is used to meeting performance standards. The approach adopted is to *guarantee* the bandwidth that has been requested for the duration of a call. For this reason, if it is not possible to guarantee that this bandwidth will be available, the call will not be allowed to proceed. This process of allowing, or not allowing, a call to proceed is known as *call admission control*, or *connection admission control* (if we wish to generalise the concept to other types of networks).

5.2.1 Connection Admission Control

Connection Admission Control allows a network to *reject* connection requests at the time when they are made. The Internet currently emphasises a "best-effort" service model according to which all requests are granted, but the quality of the service provided is not guaranteed in any way.

This type of service does not use any Connection Admission Control procedure. Despite the lack of control over demand that this implies the Internet has proved remarkably successful and up to this point the vast majority (virtually all) of the traffic on the Internet is of this sort.

However, the RSVP protocol [10], which is an Internet standard and is being implemented in places in the Internet, and the proposed DiffServ architecture and protocols, which are a more recent development [11, 12], do provide scope for the Internet to reject requests for certain *other* services.

The purpose of CAC is to establish that the network from which a service is being requested can provide sufficient resources to be able to provide the required grade of service (performance) for a proposed connection.

5.2.2 Routing and CAC

In telephone networks, X.25 networks, and ATM networks, routing takes place *as part of the connection admission control procedure*. This means that:

(i) if no satisfactory path can be found, the connection attempt can be rejected, and

(ii) if the first path considered is *overloaded* (with traffic), a second, third, etc, path can be considered.

This last phenomenon is known as *overflow*, or *alternate routing*. In the Internet, this approach is not taken because traffic levels are not considered in routing and the option of rejecting a connection is not considered.

Thus, as well as providing a good way to protect the network against overload and thereby allow a network to guarantee service to the client, the CAC mechanism allows us to divert requests along paths where traffic is lower, and thus dynamically redistribute traffic in response to moment to moment fluctuations in load on certain links or in certain parts of the network.

5.2.3 State-based Routing

Once a call / connection has been accepted in a Telephone network or an ATM network, the switches along the path have to be notified that it is established and they then proceed to store entries in their switching tables so that future packets, or bits (for the circuit-switched case), will know what path to follow. This reliance upon state information stored at intermediate points is sometimes regarded as a potential weakness from the point of view of the Internet philosophy, however it seems to work quite well. It is quite analogous to the process which happens in a router which implements NAT or in a switch-router in the case, discussed in Section 5.1.7, where a path is chosen in advance for a flow of IP packets with more than a transitory existence.

Once the connection is established, packets following the same path require an absolute minimum of processing. Checking the entry in a switches route table and diverting it to the appropriate outgoing port can be a very fast process in an ATM switch.

5.3 New Approaches to Congestion Control in the Internet

5.3.1 Congestion and its Avoidance

Let us begin by a quote from [13]:

> A complete congestion management strategy should include several congestion controls and avoidance schemes that work at different levels of protocols and can handle congestion of varying duration. In general, the longer the duration, the higher the protocol layer at which control should be exercised. Any one layer, such as datalink (backpressure) or routing (queueing/service strategies), cannot handle all congestion problems.

The number of congestion control and avoidance strategies in common use or proposed for use is considerable. Since some of the proposed methods might never come into use to a significant degree, let us concentrate on methods which are currently in use.

Definition 5.1 Congestion *is the name we give to any situation in a network where network resources (bandwidth, buffers, or processing capacity) are not adequate to fully cater for offered traffic.* Congestion Control *is the collection of strategies which respond to and manage the impact of congestion.* Congestion Avoidance *is the collection of strategies which respond to congestion in order to reduce traffic load.*

The primary congestion control *and* avoidance strategy in the Internet is the flow control mechanism of TCP. Unfortunately, there is also quite a bit of traffic on the Internet which uses UDP instead of TCP. However, let us start by considering the TCP flow control mechanism.

When congestion occurs, whether it is because of a shortage of bandwidth, buffers or processing capacity (in a router, for example), it will first show up in the fact that certain buffers will fill up and start overflowing. At this stage already the equipment (probably a router) will have to make a choice as to which packets should be discarded. We shall discuss this choice in more detail in the next subsection.

For the moment, let us just consider what happens when the packets are discarded. For simplicity, let us suppose that the congestion has been caused by shortage of capacity on a certain link.

5.3.2 Random Early Discard

An important embellishment of router behaviour based on the TCP end-to-end flow control algorithm was introduced in [14]. TCP flow control principles dictate that the participants in a TCP connection respond to three or more lost packets within a short period by reducing the rate at which they transmit. Given that the hosts at each end of a TCP connection will behave in accordance with the TCP congestion avoidance algorithm, it becomes possible to send a message with a predictable response to either end of the connection by dropping packets.

As the name suggests, in the RED technique, packets in the queue at a rouer are discarded when the buffer size rises above a cerain threshhold. The proportion of packets discarded in the RED scheme increases as the buffer level in the router increases. Below a lower threshhold, no packets are discarded, above an upper threshhold, all packets are discarded, in between, the proportion of packets discarded gradually increases from zero to one.

The RED scheme has the following goals:

(i) to control buffer levels in routers gracefully;

(ii) to communicate to sources in proportion to their proportion of the load on the link and without an unfair bias against bursty sources;

(iii) to avoid a synchronization of load peaks which can potentially be introduced by congestion avoidance mechanisms which respond more suddenly when congestion occurs.

The RED scheme proposed in [14] has achieved widespread acceptance within the Internet. More complex shemes based on the same concepts but distinguishing between different traffic classes will be discussed in the next Subsection.

5.3.3 DiffServ

As previously discussed in Section 3.5.5, there have always been, and probably always will be, calls for network providers to provide different standards of service for different customers (while charging different amounts, of course). Whether this will ever become widespread and popular remains to be seen. However, the draft standards for these protocols are at a late stage of development and they are well supported by equipment vendors, so widespead deployment of the facilities to support differentiated services in the manner descibed in, for example [11] or [12], is a distinct possibility.

It should be understood at the outset that these approaches to control and admission are not intended to *replace* the current approach, of allowing virtually unlimited numbers of requests for best-effort service to be offered to Internet gateways. However, it is not unreaonable to suppose that there might be Internet users who are willing to pay extra for a better service, and perhaps to pay even more for a *premium* service.

The intention of the DiffServ architecture is to provide differentiated service across the Internet without having to pay specific attention to individual traffic streams, except at their entry or exit to or from the network. Because most routers can therefore treat the traffic as an aggregate, and don't need to treat each individual stream separately, it is felt that the implementation of the DiffServ architecture should be readily achievable.

The TOS bits which have always existed in the IPv4 header, but have remained largely ignored, are used in the DiffServ architecture to determine the priority with which packets are served when passing through routers. There are eight bits in the TOS (Type Of Service) field of the IP header. It is proposed that two of these bits be used as follows: 1 bit used to signify that a packet is *premium*, and one a bit to indicate if the packet is *inside* or *outside* the specified limits that were agreed, implicitly or explicitly, by the source, when it initiated the traffic stream of which this packet is a component. Best-effort traffic would always have this last bit reset, indicating that it is outside the specified limits, even though this is not really strictly true.

In a way, the key idea is this: each router is expected to maintain two queues for traffic which has a high priority than best-effort: the PQ for *premium traffic* and the AQ for the *assured traffic* and all other traffic. . Maintaining such queues, even if it has to be done in *every* router, should not be particularly difficult to implement because of the fact that it is only necessary to distinguish a limited number of different traffic classes.

The premium traffic is given priority treatment at all queues in which the DiffServ architecture has been implemented, providing a service which should have minimal loss, delay, and jitter. The assured traffic has lower priority than the premium traffic but has some advantages over the best-effort traffic (i.e. all the rest) in that although it

is buffered in the same first-in first-out queue, when the queue rises above the ;threshhold where packets must be dropped, the best-effort packets are dropped first. If the buffer continues to build up because of excess load, and a second threshhold is exceeded, assured service packets will also be dropped.

There is more to the DiffServ architecture than prioritising packet forwarding at routers though.

5.3.4 Service Level Agreements

In the DiffServ architecture, traffic flows in either the *premium* or the *assured* classes must, either statically or dynamically agree to a certain *profile* for the traffic. This profile is specified in a *Service Level Agreement (SLA)*. If this is static, it is agreed off-line and remains in force indefinitely, applying to all the traffic specified in the agreement. Dynamical settings for a Service Level Agreement can be assigned by means of the RSVP protocol [10].

The traffic profile is used at the entry point of packets into a network, and also, in an aggregate manner, at the border between one routing domain and another. The way the profile that is used depends on whether it is the premium service or the assured service that has been requested. In the former case, traffic is shaped, by buffering, as far as is possible, and by dropping packets if buffering is not adequate. In this way, the peak rate specified in the SLA is rigidly enforced at the entry point to the network. In the case of the assured service, packets which transgress the profile specified in the SLA are simply not marked with the assured priority bit.

5.3.5 Random Early Dropping of In and Out Packets (RIO)

To complete the picture, we also should specify how packets in the premium and assured classes will be treated if they encounter congestion. In the case of the premium packets, this shouldn't happen. In the case of the assured and the best effort services, packets will be dropped whenever a router buffer level rises above a certain threshhold. As specified earlier, best-effort packets will be dropped first and later, if a second threshhold is exceeded, assured traffic packets will be dropped as well.

5.4 New Approaches to Routing in the Internet

5.4.1 Layered Routing – MPLS

The approach to routing adopted in networks with Connection Admission Control (CAC) tends to be to select a *path* for a given end-to-end communication and arrange that all packets of a certain connection follow precisely the same path, for as long as this connection exists. Examples of this sort include X.25, frame relay, and ATM networks.

In networks with connectionless routing, such as the Internet, it has not been possible, or at least not the usual practise, to establish a path for use by an end-to-end connection. The fact that the path for communication between a certain source and a certain destination might change is not necessarily a problem, and might even be considered an advantage. However, reasons for requiring and techniques for guiding the packets of certain end-to-end connections down the same communication path over and over again are now beginning to emerge in the Internet.

One technique with this side-effect is the Resource ReSerVation Protocol (RSVP) [15, 10]. This protocol was mentioned earlier in connection with dynamic SLA's. We shall not concern ourselves with the other possible uses of RSVP at the moment. See §5.4.2 for a more detailed treatment of RSVP.

Another way in which consistent paths through a network arise naturally in the Internet is where the TCP/IP layer makes use of another connection oriented network as a transport medium, at layer 2. Historically, the first way in which this arose was by the Internet making use of an underlying ATM network [16]. Since routing is more expensive per switched byte than ATM switching (at least this was the case previously and is probably still the case at the time of writing), it makes sense to use as many ATM hops as possible if by this means routing hops can be avoided.

This approach can also be used if the switches are not ATM switches. They could be ethernet switches, for example. This would seem to be inappropiate, since ethernet cannot be used (readily anyway) to traverse long distances. However, in a context like a campus or private TCP/IP network of moderate size, the use of ethernet

switching *instead of routing* wherever possible has the effect of dramatically lowering the load on routers. This was discussed in connection with switch-routers in §5.1.7.

The general scheme for this sort of thing, in which the layer 2 service can be provided by a variety of different switching architectures, or a mixture of such architectures, is known as *Multi-Protocol Label Switching* (MPLS) [17]. In principle, a path might be made up of links connecting a variety of different types of switches and routers. Each switch, however, has in common the ability to interpret a field in incoming packets known as the *label*. Just as in ATM or X.25 routing (and any other network in which routing occurs on top of the connection function), the next link and the next destination of a packet are determined from a table in which the incoming link and the *label* are the critical parameters. Each switch has to maintain such a table. Or, perhaps it might be more accurate to say that one such *forwarding table* must be maintained for each incoming port. This is exactly the type of table which is set up in an ATM switch to support its natural mode of connection oriented switching already.

An example of such a forwarding table is shown in Figure 5.5.

Figure 5.8: A Routing Table

Incoming Label	Outgoing port	Outgoing Label
0x83461	35	0x55823
0x622B8	22	0x86A82
0x08A34	17	0x95238
0x0C228	12	0x038C7

The *label* on a packet or cell does not have to be as long as a TCP/IP address because the identifier in this label does not have to be globally unique, only unique among all the packets on this link. In fact, however, the address range which has generally been adopted for labels is usually longer than 32 bits. The label does not have to have exactly the same format all the time – it could be 48 bits in one medium and 32 in another. In fact, it is to be expected that the label will be of a different length, etc., when the packet passes onto a different type of link.

MPLS can also be supported by TCP/IP running over the top of IP. In this case, the label is stored in a *shim header* [17], a lightweight IP-like header which fits *between* the layer 3 (TCP or UDP) and the layer 2 (IP) headers and is confined to containing just what is necessary for the MPLS *forwarding* function. Because of the simplicity and economy of the IP over IP version of MPLS, there appears to be some inclination to think of this as the *true MPLS*.

In the IP over IP version of MPLS, the extra overhead of the additional layer is not a great deal and so it is natural to consider the possibility of repeating the trick, and carrying IP packets over an IP sub-layer which is in itself carried in another sub-layer, and so on. This is explicitly allowed for in the definition of MPLS, and hence the standard refers to a *stack* of labels associated with a carried packet. Each packet will be forwarded at the routers along its path in accordance with the label at the top of this stack until the table at a router indicates that it is time to remove a label from the stack and pass the packet back to the router. However, if there is another label on the stack, the router in question will still not have to actually route the packet. Instead, it should use the label which has been uncovered to direct the next forwarding step.

It seems unlikely that in real networks these label stacks will regularly build up to more than one or two layers in height. However, the general framework fits naturally within the MPLS concept and it costs nothing in extra overhead to allow for this in the architecture, so it is natural that the standard should adopt this degree of generality.

The only other field aside from the label required in the *IP Shim header* according to the MPLS standard is a TTL field. The role of this field is to ensure that the TTL field of packets is counted down at each hop through the network, even the ones where forwarding is carried out by means of MPLS. If this was not done, the purpose of the IP TTL field could be subverted and packets might be trapped into endless loops. The MPLS TTL field is required in all the alternative layer 2 implementations of MPLS. When ATM is used as the layer 2 of MPLS, a TTL field is added to the ATM Adaption Layer header which is added as IP packets are segmented into ATM cells.

The history of MPLS is not long enough at this stage to be sure that MPLS will be successul in any of its forms. However, it does have the potential to enable IP traffic to be routed at the speed and cost of ATM cells without the need for an overly complex modification to the IP routing framework. If the Internet is to continue expanding, or perhaps even expan at an increasing rate, as new services are defined and developed, MPLS or some other mechanism which delivers significantly better performance from routers will be necessary.

For more discussion of MPLS, see §7.3.5

5.4.2 Resource ReSerVation Protocol (RSVP)

The Resource ReSerVation Protocol was introduced quite some time ago with a view to supporting real time video and audio services over the Internet, possibly in conjunction with *IP multicasting*. IP Multicasting is a technique which allows video-conferencing and audio conferencing to take place over the Internet, with large numbers of participants, with great efficiency. The efficiency of this technique is a great deal better than the equivalent collection of point-to-point links because a packet going to two nearby destinations is duplicated at approximately the latest possible moment.

IP multicasting and other audio and video distribution protocols require a level of protection of communication resources which is not readily achieved in the Internet – certainly not by the usual protocols, which provide a basic, best-effort service for all participants. There is provision for priority, and for differentiation on the basis of type of service, however the use of these bits is not well standardised and they have largely fallen into disuse except for within a private network (and except for the fact that new uses for these bits arte now be actively promoted – see [11, 18, 12]).

The RSVP protocol is not concerned with providing differentiated grades of service for different traffic classes – there are other approaches to achieve that. RSVP is targetted at ensuring that the path(s) packets will take once a connection has been established will have sufficient resources to carry all the offered traffic at the required grade of service.

At the time when it was defined, RSVP was attempting to achieve this for individual unicast or multicast connections. For example, perhaps it is desired to broadcast a convert over the Internet. RSVP could be used to reserve the required bandwidth for this planned broadcast.

More recently, in the DiffServ architecture, a more aggregated approach to resource reservation has been proposed. Instead of undertaking reservations for every new connection, the idea in this case is to make reservations, or alter reservations, when required, to allow for growth or decline of the aggregate demand for resources of all the traffic in a broad class requiring prioritised treatment in routers.

5.5 Examples

Example 5.3. A Laboratory

Let us continue from where we left off in Chapter 3. In particular, let us assume that the laboratory is set up as in Figure 3.4. A question which we might usefully address in this chapter is this: how should we configure the routing?

We have so many options to consider, let's make a list:

(i) What technology should be used for the LAN?

(ii) What address range should be used? What subnet mask?

(iii) Should we be using a switch-router? Should we be putting the computers of a LAB in a vLAN? Is there a reason to use cut-through switching?

(iv) How should we configure the routing in the case where there are several of these Laboratories all making use of the same server?

Answers

(i) A laboratory is a performance critical environment and traffic levels could easily reach the level where performance suffers; as a consequence, it would be advisable to use switched 100 Mbit/s ethernet, for example, or another layer 2 medium with similar performance. the server(s) could usefully be equipped with more than one network interface card.

(ii) Since there will not be any servers located in a laboratory, it would be sensible to use private IP addresses for a laboratory. The server or servers made use of by the laboratory computers also do not need to be visible from elsewhere in the Internet, and so it makes sense to put these machines in the chosen private IP address range also. This means that the organisation where this laboratory is located will need to provide a NAT

capable gateway if the laboratory machines are to have access to the Internet. Software for such a gateway is readily available.

(iii) The vast majority of traffic in the laboratory will be between the lab machines and the servers. The laboratory machines and the server(s) should be in the same physical LAN or the same vLAN. If this is ensured, the remaining load (for Internet access, for example) should not be a problem. Use of a vLAN for the laboratory could have some advantages in that it would then be easier to reconfigure the topology of the laboratory and associated equipment, however it doesn't appear to be a critical requirement of the laboratory.

(iv) Suppose several laboratories all use the same server. Then, a simple strategy for maintaining good security and performance would be to put the server in the same physical LAN or the same vLAN as all the laboratory machines.

□

Example 5.4. A School

The routers in a school can most likely be counted on one hand. In a relatively simple case, there might be one router at the gateway to the Internet and one other router.

The routing plan in a school should be quite obvious. The default route for all packets should be to pass through the gateway to the Internet. Each other subnet connected to the router will need an explicit route.

Some reasons for subdividing a school into a number of subnetworks are as follows:

(i) so that traffic can be segregated, for security reasons,

(ii) to reduce traffic on each separate subnetwork.

Because of security considerations, it will be advisable to put in place a collection of rules in the router which filter out packets which should not be necessary within the normal pattern of life in a school. For example any access from a student laboratory to the school's administration subnetwork could be blocked.

The router at the gateway will probably have network address translation in operation. If the school has a public IP address to which the rest of the Internet needs access, this could be placed on the other side of the router which implements NAT or alternatively put this public server on the private side of the gateway router and put a host route into the router to direct packets to the right side of the gateway. □

Example 5.5. A Campus

Let us suppose that the campus is layed out as in Figure 5.9, except that there are perhaps two or three additional Faculties which are not shown in the map.

Reliability is important in a campus network – sufficiently important that the campus network must include a basic loop which allows each building to communicate with each other building along at least two disjoint paths. A logical diagram of the campus network is shown in Figure 5.10. The complexity and size of a campus network would be an order of magnitude greater than depicted in this diagram.

The vLAN concept is designed with campus networks in mind and is ideally suited to it. The advantages of using vLANs instead of physical LANs are primarily convenience, however. For example, if staff from Faulty A are moved into a building previously occupied by Faculty B, it should be a simple matter to reassign the ports of these staff to the vLAN used for Faulty A.

There is a considerable potential for heavy load and high growth in a campus network. For this reason it makes sense to use whatever means are available to reduce load on the routers. One method would be to use ATM at Layer 2 throughout the campus and then use a protocol which diverts flows directly through ATM switches rather than through the router. However, this strategy can be used without ATM. In any case, because ATM network interface cards are so expensive, and because of the huge investment in existing technology, the client machines of the university will still need to connect to ethernet, not ATM. And finally, cut-through switching based on an ethernet layer 2 and an appropriate Layer 3 can achieve extremely good router efficiencies. A campus network is sufficiently small that the lack of a sophisticated signaling protocol for interconnected ethernet switches will not present a problem.

It makes sense to use private addresses for as much of the camus network as possible. This does not appear to be common practise at the moment. However, aside from conserving IP address space, which is apparantly not

Figure 5.9: A Campus

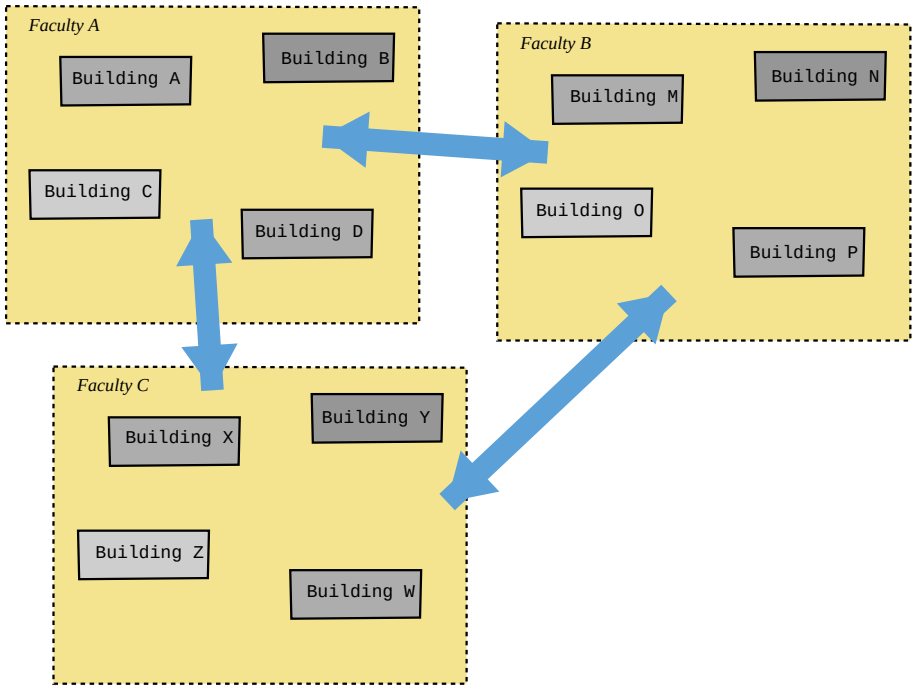
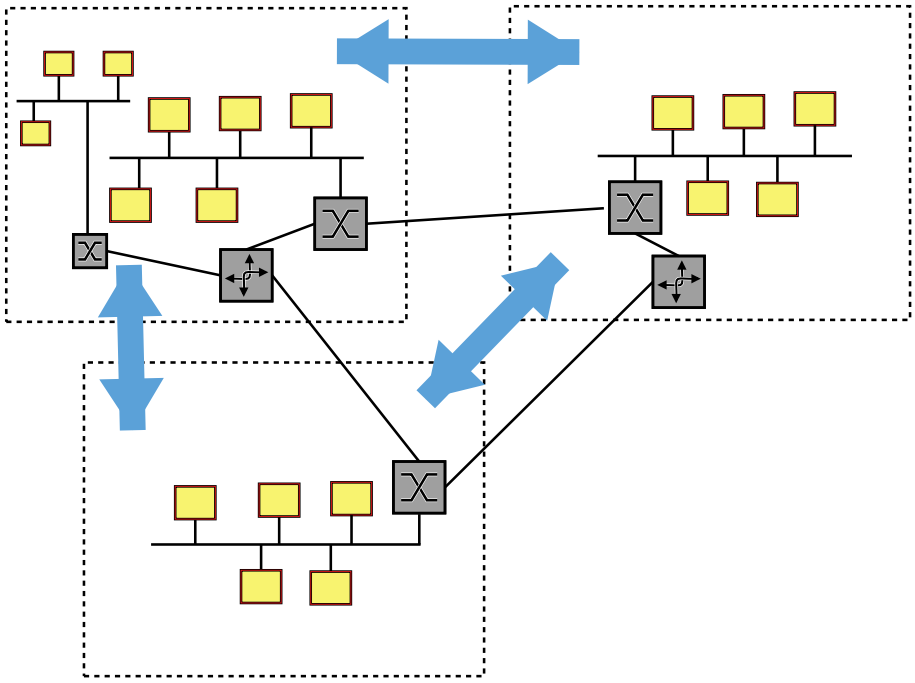


Figure 5.10: A Campus



a real concern, use of private IP addresses inside the campus has the advantage of providing virtually automatic protection against access from outside the campus.

One of the things that makes a campus interesting is the fact that there is usually the desire and the opportunity to introduce new concepts and new technology at an early opportunity. Examples of new technology which are ripe for exploitation in a campus environment include: voice over IP, wireless access networks, and video-conferencing.

Voice over IP

It appears that voice over IP is already financially attractive in the campus environment. As voice over IP handsets become cheaper, the movement away from traditional telephony will accelarate. Routing in the campus, and between campuses, needs to be configured to give voice packets priority. As voice over IP becomes more widespread, it will become more important that a consistent interpretation of the TOS bits is used throughout the Internet. See §5.3.3.

Wireless Access

It is conceivable that laboratories could be replaced by wireless access. An advantage of such a step would be that the administrative burden of maintaining computer facilities could then, to some degree, be transferred to students. Wireless access interface cards could be used in place of wired access when a workstation in moved to a new location at short notice. Routing of wireless access in a campus might be arranged so that workstations join a certain vLAN by default unless the MAC address of the workstation is present in a certain list. A mechanism of this sort is needed to ensure that separation into the three broad classifications of administrative, academic, and student subnetworks is maintained.

Video Conferencing

Video-conferencing has been waiting in the wings for a long time. However, the potential importance of video conferencing remains considerable. Furthermore, if it does achieve widespread usage, for teaching purposes, it will have a considerable impact on traffic levels. Because of the presence of voice in the video, this traffic will need to be given special treatment. If this service takes off, traffic levels might need to be monitored to avoid degradation of service.

External Access to the Campus

Students and staff often need to access campus facilities from off campus. Since security of services on campus is based to a significant degree on filtering based on IP address and vLAN membership, off campus access is likely to be severely limited. A solution to this problem is to provide a flexible virtual private network facility by means of which staff and students can *appear* to be on campus even when they actually connect from a remote location. In order to preserve security, an extra layer of authentication should be enforced on entry to this facility.

□

Example 5.6. A Statewide Retail Organisation

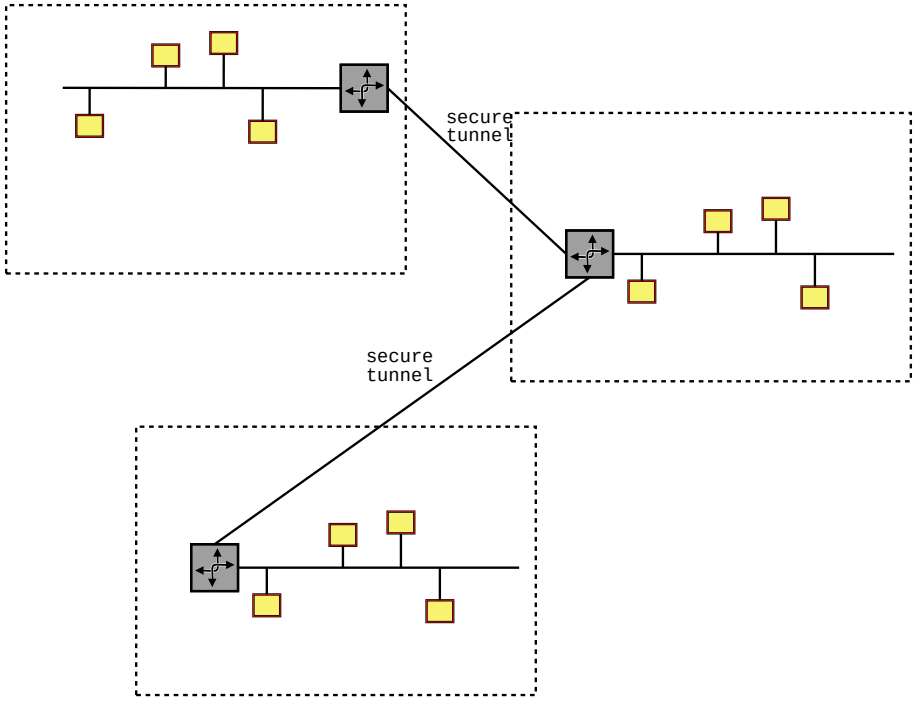
Since communication between the Internet and this private network is only required in a very limited and well-defined manner, a private range of IP addresses should be used for the entire network, except the servers which provide web services to the Internet.

Different sites will need to be connected by means of secure tunnels. A logical network diagram is shown in Figure 5.11.

The routers at each site in this diagram do not need to handle high throughput and therefore could easily be implemented in software inside the gateway machine. This gateway machine could provide a variety of local services, such as email, DNS, file sharing, in addition to providing the gateway.

The tunnels between different locations can be provided by dedicated (leased) lines or by means of a connection though the Internet, or both. The reason for using two methods is reliability. If the cost of a leased line is considered excessive, it will be necessary to maintain *two* ISP services. One ISP could be regarded as a backup for the other,

Figure 5.11: A State Wide Virtual Private Network



however both services should be used regularly so that information about the reliability and performance of both ISPs is kept up-to-date.

The virtual private network can, in principle, be used for data, voice, and other services. If the organisation spans a considerable geographical distance, significant savings on telephone costs will be possible by using the network for telephony. This does not necessarily mean using Voice over IP, however it seems likely that this will prove, in the near term future, to be the simplest and cheapest way to set up a private telephone network service over this network. Tunnels through the public Internet might not be adequate to provide this type of service, however the capacity available on these tunnels could conceivably be configurable, depending on the ISP. This is the sort of service which could be provided by an Internet which implements the DiffServ architecture (see §5.3.3). Also, if a common ISP is used at all sites, the IP tunnels might remain entirely inside the domain managed by one ISP.

If telephony *is* carried on the private network the need for a reliable backup service becomes even greater. It might be necessary, for example, to include a point-to-point dial-up digital service in the range of options which can be used to connect routers in different segments of the organisation. This could be provided by the ISDN service of a telecommunications provider. Since the charge for dial-up ISDN services is divided into rental and usage charges, and the usages charges of a backup facility would be very low, depending on the rental fee, dial-up ISDN could be an ideal backup facility. Regular use of this facility to make sure that it will work smoothly when required is recommended. □

Example 5.7. A National ISP

A nation-wide ISP can take many forms. Let us simply review the relevance of the advanced routing features discussed above to the particular role of a national ISP.

In order to remain attractive to the broadest possible range of customers, an ISP must attempt to provide new routing facilities as soon as it is clear that these will be attractive and that they will remain a long term feature of the Internet. The features which could potentially fall into this category include the following:

1. MPLS;
2. RSVP;

3. DiffServ;

4. Voice over IP.

The last service has the potential to be revenue positive for customers now. An ISP can become involved by providing gateways from their own voice-over-IP service to the public telepohone network. However, in order for this voice-over-IP service to be satisfactory, it is essential that its quality of service be maintained throughout the ISP network.

This is where item 3 becomes relevant. The DiffServ architecture is not fully defined, however there is no reason that an ISP cannot implement a version of it at this stage. The ISP may then be able to preserve quality of service for selected services (eg IP tunnels for customers, as in the previous example, and voice-over IP).

The RSVP service potentially forms part of the DiffServ architecture, and implementation of this service may prove necessary on this account. Without some sort of implementation of RSVP, management of the bandwidth allocated to premium services will be somewhat limited. In a modest network this might not necessarily be a problem, however.

The MPLS service is primarily targetted at increasing throughput and efficiency, and lowering delay, of large TCP/IP networks. Only the largest ISPs will therefore need to consider implementing MPLS. For ISPs in this category, however, adoption of MPLS should be seriously considered as an important method for controlling the load on central routers. □

5.6 Closing Comments and Summary

In this chapter we have introduced the fundamental ideas concerning routing and the most important of the practical approaches to routing which are in common use in today's networks. Particular attention was placed upon the contrast between traditional telephone network and ATM routing on the one hand and traditional shortest-path stateless routing in the Internet on the other hand. The ways in which these two approaches are now merging in the new routing strategies in the Internet such as RSVP, DiffServ, and MPLS were then reviewed.

References

[1] P. Almquist. Ruminations on route leaking. Technical Report -, IETF, 1992.

[2] K. Varadhan. Bgp ospf interaction. Technical Report RFC 1403, IETF, 1993.

[3] K. Lougheed and Y. Rekhter. A border gateway protocol 3 (bgp-3). Technical Report RFC 1267, IETF, 1991.

[4] G. Malkin. Rip version 2. Technical Report RFC 2453, IETF, 1998.

[5] J. Moy. Ospf version 2. Technical Report RFC 2328, IETF, April 1998.

[6] Y. Rekhter and T. Li. A border gateway protocol 4(bgp-4). Technical Report RFC 1771, IETF, 1995.

[7] V. Paxson. End-to-end routing behavior in the internet. *IEEE/ACM Transactions on Networking*, 1997.

[8] IEEE SA Standards Board. IEEE std 802.1q-1998. standards for local and metropolitan area networks: Virtual bridged local area networks. Technical report, IEEE, 1998.

[9] R. Gilligan and E. Nordmark. Transition mechanisms for ipv6 hosts and routers. Technical Report RFC 1933, IETF, 1996.

[10] Ed R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource reservation protocol (rsvp) – version 1 functional specification. Technical Report RFC 2205, IETF, 1995.

[11] K. Nichols, V. Jacobson, and L. Zhang. A two-bit differentiated services architecture for the internet. Technical Report RFC 2638, IETF, 1999.

[12] Xipeng Xiao and Lionel M. Ni. Internet qos: A big picture. *IEEE Network Magazine*, 1999.

[13] R. Jain. Myths about congestion management in high speed networks. *Internetworking: Res. and Exp*, 3, 1992.

[14] Sally Floyd and Van Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, 1993.

[15] Lixia Zhang, Steve Deering, Deborah Estrin, Scott Shenker, and Daniel Zappala. RSVP: A new resource reservation protocol. *IEEE network*, 1993.

[16] Peter Newman, Greg Minshall, and Tom Lyon. IP switching: ATM under IP. *IEEE/ACM Transactions on Networking*, 1998.

[17] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol label switching architecture. Technical Report RFC 3031, IETF, 2001.

[18] R. Guerin and Ariel Orda. Qos-based routing in networks with inaccurate information: Theory and algorithms. *IEEE/ACM Transactions on Networking*, 7(3):350–364, June 1999.

Chapter 6

Requirements Analysis

This chapter is about how to proceed in a structured manner to determining the networking requirements of an organization and to estimate the quantities relevant to network design.

The book by McCabe [1] is particularly strong in the area of requirements analysis. See especially [Chapters 2-4]. The approach we take here will nevertheless be a little different, as we base the concept of network requirements analysis on the concept of a *traffic stream*.

As in the design of software, in one sense, the appropriate starting point for network design is an assessment of the *requirements* of the client or clients. As with software, also, it is necessary to go beyond simply asking the client(s) what they want or need. The client will probably not have the expertise to know what their future network requirements will be. In fact, a collaborative procedure should be adopted to tease out the nature of these requirements.

In order that we have the skill to tease out and fill in the details of the clients real communications needs, we need to have a good understanding of certain concepts by means of which these needs can be descibed, categorized, and quantitied. That is the main subject matter of this chapter.

6.1 Traffic Streams

A traditional concept in teletraffic theory is that the clients or customers of a network come to their network interfaces with certain intentions, or perhaps needs, for communication. In the simplest case, this could be the intention, on the behalf of a user, A, to engage in a communication with another user, B, over a certain period of time, at a certain bit rate, and with certain performance requirements.

This intention, or need, will become translated into actual traffic as soon as the network is provided, and the end-points of the intended traffic are in place. How much traffic actually flows, and the performance it experiences will depend on the resources of the network and the other traffic offered to the network at the same time. So, an actual traffic flow anything like the desired flow is not likely to occur: it is an idealization.

This idealization is nevertheless a useful concept. We call it a *traffic stream*.

It should be clear that there is an important distinction between the traffic that users would like their networks to carry and the traffic that is actually carried. On the other hand, these are also very similar concepts. We will use the same concepts and the same parameters to describe both. The words *traffic* and *traffic stream* will be used, ambiguously, to refer to both *offered traffic* and the traffic flowing in real networks (*carried traffic*).

Traffic streams will typically have the following basic features (varying somewhat depending on the type of traffic), or parameters:

- throughput requirement (mean and standard deviation, peak load, in bytes/sec, Hurst parameter);
- delay performance requirement (mean and standard deviation, in milliseconds);
- loss requirement (a probability);
- packet length – mean and standard deviation;

- source and destination (each may be a single location, or a set of locations).

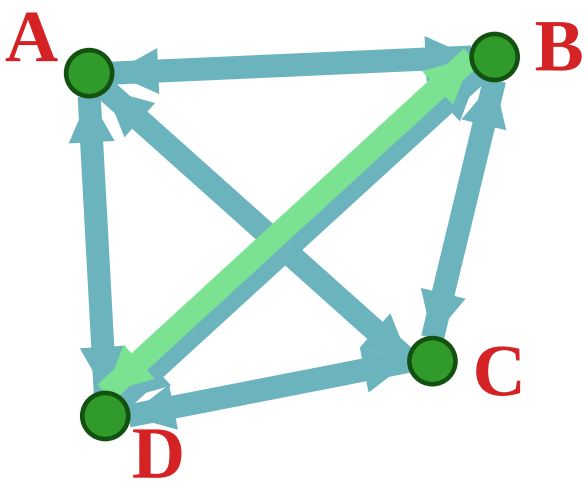
The critical parameters here are the first, the throughput requirement, however the other parameters are also important. The Hurst parameter may or may not be important. From a theoretical point of view the Hurst parameter is very important, but from a practical point of view its impact and importance is not quite as great. The mean and standard deviation *are* important.

Traffic streams may also require a clear end-to-end synchronous bit-path, for example for telephony or video transmission, although this is more and more thought of now as a limiting case of packet communication. If we were to require, for example, that the standard deviation of packet delay were zero, we would, in effect, be specifying synchronous transmission – any packet communication link with zero standard deviation for packet delay could readily be used to provide a synchronous transmission link.

A traffic stream with very tight performance constraints is much harder to provide for than one with very weak constraints. In principle, such a traffic stream might have to be segregated from other (more relaxed) traffic streams in order that its requirements can be met. Alternatively, such a traffic stream could be *treated differently*.

Suppose two traffic streams, t_1 and t_2 with the characteristics shown in Table 6.1 are required to be carried in the network depicted in Figure 6.1. The first traffic stream (in the darker shade) is distributed over the whole network. We assume that all sources and destinations are equally likely for this distributed traffic stream. the second traffic stream passes between nodes B and D only.

Figure 6.1: A tail of two traffic streams



Stream and Attributes	Value
t_1: bit rate	peak 20 Mbit/s, mean: 2 Mbit/s, σ: 4 Mbit/s
t_1: delay	mean: 200 milliseconds, σ: 400 milliseconds
t_1: loss	mean: 0.04
t_1: origin	A-D
t_1: destination	A-D
t_2: bit rate	peak 4 Mbit/s, mean: 4 Mbit/s, σ: 0 Mbit/s
t_2: delay	mean: 100 milliseconds, σ: 0 milliseconds
t_2: loss	mean: 0
t_2: origin	B
t_2: destination	D

Table 6.1: Some traffic streams

The first of these traffic streams has a higher mean bit rate requirement, but it will actually be a lot easier to

deal with than the second stream because it is much more tolerant of loss, delay, and delay variation. The second traffic stream is totally intolerant of delay variation.

This means that if the two streams were to be carried on the same link, it might be necessary to *segregate* these streams, i.e. the capacity required to carry traffic stream t_2 would have to be *reserved*. Another alternative would be for all packets in traffic stream t_2 to take priority in the output buffers at the nodes *B* and *D*. If the Internet DiffServ architecture was available and implemented in these nodes, traffic stream t_2 could be carried by the *premium service* (see Subsection 3.5.5 and Subsection 5.3.3) and the rest of the traffic could be carried using either the *assured service* or the usual best-effort traffic class of the Internet.

6.2 Services

A simple way to survey the needs of a networks clients is to enumerate the *services* to be provided across the network. Typical examples are: file-sharing, email, web access, FTP access, database access, intra-net (internal web) access, X-windows traffic, computation service communication, remote conferencing.

When this list has been prepared, the next thing to think about is the characteristics of the traffic streams associated with each service.

In many cases, services give rise to traffic streams from a class, or set, of computers to, or from, or both to and from, a single *different* computer. A good example of this is web access. The web server engages in a great deal of communication with all the computers in the vicinity and possibly with computers all over the world.

Example 6.1. Traffic Streams in a Campus Network

The following *services* can be expected to contribute significant traffic streams in a campus context:

- File Services
  - SMB (windows file sharing protocol),
  - NFS (Unix file sharing),
  - Novell,
  - appleshare.

  File sharing is a great user of networking resources. Laboratories, in particular, make an enormous amount of use of file sharing, and this can be very demanding on the network. An obvious strategy for minimizing the impact of file-sharing on a network is to locate file servers as close as possible to their clients.
- Printing
- Intranet Access
  - web access,
  - FTP access,
  - email,
  - chat.

  In universities, and schools and many businesses, an Intranet (web, email and FTP for internal use) provides more and more of the information services required by the organization.
- Internet Access
  - web access,
  - ftp access,
  - email,
  - chat.

The traffic associated with Internet access has been, up to now, somewhat limited by the fact that many of the resources on the Internet cannot deliver high bandwidths. It seems likely that this will gradually change as capacities in the Internet change. The capacity of the Internet itself, and the link to the Internet, often imposes a choke on such services. However, there is a considerable and growing demand for Internet access and as the services which are considered an acceptable part of normal work practise grow in volume and importance, these limitations are also likely to grow.

The road network has been suggested as a useful analogy to apply to the Internet. The point of the analogy is that as road networks have improved, limitations upon travel have become more relaxed, and the central freeways and highways have improved to the point where communication over considerable distances can be achieved economically and at a good speed. Over time we might reasonably expect the ratio between local traffic and traffic to *other* locations to decrease.

- Database Access

  Database access has been an important service on campus and organization networks for some time, although these services are tending to migrate to the Intranet. When this happens, part of the access load migrates to the Intranet category, although the need, and the traffic, is not obviated by this transition.

  On the other hand, databases also have more roles and new roles so that database traffic is probably increasing at roughly the same rate as many other types of traffic. As Internet speed and capacities improve, database access over longer distances may well increase. At present, database outside a local LAN is probably quite rare, however it is possible that this could change.

- Application Access

  Sometimes it is useful to run applications on a remote computer. This is different from loading an application from a remote computer and running it locally. In either case, the remote computer can legitimately be called an *application server*. However, in the latter case, the server is really providing *file services*, not an application service, and in this classification of traffic types this case has already been dealt with above. The former case does need to be distinguished.

  An example which might occur on a university campus is a tutorial booking system, which would be used intensively for a few weeks at the start of each semester. However, nowadays this sort of functionality can be readily incorporated into an Intranet.

  Another example is the remote use of parallel computing facilities. It is common for university laboratories to be used as parallel computation facilities for research at times when the laboratories are not used for classes. Communication between the user of these facilities and the computers in question make use of network resources. A parallel computation facility provided by a collection of computers in a laboratory also, obviously, makes heavy use of network resources on the laboratories LAN.

  One more example is a *games engine* to which a collection of gaming client machines are connected. There is a great deal of this activity taking place in networks today, although this sort of use of networks is often considered frivolous. (Not by the game software companies though!)

  In any case, although games are not usually a *legitimate* use of university or business or school resources, it may be unnecessary and difficult to prevent their use. Game software is readily accessible and amongst the students and staff population of a university or school there are likely to be a significant number of people interested in running games software.

- X-server protocol traffic

  The Unix operating system allows for an application running on one computer to display its results and accept input from a different computer. This is often useful, and especially when the application software is restricted for use on a particular computer. However, the consequence is that a great deal of network traffic is generated – all the instructions to draw a circle, a square, lay out a bit pattern, and so on, have to be transferred across the network.

- Internet and LAN management protocols

  - DNS access,
  - router protocols,
  - ARP and other broadcast protocols in LANs.

  An individual LAN requires a certain amount of broadcast traffic simply to support the other functions of the LAN. Similarly, routers and DNS servers need to communicate between themselves, and all hosts need to communicate with DNS servers. This traffic can reach significant levels, depending upon the protocol in use. In particular, it is not unusual for router communication protocols to occupy a significant proportion of the bandwidth of a network.

- Telephony is a traditional and important load on an organizations networks. In the past, the telephone network would be completely separate, and even now, telephone traffic is not normally carried on the TCP/IP network. However, sharing of resources *is* occurring at some level in more and more organizations. The raw transmission resources of an organization are often used for telephone traffic as well as for TCP/IP networking. Telephone traffic is normally carried as a collection of synchronous bit-streams, one for each call in progress. However, fluctuation of the number of calls in progress will, naturally, cause the offered, and carried, traffic, to exhibit a significant variance.

- Video Access and Distribution

  Video traffic is already a significant component of total traffic in national communication networks. Broadcast television requires a national network to support transfer of television signals at various stages of development from one place to another. The final product also has to be distributed to the locations from which it has to be transmitted. Because advertising tends to be more local in content than material which people actually turn on to watch, a variety of sources are merged together at the point where broadcasting takes place.

  Cable TV requires terrestrial networks even for the distribution of the video signal to the home. Cable TV cannot be distributed to the home on a twisted pair cable, as is used for telephony – not at present, anyway. There are two options: optical fiber to the home, which is not common, because of the cost of the optical fiber cable and of the terminal equipment with which it must be equipped, and coaxial cable. Coaxial cable is capable of carrying high bandwidth signals, in either digital or analog form.

  Once a home is connected to a cable TV network, it becomes natural to consider additional uses of the high capacity cable which has been installed. It is already common for cable television access facilities to be used for telephone access and Internet access.

  Video traffic is not a common component on today's TCP/IP networks, however this could change. If multicast protocols become widely used for distribution of video conferencing and video broadcasting, this component of traffic could expand greatly over a short period of time, putting great demands on networks. The successful handling of video traffic might also require improvements in the way quality of service of individual traffic streams is protected.

□

6.2.1 Tabulation of Demand for Services

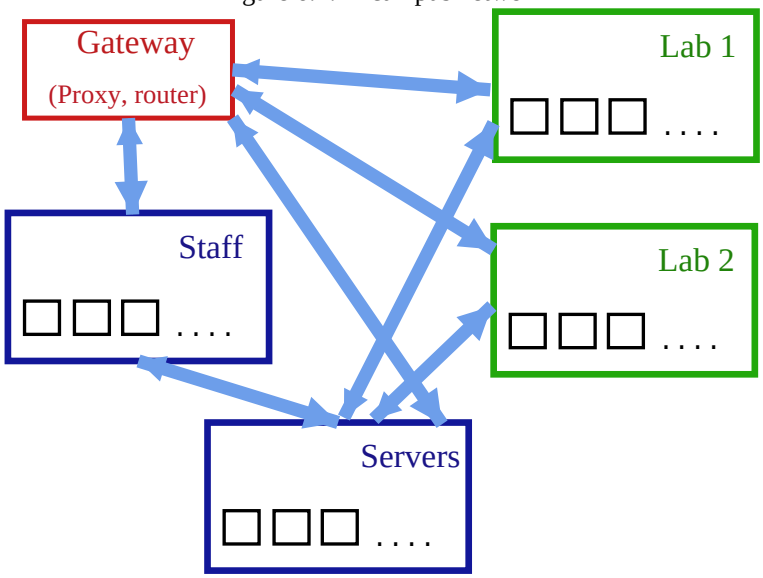
For convenience, a form for recording current and forecast traffic levels is shown in Figure 6.3.

Example 6.2. A Campus Traffic Survey

A traffic survey, made using Figure 6.3 has been prepared and is displayed in Figure 6.4.

□

Figure 6.2: A campus network



6.3 Growth and Forecasting of future traffic

We know growth is high – almost wherever we look. The Internet at large, and each individual enterprise network is expected to grow at a very significant rate for the next year or two. After that, who knows?

Because growth is currently so high, it is appropriate that all network resources should be *over-dimensioned* by a considerable margin, that is to say, much more capacity should be installed than would be required for the obvious reasons.

Exercise 6.1. Summarised Traffic

Summarise the traffic classes in Figure 6.4 into just three classes of traffic: *best effort* traffic, which is not performance sensitive, *high priority* traffic which *is* loss sensitive, but can tolerate delay, and *voice quality* traffic, which cannot tolerate significant delay at all, but can tolerate a small amount of loss. □

Exercise 6.2. Traffic in the Future

Starting with Figure 6.4, estimate how the table might look in three years time. Use the growth estimates provided in the table or, if you wish, use your own judgement as to which traffic types are likely to grow at a faster rate. □

Exercise 6.3. Survey the Requirements for your Organization

Suppose you are a working network manager. If you are not a working network manager, imagine that you are. Draw up a list of the services and traffic streams which your network attempts to carry. Start off with the services, then add detail by specifying the traffic streams. The traffic streams should first be specified in as broad a manner as possible. For example, if there is a background type of traffic engaged in by all machines, this could be described as a traffic stream from every machine to every other machine.

You should use Figure 6.3 as a starting point for this exercise and modify it as you see fit (if necessary). □

6.4 Closing Comments and Summary

Specifying the requirements for any new service or product is difficult. Accuracy is barely achievable, except in the rare cases where a new installation can be expected to follow a pattern already established. However, like planning, requirements analysis is one of those tasks which is useful even when it is bound to “fail”. If the process

Service	Source	Destination	Volume	Statistics	Performance Requirements	Growth
File Services						
Printing						
Internet						
Database Access						
Application Access						
X Protocol						
Protocol Traffic						
Telephony						
Video Distribution						
Other						
Other						

Figure 6.3: A Table for Recording Traffic Demand

Service	**Source**	**Destination**	**Volume**	**Statistics**	**Performance Requirements**	**Growth**
File Services	All hosts (450)	File Servers (6)	200 kbps per host	H=0.8; $\sigma =$ 500	best-effort	20% pa
Printing	All hosts	Printers (25)	20 kbps	$\sigma = 50$	$<$ 5% loss, $<$ 5s delay	20% pa
Internet	All hosts	Gateway	20 kbps	$\sigma = 40$	$<$ 5% loss, $<$ 2s delay	30%
Database Access	35 hosts	DB Servers	50 kbps	$\sigma = 250$	$<$ 5% loss, $<$ 0.1s delay	15%
Application Access	All hosts	para.uni.edu	10 kbps	$\sigma = 200$	$<$ 5% loss, $<$ 1s delay	10–12%%
X Protocol	Unix hosts	para.uni.edu, mat.uni.edu, comp.uni.edu	100 kbps	$\sigma = 500$	$<$ 5% loss, $<$ 1s delay	10–12% %
Protocol Traffic	Routers	Routers & Gateway	250 kbps	$\sigma = 250$	$<$ 5% loss, $<$ 1s delay	15–18%
Telephony	Staff Offices	Gateway	$0.3 \times 64 \times$ 250kbps	$\sigma = 64 \times \sqrt{0.3 \times 250}$	$<$ 2% loss, $<$ 0.1s delay	5%
Video Distribution	?	?	?	?	?	?

Figure 6.4: A Campus Traffic Survey

of thinking about requirements is able to stimulate some fresh thinking on the job of how the network should look, it will have done its job. The reason we try to formalize the process of requirements analysis is merely to increase the chances for this to happen.

In summary, requirements can be expressed in two ways: by breaking them down into *services*, and into *traffic streams*. Traffic streams are an idealization of the flow of traffic which *wants to go* from one location to another. Each traffic stream has a number of parameters, such as *mean*, *standard deviation*, *tolerable loss*, *tolerable loss*, *tolerable jitter*, and so on. These parameters should be noted carefully, as far as this can be done, and where necessary, these characteristics should be respected in the design which is selected. In some cases, the special requirements could be interpreted as dictating that a certain traffic stream must be *segregated* from other types of traffic. Alternatively, it might be possible to assign traffic classes to one of the *premium*, *assured*, or *best effort* classes which are allowed for in the DiffServ architecture for traffic management in the Internet.

References

[1] James D. McCabe. *Practical Computer Network Analysis and Design*. Morgan Kauffman, 1998.

Chapter 7

Architecture

In this chapter we shall tackle the important concept of network architecture, and in particular, we shall try to understand the concepts of layering and hierarchical sub-division of networks. We shall then study the philosophy underlying the architecture of TCP/IP networks, together with that underlying the architecture of ATM networks and how, or whether, these different network architectures can be coordinated. In the last two sections we consider two other relevant architectures which exist in today's networks: security architecture and network management architecture.

Network architecture comprises, amongst other things: the protocols; the ideas of the protocols; and the relationships between the protocols. The choice of protocols and the choices concerning relationships between protocols is taken early, not at the time when a network is being maintained, not when it is being designed, and mostly before the protocols are even implemented. However, there are some architectural choices which can be made *after* the protocols have been designed – for example, sometimes an additional (existing) protocol layer can be inserted in a layered network architecture without affecting the other layers all that much.

Architecture is an aspect of networks which evolves rather slowly. Therefore, the decisions about architecture do not come along very often. Architectural decisions have to be taken well in advance of other decisions: when a major upgrade is contemplated, for example. The decision might be made to move to throughput and performance objectives an order of magnitude in advance of the current network. Or, it might be decided that all staff of an organisation should be networked at all times – dictating thereby that wirelass access is required in all sorts of places, and by means, that had previously been considered "out of reach". Or, it might be decided, at some stage in the future, that all devices which require maintenance must be networked. Or, perhaps it might be decided that all staff, clients, and active devices must hold *certificates* proving their identity!

It is not possible at this stage to anticipate the precise reasons why a major networking installation will be contemplated. However, we can be fairly certain that such major upgrades and expansions of networking will occur. When this happens we will need to make sensible and well-informed choices concerning architecture.

There are certain well-established architectural principles which have applied to networks in the past and can therefore be expected to apply to networks in the future. The key ideas concern *layering* and *hierarchy*. These ideas, and examples of how they are applied, are the subject of this chapter.

Before we start with the first principle, layering, let us consider *why* the two key principles are layering and hierarchy. There is a simple explanation. Architecture is really about structure, and structure is concerned with how a complex object may be sub-divided into simpler objects together with some simple principles by means of which these subobjects combine together. In the case of networks there are two natural approaches to sub-division: sub-division by logical function, and sub-division by geographical location. The first approach to sub-division leads to the principle of layering, and the second leads to the principle of hierarchy.

7.1 Layers

One of the key concepts in network architecture is *layering*, in particular, layering of protocols. The seven layer model known as the OSI (Open Systems Interconnection) Reference Model, developed by the ISO (Internation Standards Organisation) (jointly with other standards organisations) is a classic example. The OSI Reference

Model was not the first use of the concept of layering in networks and, although for a time it seemed that the OSI reference model was attempting to be the last word in layering, we now know that there are a great variety of ways in which the layering concept usefully arises in networks.

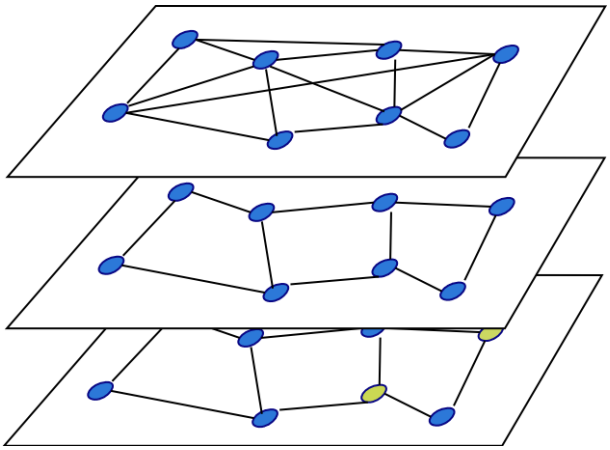
One the legacies of the OSI reference model is the idea that we can have a standard set of numbers by which we can refer to certain protocol layers. "In Layer 2 we shall use ...." "This protocol might be used in between Layer 2 and Layer 3 if ...." These sort of statements imply that there is a standard architecture of layers for protocols within which all other layers must find their place.

However, such a framework, if it ever really existed, should be viewed as a convenient fiction. The world of networking is constantly changing and any such framework can only have a certain limited life expectancy. At the moment we think of Layer 2 as the layer which provides connectivity at the local level, and Layer 3 as the layer which provides routing from one side to another across a wide area.

But there is something universal about the concept of a layered architecture. The universal idea is just the idea of stucturing services in layers.

In a layered model, each protocol layer provides a collection of *services* to the layers *above*, and it does this by making use of the collection of services in the layer, or layers, immediately below. See Figure 7.1.

Figure 7.1: Network Layers



One layer provides a service to another layer by means of so called *service primitives*. We will not need to go into this level of detail. However, some examples of service primitives might be:

(i) make a connection, C, from host A to host B;

(ii) send packet p to host B on connection C;

(iii) clear the connection C.

In the case of a connectionless service, such as the Internet Protocol (IP), there wouldn't be any primitives for dealing with connections, just primitives concerned with individual packets.

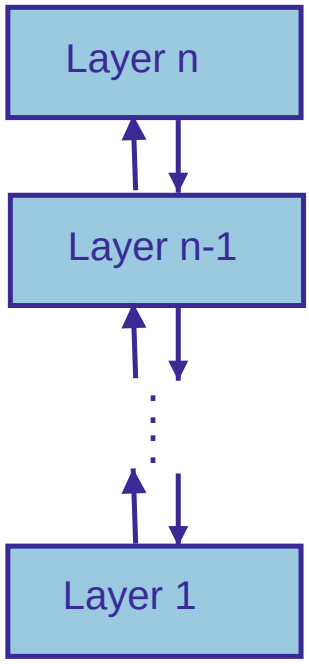
The classical arrangement of protocol layers is for a series of protocols to be stacked one above the other, as in Figure 7.2. The term *protocol stack*, reflects the predominance of this situation.

It is not uncommon for a layer to provide exactly the same type of service that it makes use of itself, and by means of identical primitives. This is not to say that the layer in question does nothing. A layer of this sort might enhance the reliability of the service, or reduce the error rate of packets, reduce the loss rate, and so on.

Example 7.1. $N+1$ **Service Protection Systems**

A simple scheme for protecting a point-to-point transmission system is as follows: Suppose the link in question carries sufficient traffic to justify installing n transmission systems each of capacity C. Instead of installing n systems, we install $n+1$ systems and a switch at either end which makes sure that if any individual system fails,

Figure 7.2: A Protocol Stack



the switch replaces this system by the standby system. The switching in this case can be very fast – in fact the main delay is likely to be due to the time it takes to be sure that a system is down.

In theory, this provides a very economical way to improve the reliability of a transmission network. However, there are a couple of issues which need to be mentioned. First, because the development of very high capacity transmission systems (using optical fibers), in the majority of cases $n = 1$, and so this system is not so economical at it appears. Secondly, this system does not protect against failure of switching systems, or multiplexing systems. Thirdly, one of the main causes of transmission system failure is physical damage to the duct, and this type of failure would take out all the transmission systems at once.

A more complex but potentially much more effective approach to protection is to establish a *network layer* which sits just above the raw transmission network and provides a way to switch to a physically separate backup path which bypasses a problem, be it damage to a duct, failure of a transmission system, or failure of a switch.

We shall return to this question in Example 9.6 in Chapter 9. □

Example 7.2. Service Protection

A *service protection network* has in recent times past been provided by major telecommunication companies.

This network makes use of raw transmission services and provides, to the layers above it, something which also looks like a raw transmission service. The difference is that when a failure occurs in one of the raw transmission services upon which the service protection network relies, control equipment within this layer detects the fault and quickly makes use of an alternative raw transmission service to maintain connectivity for the transmission service provided to the upper layer. The time taken to switch from the failed transmission service to the alternative might, in some cases, be as little a few milliseconds. Even so, some data is likely to be lost. However, this loss of a few milliseconds of data is not likely to cause a major problem for the layer above because short periods of lost data are something which any higher layer protocol is likely to be designed to handle.

Some types of failure are always likely to be beyond the capacity of a service protection to disguise, however, the probability of an unmasked failure can be reduced considerably. Of course, the service protection layer relies on the existence of *spare* (*un-utilized*) transmission capacity in the layers below. The interesting design question here becomes: how can we provide a very low probability of service interruption by means of a relatively small addition of spare transmission capacity? □

7.2 Hierarchy

The second universal principle has an even longer history than the first. From the first days of networking there has been sub-division into the local, and global, or local, trunk, international, or, when necessary, local, transit, trunk, national, international.

It doesn't require an international standards body to set up these sub-divisions – it just happens.

7.2.1 Hierarchy in Telephone Networks

The nodes of a telephone network are known as *telephone exchanges* or *end offices*. We shall use the former terminology since it clearly identifies these facilities as forming part of a telephone network. As might be expected, in a sub-division by geography, the first broad sub-division is between the local, or *access*, network and the inter-exchange network.

The Access Network

Local telephone exchanges form the hub of the *local access network*, by means of which each telephone in a home or office is connected to the telephone network. Sometimes there are active components, i.e. switches, concentrators, or *pair gain* systems, in the ducts, pits, and manholes of the access network, however it is more common for telephone access network to be formed purely of cables and passive cross-patching equipment. the primary reason for this reluctance to install more intelligent, or active, equipment in the access network is the hostility of this environment and the difficulty of maintenance of equipment stored in these locations. If it becomes possible to develop equipment which is sufficiently cheap, reliable, and robust to survive unattended for long periods of time in this environment a change could easily occur here.

A central feature of the access network is the huge once-only cost of installation. Local access networks are put in place years in advance of the time when they become moderately well utilized and continue to be used for decades. When the access network becomes *fully* utilized, which will only happen in certain locations, upgrading can be difficult and expensive. In such situations it might be economical to use *pair gain* systems, which enable m lines from the telephone exchange to some point in the access network to be transformed into $n > m$ (e.g. $n = 2 \times m$) pairs continuing along to houses and offices in the area.

Further development of the local access network is likely to involve ADSL (Asynchronous Digital Subscriber Loop) and the IP protocol as an integrated base-level protocol for data and voice services in the access network.

The Inter-exchange Network

Once a telephone signal (with or without its IP payload – or should this be the other way around?) reaches the local exchange, in many cases its journey has just begun. The Telephone network from here on forms a grand scheme of networks within networks within networks. Figure 7.3 depicts the situation for a level or two.

The hierarchy of routing in a telephone network is not as rigid as you might, at first, imagine. It is true that calls can be, and are often, routed up the hierarchy till they reach the necessary level, and then down the hierarchy level by level at the other end till the destination exchange is reached. But there are also usually routes which go from one exchange to another at the same level. The reason for this is simple enough: these *direct routes* are shorter, and therefore *may* be more cost-effective. Whether they *are* cost-effective in practise depends not only on the total cost of the path, per carried traffic, but also on the efficiency of the traffic which is carried on each link. If a link is cheap, but is only able to be occupied up to 3% occupancy, because of quality of service considerations, then it is unlikely to be cost-effective in practise. Such links are simply not installed in telephone networks.

Cable TV Networks

Cable TV (CATV) networks present an alternative architecture for access to the home. Since Cable TV was set up as a distribution network it is, effectively, *all access*. This can't strictly be true, because somehow the television signal must pass from the point where the video signal is composed to the place where it is distributed in a neighbourhood. However, this part of the distribution process can be dealt with by standard procedures – leased lines for example – and is therefore not usually discussed.

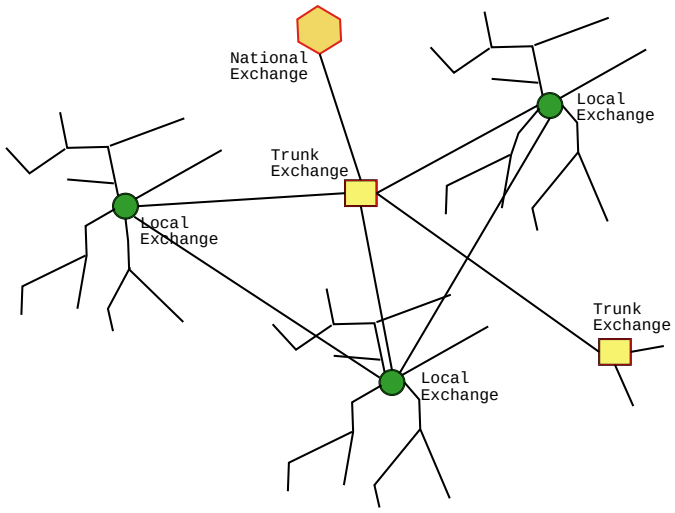


Figure 7.3: The Hierarchy of Telephone Exchanges

A traditional CATV access network is shown in Figure 7.4.

The path between the *head end* and the home contains a succession of stages of *amplification*, and splitting. Traditionally, the signal is analog in nature. A collection of analog television signals are combined together by means of frequency division multiplexing into one large analog signal which contains the entire offering of the Cable TV company.

Changes to this basic broadcast technology are driven by the following aims:

(i) Cable TV companies want to be able to provide the full range of services: telephony, Internet access and interractive services, including *video on demand*.

(ii) Communication standards and practice for cable TV are expected to migrate to digital transmission over the next 5-10 years, thereby expanding the capacity of the downstream path and introducing for the first time an upstream communication path.

(iii) It is expected that the entire CATV network should be accessible to remote monitoring and management.

These aims can be supported by means of the following technologies which are now available and can be expected to become as cheap, or cheaper, than the existing technology to install and maintain:

(a) Use of optical fiber for all or part of the access network;

(b) Digital transmission technology;

(c) Standardisation of protocols for services currently not often provided on CATV networks, such as Intertnet access.

As with every aspect of communication technology, there are competing standards under development for the new digital CATV access network [1, 2].

7.2.2 Hierarchy in the Internet

The Internet is not without hierarchy. In particular, the hardware, techniques and protocols used in the *access network*, are quite different to that used in the rest of the Internet. In fact, up to this point, the Internet does not

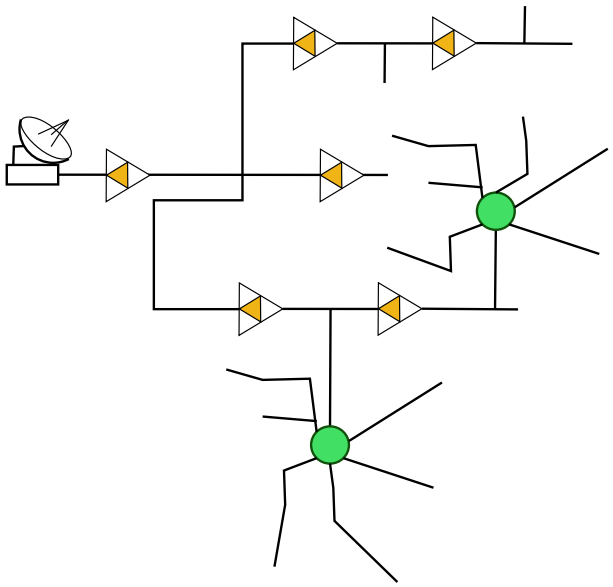


Figure 7.4: CATV Access

really have its *own* access network. Instead, largely, we use some other network to provide access to the Internet – typically the telephone network, but nowadays, with increasing frequency, a CATV network.

This could change in the future if IP is used in the access network as the substrate on which other protocols are laid to provide services such as telephony, and broadcast video.

The geographical sub-division of the Internet into *autonomous systems* also represents a hierarchical subdivision of sorts. However, this particular sub-division does not simplify routing very dramatically because it only affects communication between routers.

Some other sort of geographical subdivision of the Internet is required in to manage the complexity of routing. This second sub-division is provided by the concept of *subnetworks*. When a collection of IP addresses sharing a common prefix can be identified, routers outside the region where these IP addresses are in use can use aggregated routing information for this whole collection. One entry only is required for the entire collection of IP addresses.

The effectiveness of this strategy depends critically upon the degree to which collections of co-located hosts have been allocated IP addresses with a common prefix. A strategy for allocating IP addresses which seeks to maximise the degree to which physical proximity and IP address proximity are brought into line was introduced into the policies for assigning IP addresses some time ago, for this purpose.

Ideally, perhaps, the entire Internet could be sub-divide into a small number of regions, e.g North America, South America, Europe, Asia, South East Asia, Australasia and the Pacific, Africa, and Antarctica. Each of these could be allocated an IP address prefix. Routing from one region to another could then be handled only at the aggregate level. Within each such region, further sub-regions could be defined and routing decisions could then be aggregated again, at a lower level. Presumably this sub-division could be continued right down to the lowest level – suburbs and towns.

Of course, this is not the way Internet addressing works. Such a scheme would not be possible without rigid central control of address allocation – which does not exist in today's Internet. In fact, the suggestion that *all* traffic between two large regions could be routed in a like manner is already at variance with the highly decentralized approach to management of the Internet.

Thus, this "ideal" of hierarchical allocation of IP addresses is not sought to any significant degree. There is a degree of encouragement afforded to alignment of IP address proximity with geographical proximity. More than this is not necessary.

In the Internet, multiple strategies for managing the complexity or routing are necessary, and we discuss the other important strategy with this goal in §5.4.1. However, this goal is not given so much priority that the structure of the network as a whole is dictated by it, because that sort of approach is not possible without rigid control by a centralised authority.

7.3 Networking Philosophies and their Interaction

7.3.1 Philosophy of TCP/IP Networks

A few key ideas have been had a lot of influence in the design and development of the Internet, so much so that they deserve to be included in a putative *philosophy* of the Internet:

(i) protocols and services should be *scalable*, i.e. they should continue to function and provide satisfactory levels of service for larger and larger collections of clients, almost without limit;

(ii) protocols should preferably be *stateless* – that is to say, it is preferably not necessary for the network to store information about the two (or more) parties who are engaged in a point-to-point (or point-to-multipoint) communication;

(iii) no essential aspect of the service provided on the Internet should rely on a central control authority or server;

(iv) connection-less communication protocols are to be preferred to connection-oriented;

(v) access to service should be unrestricted, and when network resources are limited, all contending requests should have *equal and fair* access;

(vi) standards are openly discussed and defined by means of a debate in which participation is invited from the whole Internet community. Current draft standards are publicly available from a source accessible to all, in a form suitable for any reader, and at no charge;

(vii) access to the service is not restricted to certain segments of the industrial, cultural, or economic community;

(viii) charging is not built in to the architecture, but is handled, rather, as an afterthought, e.g. by charging a monthly access fee which varies depending on the access rate; charging schemes which emphasize the low marginal cost of providing additional service are preferred.

(ix) off-the-shelf hardware should be adequate to provide most functionality, with the addition of certain facilities by means of software;

(x) the basic components of the Internet should be able to recover from a failure of one node in the network and provide the best possible level of service without reconfiguration by any centralized agent, human or otherwise;

(xi) there is no need to be able to define the *owner* of *a sub-Internet*, and the boundaries between the part of the Internet owned by one provider and another; such issues can be worked out on an ad-hoc basis, as they arise.

Item (iv) is really a consequence of Item (ii), because the establishment of a connection through a connection-oriented network must store some state information about the connection. The reasoning behind these philosophical positions can be explained variously – because the work was funded by the United States Defense Department, the network which result needed to be able to recover from a nuclear blast; or, because the work was guided by computer science academics it naturally relied on a decentralized funding and control scheme. At any rate, the success of the Internet design has been remarkable by any account.

7.3.2 Philosophy of ATM Networks

(i) ATM networks should be capable of carrying *any* service (voice, video, data, control) and *guaranteeing* performance levels similar to that provided now in switched synchronous data networks, telephone networks, and packet networks. In particular, levels of loss as low as 1 packet in 10^{-10}, should be achieved on a routine basis, queueing delay should be well below propagation delay most of the time, and *delay variation* should be kept to very low levels, when required.

(ii) The architecture of ATM networks was implicitly modeled on that of telephone networks – centralized control, special equipment and protocols used by the network authority only, and the basic communication protocol on which everything else is built is connection-oriented;

(iii) small, simple packets (*cells* 53 bytes long, with 48 bytes of data and 5 bytes of header), were considered essential so that voice and associated (video) services could be accommodated efficiently without requiring excessive packetization delays;

(iv) access to services is restricted so that connections which have already been established can have their service guarantees upheld, while new requests are *denied access*, by the *connection admission control* (CAC) whenever utilization of network resources is reaching saturation;

(v) standards are defined by *standards bodies* , such as the *ITU*, which is dominated by Telecommunication bodies and does not publish its draft or final standards in a manner which is generally accessible. Standards are normally developed by a limited circle of privileged *experts* selected primarily by telecommunication companies. This approach to standardisation has been modified fairly significantly in the last ten years by the activity of the *ATM Forum*[3] which is an industry sponsored body which has sought to speed up the standardisation process for ATM and to improve the resonsiveness of the standards process to influence from industry. However, although the ATM forum has changed the approach to standardization, the centralized, bureaucratic style of ATM standardization has not been entirely eliminated;

(vi) charging for service on the basis of packets transmitted, duration of access, and rent, is provided for from the ground up;

(vii) it was assumed that specialized hardware would be required to make the service work, and that development of this hardware was the key technical development required to introduce networks capable of operating at and providing end-to-end services at *broadband* speeds (in excess of 2 Mbit/s, and potentially 100's of Mbit/s to the end user);

(viii) ATM networks shall each be *owned* by a single authority, or organization, and clearly defined boundaries shall exist between ATM networks, across which protocols for inter-network communication will need to be defined and used (although these protocols might be very similar to the internal protocols used in each separate ATM network).

7.3.3 Philosophy of SONET/SDH Networks

Nowadays it is widely accepted that the interface we expect a terminal device (a computer) to interface to is primarily the TCP/IP stack running on top of an Ethernet card. However, the path followed from this individual piece of equipment to the server, or whatever that it connects to, is nevertheless likely to pass through equipment which makes use of the ATM and SONET/SDH protocols.

The majority of optical fiber which is installed today make use of the SONET/SDH architecture. In addition, a significant proportion of these optical fibers carry at least some ATM packets. When this is the case, TCP/IP protocols effectively make use of the services provided by the ATM layer, which then, in turn, make use of the SONET/SDH layer.

It is possible to run TCP/IP directly on top of SONET/SDH, or directly on the fiber, with only a very basic framing protocol (basically `ppp`), in between the TCP/IP protocols and the hardware. Also, ATM can be run directly on an optical fiber, without the use of SONET/SDH. However, each of the protocol layers – SONET/SDH, ATM, and IP – provides certain functionality, and to emphasize efficiency to the degree that the overhead of the

SONET/SDH or ATM headers and framing is begrudged entails the risk that some of the functionality of these protocols will be missed.

In particular, each layer provides either switching or routing and the cost per switched bit rises as we go up through the layers because, in broad terms, as we go up through this sequence of protocols, the switching/routing activity becomes more complex per switched bit. For this reason, even though, in principle, it is not necessary to make use of either SONET/SDH switching or ATM switching, it may be possible to create a cheaper network by making use of two or three layers rather than just the one IP layer.

The philosophy of the SONET/SDH architecture was outlined, to an extent, in Subsection 2.3.4. Here it is explored in more detail:

(i) The SONET/SDH layer is provided bandwidth, by the layers below, and it provides bandwidth, in smaller modules, to the layers above; this bandwidth takes the form of a synchronous bit-stream, both the one below, and the ones above.

(ii) SONET/SDH systems are synchronized to the maximum degree feasible (factors such as temperature variation mean that a certain degree of asynchronism is virtually unavoidable) with reasonable effort using todays hardware, and are able to insert and drop bits from the transmission system in order to be able to maintain synchronization, when necessary.

(iii) SONET/SDH systems incorporate a full complement of transmission overheads for transmission system maintenance purposes, such as end-to-end error checking and monitoring, repeater-to-repeater error checking and monitoring, end-to-end and repeater-to-repeater voice communication links (for use by maintenance staff), and so on.

(iv) The SONET/SDH protocols are standardized sufficiently well that equipment from different manufacturers may be interchanged.

(v) The range of bit-rates at which SONET/SDH may operate is restricted to multiples of a certain specific rate (51.84 Mbit/s, known as OC1), but is not limited as to *the number of multiples* of this basic rate at which it operates, without significant embellishment of the standard. This implies that the standard anticipates transmission systems of arbitrarily large capacity.

(vi) Individual bit-streams at rates down to 64 kbit/s may easily be extracted from an SONET/SDH system. The cost of such extractions is linear in the number of bit-streams extracted. The smallest bit-stream likely to be extracted in this way is more likely to be at around the rate 2 Mbit/s.

The key limiting factor in this philosophy is the concept that a synchronous bit-stream is the basic unit of service. In today's world, a synchronous bit-stream of any rate is really a bit too inflexible for end-users. On the other hand, as a means for providing tailored synchronous services to communication providers, the SONET/SDH protocols continue to provide a valuable basic facility which would be difficult to replace. For example, if TCP/IP were carried "directly" on an optical fiber (inside a simple framing protocol based on `PPP`), the end-to-end and repeater-to-repeater transmission system maintenance systems provided by SONET/SDH would be missing. This could prove to be a problem under some circumstances.

7.3.4 Cross-fertilization of ideas

Some elements of these philosophies can claim to be firmly founded in the carefully constructed world-view of the respective participants. For example, the choice of a 48 byte payload for ATM cells was considered essential for ATM to provide the quality of service required of a truly universal multi-service network. Similarly, the adoption of stateless routing in the Internet can probably claim to be a carefully reasoned choice on the basis of deeply-held beliefs by members of the Internet community.

On the other hand, other elements of the philosophy merely reflect the fact that the cultural background of the rival groups is different. The way in which standards are developed and published is an example of this; also, the differing approaches to charging and funding. Nevertheless, these *peripheral* aspects of the debate and conflict which has accompanied these rival approaches to networking have a strong claim to be just as significant in the outcome, as we see it today.

That outcome, for those who have not already drawn the obvious conclusion, is that the Internet protocols have won the day. ATM protocols are reserved for the provision of bulk transmission facilities on networks owned by and used primarily by the telecommunication companies themselves.

However, it is by no means clear that the deeply-held philosophical differences in regard to networking have been the fundamental reason for this. Taking the philosophical positions of the two camps one-by-one, we shall see that in several cases, the supposedly fundamental differences between the two approaches have gradually dissolved – and ideas from one camp incorporated in the other networking philosophy in an ad-hoc manner.

On the other hand, the cultural differences between the two camps can be readily seen to be sufficient to easily explain the much greater acceptance and success of the Internet community:

ATM Networking Concepts adopted in the Internet

Despite the fact that stateless routing is one of the key elements in the philosophy underlying the Internet, the use of state-full routing is increasing at a pace. This arises in the situation, as it described in the Internet, of *Network Address Translation* (NAT) , also known as IP masquerading. Since this approach is used widely without apparent harm or inconvenience to those who take advantage of it, the risks of a state-full approach cannot apparently be so bad as originally feared.

A second example is the concept of reserving a path for a connection at the time when the connection is set up. This approach is seen now as essential for certain classes of service.

Internet Concepts adopted in the ATM Architecture

An important example of a concept from the Internet adopted in ATM networks is the idea that switches should collect routing information, from other switches, and dynamically determine the chosen routes. This approach is being adopted increasingly, in preference to the more traditional approach within telecommunication companies, where routing is centrally decided, by a static process.

7.3.5 Multi-Protocol Label Switching (MPLS)

A relatively new concept which attempts to take maximum advantage of any multi-layered network is *multi-protocol label switching*.

Let us assume, for simplicity, that a network has been constructed using the three layers: SONET/SDH, ATM and TCP/IP. This means that every *bit* is handled by all three protocol layers. At some nodes of this network, each bit is processed in the same way by the hardware and software of all three layers. However, there also may be some nodes at which only *some* of the layers are active. For example, an optical fiber across the Pacific must include repeaters, at which SONET/SDH hardware will monitor and manipulate the bit-stream, but the ATM and TCP/IP layers are completely absent.

The question now arises, therefore, as to how we can bypass as many layers as possible, and thereby reduce the investment which is necessary in that layer.

Note: if the equipment at a lower layer is more expensive, per bit, than equipment at a higher layer it would make sense to use a different strategy, and attempt to bypass the lower layer. In fact bypassing the lower layer might be the best strategy even if the cost of the lower layer is only a little bit cheaper per bit than the higher layer, because it can be assumed that the higher layer cannot be completely eliminated, whereas under some conditions the lower layers *can* be completely avoided.

The choice of which layers should be used and which should be bypassed is an important one for network administrators from time to time. There is no obvious *right strategy* which applies across the board, but instead it is necessary to explore the alternatives in each specific situation. A simple but useful model of the cost of a network with and without a certain layer is presented in Section 8.2. This model can be used to decide whether a certain layer should be adopted.

Further discussion of MPLS occurs in Example 8.2.

Example 7.3. IP Over IP

Let us explore what appears to be a silly idea: using an IP network as the link layer for another, different, IP network. The packets of the upper layer are *encapsulated* within packets of the lower layer. The configuration is illustrated in Figure 7.5.

In order to keep things simple, let us assume that the upper IP layer uses a totally disjoint range of addresses from the lower IP layer. This implies that the lower layer is used *only* for carrying IP packets from the upper layer and for network management purposes.

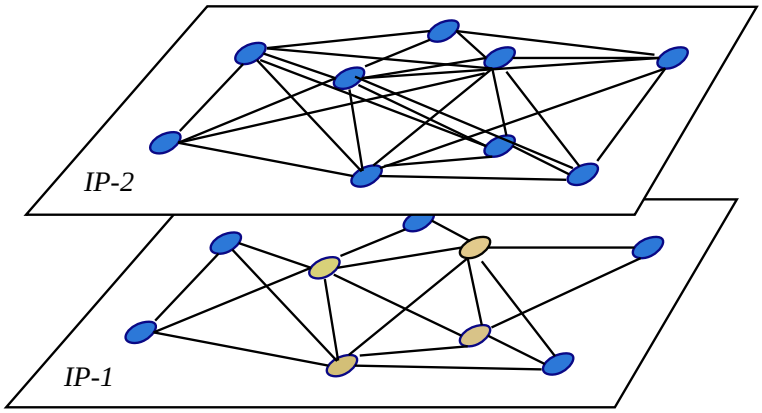


Figure 7.5: IP Over IP Layers

Why would we want to do this? The reason is simple enough: as an IP network gets larger and larger, the routing tables in the core routers become larger and larger and so, those routers become more and more difficult to manage and routing becomes slower. The division of an IP network into Autonomous Systems does not directly address this problem. The division into Autonomous Systems is designed to keep a bound on *router communication* rather than to keep the size of routing tables down[4]. Routing tables must still be formed as lists of routes, with each route referring to either a network or a host.

The two-layer IP network architecture is not useful in a small network, but in a large network it has potential advantages, which can be quantified, and as the IP network under consideration gets larger, these advantages will also grow. Conceivably one might even want to make use of more than two layers of IP in a sufficiently large network.

Let us quantify the costs and benefits of an IP over IP network. For this purpose, we can use the layered network cost model, which is explained in more detail in Section 8.2. Acording to that model, the cost of the whole network is changed by the factor

$$R_{B/A} = O_h \times (R + (L_B/L_A)(1+R) + C/C_A)$$

by the use of the extra layer, where the two IP layer network will be called B and the one IP layer network will be called A, R is the ratio of routing/switching in the lower IP layer of B to the cost of routing/switching in A, C_A is the cost of the A network as a whole, C is the one-off startup cost of having the second IP layer, L_A, L_B are the lengths of the IP routed paths in the top layer of each network, and, finally, O_h is a factor to take into account the cost of the extra protocol overhead of having packets within packets, so that the payload is a smaller proportion of the carried bits.

For simplicity, let us take $C = 0$, i.e. no startup cost for the extra layer. In this case the cost ratio formula becomes even simpler. We find

$$R_{B/A} = O_h \times \left(R \left(\frac{L_A + L_B}{L_A} \right) + \frac{L_B}{L_A} \right).$$

The value of R can become lower and lower as routing tables in network A become very large. The routing tables in the lower network will remain insignificant in size by comparison even for quite large networks, so long

as their size is considerably less than that of the upper IP layer. In the most effective case, which applies in the limit as the two networks become larger, and $L_B \ll L_A$, the cost of routing / switching in the whole network, B, approaches $O_H \times R\times$ that of network A, even though every packet must go through both layers at some nodes.

For example, suppose the lower layer can handle four times as much traffic through the same router, so $R = 0.25$, suppose the extra IP overhead adds 10% to the overhead cost and the length of paths in the B network upper layer is $1/5$ of the length in the A network. Then

$$R_{B/A} = 1.1 \times (0.25 \times 6/5 + 1/5) = 0.55.$$

So, under these circumstances, the IP network with two layers will be a little over 50% of the cost of the corresponding single layer network. □

This example shows how layered routing can be beneficial in itself, even without the use of any particular new approach to or technique for routing. However, there are inefficiencies in the IP over IP approach which are really unnecessary. We don't need an entire IP header, twice. Instead, we could abbreviate the header of one of the layers (the lower layer, and therefore the outer IP header) to just the IP address.

In addition, we could take notice of the fact that there are certain long-term *flows*, of packets, which traverse the same paths through the lower IP layer over and over again, and instead of using an end-to-end address, we could use a label for the flow. Under these circumstances, the routing process required at each incoming port of a router becomes completely independent from the routing process at another port. This simplifies routing even further because the complexity depends ultimately on the number choices which have to be made.

With these additional simplifications, and with this additional attention to efficiency, we have arrived at Multi-Protocol Label Switching (MPLS). There are other aspects of MPLS which we have not considered – for example, the idea that the label should be associated with a *Forwarding Equivalence Class*, i.e. an *anywhere-to-somewhere flow*, rather than an end-to-end flow. This is a very useful idea because, by the 80-20 rule (or whatever variation of it that applies to Internet traffic), 80% of traffic is destined for 20% of destinations. What the actual statistics are here is unclear – perhaps 98% of traffic is destined for 2% of destinations. In any case, by aggregating traffic addressed to each of these very intense destinations and handling it more efficiently, we are obviously going to improve the efficiency of routing significantly.

With all this talk of routing efficiency it is perhaps appropriate to reflect on how important routing efficiency really is. Traffic volume in the Internet is growing rapidly. This requires more and more resources all the time. It is not sufficient to merely increase the production of routers because we *already expect* increases in efficiency. The expectation of increases in efficiency of all aspects of communication technology is factored into the way our economy functions. A communication service company which is not becoming more efficient all the time will not be able to compete, and the same goes for the equipment of which networks are made.

Networks are made of switches/routers on the one hand and transmission systems on the other hand – and perhaps some higher level functionality, like DNS servers, network management facilities, and so on. If one aspect of network technology fails to increase in efficiency, in time, this aspect will become the limiting factor. Transmission technology has provided steadily more and more efficient and cost-effective point-to-point communication services. These improvements can be expected to continue. As a consequence, the focus must turn, increasingly, upon the switching/routing functionality of the Internet. Routers must also become more cost-effective.

7.4 Security Architecture

As with routing, switching, and transport, in the area of security, there are competing architectures. The importance of security in the scheme of things is growing steadily.

Security is not an area of network design, analysis, or management with a long history and a well-developed theory. Although the subject of encryption and decoding has a long history, the momentum of research and development in this area really built up during the second world war. At that time, decryption of encoded messages of the enemy was a crucial war strategy. All the major participants in the war encrypted messages as a matter of course and also put large amounts of effort into decrypting the messaqes of their enemies, usually with considerable success. It seems that all sides tended to overestimate the quality of their own codes and to underestimate the ease with which they could be broken.

The most famous development in this area since the war has been the discovery of public key encryption. By means of this technique it is possible to send an encrypted message to someone you don't know, and have never met before, over a public network in such a way that only the intended recipient is able to read the message.

7.4.1 Key Concepts in Security

The key issues in security, and their apparant solutions [in brackets, like this], are commonly believed to be the following:

(a) protection of identity [authentication],

(b) protection of privacy [encryption], and

(c) protection of resources [filtering].

In reality, there is more to security than these three issues and it would be naive to beleive that the techniques just mentioned are actually a satisfactory complete solution of these problems either. Three security issues which immediately come to mind and which clearly go beyond the scope of the "big three" issues just mentioned are: *non-repudiation*, i.e. the facility to ensure that once a document has been acknowledged as sourced from a certain location and has been signed to acknowledge this fact, it cannot later be denied that this is the case (repudiated); *replay protection*, i.e. protection against re-use of data which has been collected from the Internet for its original purpose by a party other than the one authorized to do so; and *Protection of the valid use policy*, i.e. mechanisms to ensure that authenticated users do not go beyond the agreed valid use policy which applies to services that they use.

However, good strategies for authentication, encryption, and filtering do represent the bulk of current good practise in this area.

Let us consider the "big three" issues one-by-one, and then return to consideration of the other ways of looking at security to see if there are some important matters we have overlooked.

Protection of Identity

Given that communication networks are now often used for important commercial activities, there are many situations when it is essential that the identity of the actors in a transaction is verified and any uncertainty as to this identity reduced to a minimum.

The name we give for the mechanism which certifies identity is *authentication*. The traditional method for authentication is the use of a username and a password.

Usernames and passwords as an authentication mechanism leave a lot to be desired. Passwords can be guessed, discovered, and distribution of usernames and passwords is difficult because it almost inevitably exposes them to these risks. Also, if used in an unsophisticated manner each use of a password exposes it to the risk of interception.

A better mechanisms is available. This better mechanism is based on the technique of public key encryption, described in a little more detail in Section 7.4.2. Public key encryption is too complicated to be usable without some fairly sophisticated packaging. The most common packaging, which is used widely throughout the Internet, is the use of *certificates*. A certificate is actually a public key, to which a private key is associated, and which has been certified by an independent authority. The most widely used standard for digital certificates in the Internet is X.509 [5].

Let us defer further explanation of how certificates are used in authentication till we have discussed public keys.

Protection of Privacy

Protection of privacy of data usually relies on the requirement for authentication prior to any form of access. A more interesting issue is how to protect the privacy of communication through a network in which interception (eavesdropping) cannot be prevented with certainty. The solution to this problem is to use *encryption* of the data flowing in each direction.

In order to use traditional encryption, the two parties communicating must both know a common key and this key should not be known to any other party. If the parties have not communicated prior to this time, they will need to communicate the common key before they can communicate in private. But transferring this key from one party to the other appears to present the same problem again! For if the key is sent on the public network by which the communication is to take place, it will be possible to intercept this message and invalidate the security of all subsequent interchange.

There is a solution to this problem – more than one, in fact. Again, a solution is provided by the concept of public and private keys. Alternatively, there are methods for distribution of keys by means of which their privacy can be guaranteed even when all communication takes place in public.

Protection of Resources

The last category of security can refer to a great variety of security problems. The resource in question might be files on a networked host – but we have dealt with this issue under the heading of privacy already. But consider the case where the resource to be protected is a public server. The legitimate users of this resource make use of it by sending packets to it, and engaging subsequently in a communication. The other users, the ones who are not "legitimate", are distinguished merely by their intentions. The mode of interraction of the non-legitimate user is almost indistinguishable from that of the legitimate user. The malcontent or mischevious misuser can probably be distinguished from the legitimate users by the fact that their requests come at a much faster rate than ordinary users. If this wasn't the case, these "non-legitimate" users would not actually present too much of a problem.

7.4.2 Public key encryption

Now we come to the key technical ideas in the area of security: *Public key encryption*, and *public key distribution*.

The concept of publicy-private key encryption was announced by Diffie and Hellman in [6]:

> In public key cryptosystem enciphering and deciphering are governed by distinct keys, E and D, such that computing D from E is computationally infeasible (e.g., requiring 10^{100} instructions). The enciphering key E can thus be publicly disclosed without compromising the deciphering key D. Each user of the network can, therefore, place his enciphering key in a public directory. This enables any user of the system to send a message to any other user and enciphered in such a way that only the intended receiver is able to decipher it. As such, a public key cryptosystem is a multiple access cipher. A private conversation can therefore be held between any two individuals regardless of whether they have ever communicated before.

A satisfactory implementation of public-private key encryption was not provided in the paper of Diffie and Hellman, but was provided soon afterwards by Rivest, Shamir and Adleman [7]. The technique proposed by Rivest, Shamir and Adleman can be described quite briefly and is not difficult to understand for mathematicians with the appropriate background. For the record, here is the description provided in [7]:

> A message is encrypted by representing it as an integer, M, raising M to a publicly specified power, e, and then taking the remainder when the result is divided by the publicly specified product, n, of two large secret prime numbers, p and q. Decryption is similar; only a different, secret, power d is used, where $e.d \equiv 1(\mod (p-1)(q-1))$. The security of the system rests in part on the difficulty of factoring the published divisor, n.

In the same paper by Diffie and Hellman, [6], already mentioned above, another important technique for enabling private communication to take place over a public network was also described. This technique is known as *public key distribution*. In this technique, two parties are able to engage in a public conversation in order to define a secret method for encryption and decryption which they can then use to engage in a private conversation. The method they proposed in [6] is quite simple (simpler than the RSA method) and can also be described unambiguously in one paragraph (taken from [6]):

> Suppose that A and B are two individuals (or agents) wishing to choose a common, private, symmetric key with which to encrypt their subsequent communication. They should proceed as follows:

A chooses an integer *a* and *B* chooses an integer *b*. *A* now forms the integer $\alpha = g^a(\mod p)$ where *p* is a large prime selected by one or the other party and *g* is an integer similarly known to both. *A* now sends α to *B*. *B* likewise forms the integer $\beta = g^b(\mod p)$ and sends it to *A*. Now *A* determines the common key as $Z = \beta^a(\mod p)$ while *B* finds the same integer as $Z = \alpha^b$.

Digests

One more fundamental idea from the technical world of crytography must be understood before we can discuss how public key encryption is applied. this is the concept of a *digest* [8].

A digest is a brief summary of a large document. The precise form that the summary takes depends only upon the document. This brief summary is not intended to be *readable* – not in a meaningful way anyway. The MD5 digest algorithm, for example, produces a "summary" of 128 bits in length. However, the digest does have the following characteristics:

(i) it is computationally infeasible to produce exactly the same digest from a different message;

(ii) it is quite straightforward to reproduce the digest exactly given knowledge of the document and the algorithm used to create the digest, which is usually publicly available.

Digital Signatures

A digital signature of a document is a special type of digest. The signature is formed by creating the digest and then encrypting the digest with the private key of a public-private key pair.

This digital signature cannot be reproduced by any party without a knowledge of the *private* key. However, it is straightforward to *verify* the digital signature by re-creating the digest from the original message, decrypting the digital signature with the public key, and comparing the two. If there is any difference, the digital signature is invalid.

Let us now revisit the three "Big Issues" of security and see how public key encryption and key-exchange methods can be used to provide appropriate security mechanisms.

Protection of Identity

Public keys can be readily used as proof of identity. It works as in the following example. It may seem as if the use of a digital signature verifies the identity of the signing party immediately, however there is a little more to it than that.

Example 7.4. Digital Signatures and Certificates

Let us assume that party *A*, the Electric Toast Company, for example, is sending a document to party *B*, the Wet Blanket Fire Protection Authority, and wishes to prove that they really are who they say they are. Furthermore, they have a *certificate*, issued by the well-known certificate authority, Certisign, together with the private key which matches that certificate. The certificate is basically the public key corresponding to that private key together with some public information about that key, all of which is *digitally signed* by a certificate authority, in this case *Certisign*.

So, what does the Electric Toast Company do with the document, to ensure that the Wet Blanket Fire Protection Authority can be 100% confident that the message was sent by them? Naturally enough, they append a digital signature to the message. This protects the message against modification by any party along the path the message traverses between sender and receiver and it also confirms that the sender holds a certain private key, namely the one certified by Certisign. Next, the Electric Toast Company adds the certificate which certifies their public identity. This does the trick.

So, what does the Wet Blanket Fire Protection Authority do with the document, the signature, and the certificate? First, it can verify that the signature was computed using the private key corresponding to the public key in the certificate purportedly held by the Electric Toast Company. This doesn't confirm the identity of the sender, so much as verify that the signature was formed by the party who holds that certificate.

The certificate does contain a name – the name of the Electric Toast Company. However, another step is required to gain confidence that the real Electric Toast Company does own this certificate. We need to verify that

this certificate is genuine. This is where the signature on the certificate comes in. The certificate has been signed by Certisign. Certisign's public key is well known (and can be looked up readily). So we can use this to verify that the signature is genuine. This then confirms that, as far as Certisign is concerned, the certificate is genuine. So, assuming that we trust Certisign, we can now be completely confident that the sender of the message really is the Electric Toast Company. □

Protection of Privacy

Encryption is the obvious method to use to protect privacy. Encryption methods are well known and widely available The primary difficulty is establishing a common key that both parties can use in the encrypted exchange. Two of the ideas discussed above can be used to find a common key. The Diffie-Hellman key exchange method is an obvious choice. Algorithms based on this method have been standardised for use in the Internet [9].

Another approach is to use a pair of public and private keys to exchange a common once-only session key. This is the approach used in PGP [10], for example, and also in SSL [11].

Example 7.5. Denial of Service Attacks

A denial of service attack [12] is an attack in which a server is flooded with many copies of the opening packet of a TCP connection, a SYN packet. These SYN packets do not have the IP address of the sender in the appropriate place in the header. As a consequence, no successful connection is ever set up as a consequence of one of these attacks. However, processing these opening SYN packets does take time and if sufficiently many are sent, the server can be prevented from handling the load of genuine requests for service that is being received at the same time. □

Protection of Resources

Neither public key encryption nor public key exchange methods appear to have any special relevance to the issue of protection of resources. However, if the problem of denial-of-service attacks, for example, becomes more serious, it may become essential to incorporate authentication as a feature of more and more services on the Internet. In an extreme case we could insist that *all* requests arriving at a server must be authenticated, e.g. using the IP Authentication Header [13]. Packets which do not include authentication could be filtered out entirely. This would solve the problem, because it is unlikely that an attacker would be willing to identify themselves in every packet that they sent to the server they are attacking.

Example 7.6. SPAM

SPAM, or email which is broadcast indiscriminantly to valid email addresses which have been trawled from the Internet, is a serious and growing problem [14]. There are approaches by means of which *some* SPAM can be filtered out by ISPs or email gateways. However, it is very difficult for such filters to identify all the SPAM, only the SPAM, and nothing but the SPAM.

SPAM is a good example of one of the "lesser three" security problems – namely, contravention of the valid uses policies of the Internet, and the organisations involved in its transport.

The SPAM problem can be addressed in a variety of ways: by the receiver, by the Mail Transport Agent (MTA), by attempting to stop it at the source, and by legal restraints. Of these, all are useful, except perhaps the last.

Until quite recently many SMTP servers on the Internet would forward any message on to wherever it indicated it wanted to go. Nowadays most SMTP servers insist on validation of the requestor before a message will be forwarded. This strategy cuts down on the range of possible sources.

Filtering can be applied at both the MTA and at the ultimate destination. In both cases, there is a need for good filtering criteria. A good starting point is to filter our messages that do not come from a valid DNS domain. As for the messages which do have valid sources, since these sources at least appear to be valid, these can be checked against a list of known SPAM sources and filtered out on that basis. Lists of SPAM sources can be obtained from a variety of sources on the Internet, e.g. [15]. □

7.4.3 Kerberos

Kerberos is a system for distribution of network access *tickets* which does not use public key encryption technology [16]. Instead of using public key encryption, the Kerberos system uses a trusted third party to authenticate users.

The central server maintains a record of the common keys of all the hosts in the network. These common keys are traditional symmetric keys. They need to be shared so that the central Kerberos authentication server and the individual hosts requesting authentication can communicate in a private (encrypted) manner. Using this common key, assuming all goes well, a session key is generated and included in the ticket which is sent to the client. Once the client has received its ticket, it is passed on to the server the client is attempting to reach, for authentication. The ticket contains the session key which enables the server to communicate henceforth with the client using the session key.

7.4.4 Lightweight Directory Access Protocol (LDAP)

LDAP is a development from, or perhaps better said a reduction from, the X.500 directory services protocol defined by the International Standards Association [17]. The LDAP protocol, which runs over TCP, has been implemented in publicly available LDAP servers. An LDAP server provides *directory services*

An LDAP server is a convenient location to store information about individuals, e.g. lists of staff of an organisation, or students studying at an institution. In particular, certificates assigned to these individuals can readily be stored in such a server and thereby accessed by other parties who need to make use of the public keys contained in the certificates for secure communication.

7.4.5 IPSEC

The IPSEC security architecture is described in a suite of RFCs, in particular [18, 13, 19]. These protocols can be used to provide the following functionality:

(i) authenticated and integrity-protected point-to-point communication over the Internet between one host and another;

(ii) encrypted, integrity-protected, and authenticated point-to-point communication over the Internet from one host to another;

(iii) integrity-protected and encrypted communication through an IP tunnel;

(iv) authenticated, integrity-protected and encrypted communication through an IP tunnel;

The term *integrity-protected* here is meant to indicate that any alteration of the communicated messages will be detected by the authentication protocol.

7.4.6 SSL

The secure socket layer sits above the TCP protocol and below application level protocols such as http, ldap, ftp, and so on. A more recent standardised version of SSL is known, instead, as TLS [20]. The TLS/SSL protocol is used between browsers and secure web servers when authentication and/or encryption is needed.

The TLS/SSL protocol can be configured to use a considerable variety of different algorithms for authentication and encryption, including those based on the RSA public key encryption technique.

The protocol can be further sub-divided into a handshake protocol, which is used to authenticate the server, establish parameters which will be used during the connection, optionally authenticate the client, and so on. This handshake protocol, and the subsequent transmission, make use of a TLS/SSL record protocol, which governs the exchange of records during the entire transaction.

7.4.7 Pretty Good Privacy (PGP)

A public domain architecture and collection of algorithms for encryption and authentication based on the public key encryption idea has been defined and developed, primarily by Phil Zimmerman [10, 21, 22].

PGP makes use of the RSA technique of public-key encryption. This is used to provide the facilities of: digital signature, encryption, and certification of public keys (certificates).

PGP is configured to work conveniently with email, to provide digital signatures, encryption, and transfer of certified public keys.

Exercise 7.1. Use PGP for Email

Download software for your preferred computer operating system from the Internet (eg, [22]), install it on your computer, and use it to send a digitally signed message to a friend or associate. □

7.4.8 Secure Shell (SSH)

Another useful product based on public-key encryption is the *secure shell*. Ssh provides a service analogous to `telnet` except that public-key style authentication and encryption are available. The authentication facility can be configured to operate in a very convenient manner, by means of an agent on the client machine. If the user provides ssh access to the public *and* private key on the client, and the public key is stored in a standard place on the server, the software allows `ssh` connections to be set up without any further authentication required.

In addition, `ssh` has all sorts of very useful functionality by means of which the `ssh` connection can be used to transport other services, such as the X protocol. Ssh can also be used to create secure tunnels through a TCP/IP network.

Exercise 7.2. A Secure Tunnel

Use `ssh` to create a tunnel from one host to another across the Internet. There is a trick to how this is achieved, which is that once the `ssh` connection has been established, it is necessary to run the `ppp` protocol over it. This may seem a little odd, however, given that an `ssh` connection is a continuous byte-stream, it should be clear that the IP protocol will need some sort of *framing* by means of which to slot into the `ssh` tunnel. In addition, `ppp` provides the necessary routing, by means of which packets at the client end of this connection can be advised to route through the tunnel (in some cases).

The Linux Documentation Project [23] includes a HOWTO which gives fairly detailed instructions for how to set up a VPN between two Linux hosts. The path between these two hosts can be configured to encrypt all communication. Also, the hosts at each end can be configured to route all communication to hosts on the network to which the host at the other end of the tunnel to make use of the tunnel.

This exercise is a little tricky, and relies, in addition, on having access to a remote host prepared to accomodate some guests. □

7.4.9 Key Distribution and Certificate Services

The technology and key ideas and concepts of public key encryption have been described. But there is one more important idea which needs to be introduced. The related concepts of key distribution, certificate authority and the certification services provided by a certification authority (Public Key Infrastructure).

Key distribution is the name given to the process which allows secret keys to be distributed across a public network while minimising the possibility of interception to an acceptably low probability. In principle, this can be achieved by means of the Diffie-Hellman algorithm for private communication over a public network. Alternatively, public key encryption methods can be used to encrypt a dialog in which keys are interchanged.

In practice, in order for this to be practical the precise procedures need to be carefully defined, so that all parties wishing to engage in this type of intercourse may do so effectively. There are different approaches to key distribution in the various standards. In IPSec, key distribution is achieved by means of the Internet Standards known as Internet Security Association Key Management Protocol (ISAKMP) [24, 25].

Another, related, issue, that of the certification of certificates. We have already seen how certificates are certified – by means of a digital signature from a *Certification Authority* (CA) – but this begs the following questions:

(i) How does the Certificate Authority gain sufficient confidence in the party whose certificate they are signing that they are willing to append their signature?

(ii) How do we gain sufficient confidence in the Certificate Authority that we take their signature for a sufficient guarantee of the facts being asserted?

These considerations impose considerable constraints upon the certification process. If a certification authority signs any document supplied to it without discrimination, in time, it will be clear that the signature does not guarantee anything. So, a certificate authority must attempt to *check its facts before signing anything*. Typically, a certificate authority must obtain a number of independent confirmations of the fact that is being asserted (typically, an identity) before being willing to append the signature.

Given this constraint, that the certificate authority behaves in a responsible manner, the second requirement can be met in a rather informal manner – we expect the certificate authority we deal with to be large, well-known, and to exhibit plenty of public presence. In other words, authority in the informal sense of authority in public life is the criterion that we use to invest a certificate authority with authority in the technical sense.

Example 7.7. Authority in PGP

Authority in PGP is modelled exactly on the description in the previous paragraph. In the PGP approach anyone can issue a certificate and when they do so they would normally sign it themselves. This may seem a bit odd: Suppose someone came to you with a note which read:

> This person is Ron Addie.
> Signed: *R. G. Addie*

the signature would not seem to add much to the document! However, in the case of a digital signature, it does, because it does more than simply certify that the signer *believes* the message – it also confirms that the message has not been altered since it was signed.

Nevertheless, this is not sufficient to convince another party of the validity of this certificate. So, in the PGP approach, we seek signatures from other parties for our certificate. If we can find 10 well-known individuals or parties to append *their* signature to the document, it should carry some significant weight. How much weight depends upon whether the party receiving the certificate knows any of the signing parties, however the basic idea is clear enough – authority is gained by establishing a network of supporting evidence. □

Example 7.8. Secure DNS

An obvious candidate for a system for conveying authoritative information about identity is the Domain Name System (DNS), which forms part of the Internet. The DNS system is already hierarchical, with a small collection of *root servers* at the top, with a heavily branched tree of child DNS servers emerging therefrom. It would be natural to include certification of identity in this system, and thereby every public node in the Internet could readily receive certification of identity from their DNS server.

In addition, in this way, the DNS service itself would be rendered *authoritative* to a degree which, although it doesn't seem to be essential at the moment, may become so in time, as we rely on the Internet more and more for vital services in daily life.

A secure version of the DNS system has been defined in [26, 27, 28, 29, 30, 31, 32] (amongst other documents). □

The Secure DNS system defined in these documents does *not* provide the key functions of a Certificate Authority Server or Hierarchy however.

Certification Services (Public Key Infrastructure – PKI)

There is provision for the Internet to provide a Certificate Authority Hierarchy as defined in [33] and [5]. According to these standards, the IETF authorizes a root certificate authority, known as the Internet Policy Registration Authority (IPRA), to provide certificates to other certification servers known as Policy Certification Authorities (PCAs), which in turn certify Certification Authorities (CAs). These CAs will, in principle, provide certification services to the mass of Internet servers and hosts.

Here is an extract from [5] which defines the services one might expect from a *certification authority* :

Management protocols are required to support on-line interactions between PKI user and management entities. For example, a management protocol might be used between a CA and a client system with which a key pair is associated, or between two CAs which cross-certify each other. The set of functions which potentially need to be supported by management protocols include:

(a) registration: This is the process whereby a user first makes itself known to a CA (directly, or through an RA), prior to that CA issuing a certificate or certificates for that user.

(b) initialization: Before a client system can operate securely it is necessary to install key materials which have the appropriate relationship with keys stored elsewhere in the infrastructure. For example, the client needs to be securely initialized with the public key and other assured information of the trusted CA(s), to be used in validating certificate paths. Furthermore, a client typically needs to be initialized with its own key pair(s).

(c) certification: This is the process in which a CA issues a certificate for a user's public key, and returns that certificate to the user's client system and/or posts that certificate in a repository.

(d) key pair recovery: As an option, user client key materials (e.g., a user's private key used for encryption purposes) may be backed up by a CA or a key backup system. If a user needs to recover these backed up key materials (e.g., as a result of a forgotten password or a lost key chain file), an on-line protocol exchange may be needed to support such recovery.

(e) key pair update: All key pairs need to be updated regularly, i.e., replaced with a new key pair, and new certificates issued.

(f) revocation request: An authorized person advises a CA of an abnormal situation requiring certificate revocation.

(g) cross-certification: Two CAs exchange information used in establishing a cross-certificate. A cross-certificate is a certificate issued by one CA to another CA which contains a CA signature key used for issuing certificates.

The system which allows the IETF, through IPRA, to provide certification services to the world at large does not seem to have achieved a great deal of penetration.

There is a constant day-to-day need for and use of certificates and certification services because of *secure web servers*, which are web servers with appropriate security features, which are used regularly on e-commerce sites. Such servers need to make use of secure transactions both when they communicate with their clients, by means of web browsers, and when they communicate with their suppliers, or suppliers of services, for example, banks.

A typical example of such an instance would be the situation where a client wishes to submit their credit card details so that they can pay for goods purchased on the site. In this case, the client will want and expect the transfer of information between the browser and the web server to be authenticated and encrypted. This is within the capability of virtually all browsers and many web servers. In order to do so, the web server must have a *certificate* from an appropriate certificate authority. The choice of which certificate authorities are appropriate is largely left to the browser, or the developer of the browser, i.e., most often, Netscape or Microsoft.

The need for the certificate arises when the browser attempts to check the identity of the server. For this purpose, the server supplies a certificate and uses a digital signature to show that it has the private key corresponding to the public key in the certificate.

The protocol used for authentication of the server (and, optionally, of the client) and also for encryption of the end-to-end communication between the client and the server will be either SSL or its successor, TLS [20].

Secure servers also need to communicate with their suppliers, e.g. banks, and for this purpose a certificate will also be required, although the certifying authority in this case is likely to be one selected by the bank.

Example 7.9. Verisign

The certificate authority with most presence in the Internet today is *Verisign* [34]. Verisign supplies most, if not all, of the services listed above. The degree of checking undertaken by Verisign at the time when a certificate, and the associated private key is issued depends on the type of certificate being issued. □

7.4.10 Security of Action

By security of action is meant the following:

> Protection against the actions of any user who fails to use an Internet service in the manner nominated in the rules of behaviour for this service.

This is a very broad concept, and yet it can be made quite specific. the concept of *security of action* can be adapted, if desired, to include all the other security concept already considered. On the other hand, there are quite a few security issues which do not fall under one of the "Big Three" headings which are naturally included under this heading.

For example, the sending of SPAM goes outside the guidelines for appropriate use which are envisioned in the use of email on the Internet.

7.5 Network Management

Another area of network technology and administration which bears the name of an "architecture" is *network management*. As in the areas of routing and security, there is a different approach to network management in the Internet vs the approach taken in public Telecommunication networks.

But what is network management? And why does it need an architecture?

Here is a quote from [35]:

> A large network cannot be put together and managed by human effort alone. The complexity of such a system dictates the use of automated network management tools. The urgency of the need for such tools – and the difficulty in supplying them – is increased if the network includes equipment from multiple vendors.

Stallings further breaks down network management functions as follows:

(i) Fault Management

(ii) Accounting Management

(iii) Configuration Name Management

(iv) Performance Management

(v) Security Management

Basically, network management is concerned with monitoring and control of equipment which makes up networks. For this purpose, a communication network is required – a network for communicating the network management information and control signals – how else will the information and control signals pass between the equipment being controlled and the equipment doing the controlling? Since the object under management is already a network, however, why can't we just use this network?

This is certainly the approach used in the Internet, with SNMP. However, at the time when the *Telecommunication Management Network (TMN)* was being defined, by the International Telecommunication Union, the assumption that a universally accepted transport medium was already available was not apparant.

The main complexity which invests network management protocols, standards, and software is the considerable bulk of objects, parameters, statistics, control functions, actions, and types of data which need to be referenced. There is a *terminology explosion* to be tackled.

Quite a bit of terminology has been generated specifically for network management purposes, which, to a degree, just adds to the problem. In the context of network management, we refer to *agents* which are the entities

making up the active components of the network management network, to a *Management Information Base (MIB)*, which is a set of structured variables, called *objects*, each associated with a *network element* and managed by an *agent*. A MIB can also refer to a collection of *associated objects* related to a *collection* of resources each of which is a part of a *collection* of network elements.

Each object referred to in a MIB needs an *identifier* by means of which it may be referenced in the information messages and control messages of the network management system.

This terminology seems to be reasonably consistent across the various different approaches to network management, of which there are two noteworthy contributors:

Simple Network Management Protocol (SNMP)

The Simple Network Management Protocol (SNMP) was introduced into the Internet circa 1989. It has since evolved to SNMPv3, which is described in [36, 37] as well as some 100 or more other RFC's [38]. Many of these documents provide details of one or other MIB, so the plenitude of documents should not perhaps be seen as quite so daunting.

Telecommunication Management Network (TMN)

The Telecommunication Management Network has a slightly older history than SNMP. The importance of network management in telecommunication networks was recognised quite some time ago. A great deal of cross-fertilization appears to have occurred between the TMN and SNMP, although there are very few references from one set of standards to the other.

The fact which now dominates the growth and further development of the TMN is the fact that development of communication equipment is increasingly dominated by the Internet Protocol and the family of protocols with which it is associated. Staff in Telecommunications research and development organisations are being shifted from projects which follow in telecommunication traditions to projects with closer ties to IP. The energy, the time and money, has moved from the TMN to SNMP.

This is illustrated by the fact that within the catalog of recent standards on the TMN, many were drafted in the late 1990's, whereas the collection of current draft standards on SNMP includes many documents written within the last 6 months.

This same observation would seem to apply to the *Telecommunication Information Network Architecture (TINA)* [39], which is a collection of standards, practices, and software for access to and control of information in and through telecommunication networks. Here is a quote from the TINA-C Web page:

> The architecture is based on four principles.
>
> (i) Object-oriented analysis and design,
>
> (ii) distribution,
>
> (iii) decoupling of software components,
>
> (iv) separation of concern.
>
> The purpose of these principles is to insure interoperability, portability and reusability of software components and independence from specific technologies, and to share the burden of creating and managing a complex system among different business stakeholders, such as consumers, service providers, and connectivity providers.

However, it appears that further progress on this *Information Network Architecture* appears to be rather slow.

7.6 Examples

Example 7.10. Security in a Campus Network

It makes good sense for any large organisation to consider whether it should set up its own certificate authority. Whether the services which could be offered by this means are sufficiently attractive or useful to justify the cost is unclear, however.

At present there is no pressing need for individuals to hold a certificate. Private or authenticated e-mail is not usually considered such a necessity that the trouble of obtaining a certificate is warranted. However, this could change if a service which requires a certificate is defined which is sufficiently exciting and attractive to generate a mass market. Or, if the SPAM problem worsens, it might become common for people to make use of email filters which reject all email without a digital signature. If this happens use of digital signatures on email will become virtually mandatory.

A University, or a College, or indeed any institution with a large number of staff or clients, is a good candidate for the use of Public Key Infrastructure to support normal activities. Possible uses of personal certificates in a campus setting include:

(i) Access to facilities. At present, on most campuses, passwords are used for this purpose. Management of passwords is time consuming and involves privacy risks. Public key encryption technology is not necessarily the key to addressing these problems though.

(ii) Access to academic records. Academic records are, in some respects, just another case of the previous item. However, there are special problems associated with academic records. In particular, the student may wish to pass a certified academic record to a third party. An obvious mechanism for achieving this is for the academic record to be digitally signed by the university. Protocols for requesting this service, and for providing it, have yet to be defined. This may be another application of PKI to the campus environment.

□

Example 7.11. Layers in a National Carrier Network

A public telecommunications carrier will need to maintain at least the following layers:

1. physical networks: inter-nodal and access networks,
2. SONET/SDH network,
3. one or both of: circuit switched networks (e.g. telephony, ISDN) and packet switched networks (e.g. ATM, frame relay, IP); in future a telecommunications carrier might choose not to provide any circuit-switched services.
4. IP, TCP and services provided over TCP/IP.
5. voice over IP;
6. management and control networks.

As indicated already, telecommunication carrier's can now consider the possibility to carry *all* higher-level services over an underlying packet layer. The protocols used in this packet layer could be ATM, TCP/IP, or TCP/IP *over* ATM.

The choice of skipping the ATM layer altogether and just providing a TCP/IP layer is becoming increasingly attractive. In some venues, the discussion is not so much about whether to take this step as about *when* to take it, and *how* to take it.

The IP-only version of MPLS holds the potential to provide any switching efficiencies that previously were only available with ATM. There are very few services which need the ATM infrastructure as such and very few native ATM services, so as soon as high-speed switching infrastructure can be readily provided by MPLS switches, the role of ATM will disappear.

Standardisation of MPLS routers to a degree which enables a telecommunication companies to obtain MPLS equipment from a variety of sources is another condition which is needed to justify the switch to an IP/MPLS architecture as the integrated packet sub-layer for all higher level services. □

Exercise 7.3. A Layer to Improve Performance

Consider the following question: is it possible to insert an additional layer for the purpose of improving performance of a network? This question has already been considered in some detail in connection with the *reliability* of networks. What about the *loss* performance? What about *delay*? What about *security*?

Your answer should take the form of a short essay in which you answer this question, either in the affirmative (the performance *can* be improved by a special additional layer), or the negative (no such layer exists). In the former case, please give an example, and in the latter, please explain why such a layer cannot exist. Answer the question once for each type of performance: loss, delay, and security. □

References

[1] Saleem N. Bhatti and Graham Knight. On management of catv full service networks: A european perspective. *IEEE Network*, 1998.

[2] Richard Rabbat and Kai-Yeung Siu. Qos support for integrated services over catv. *IEEE Communications Magazine*, 37(1), 1999.

[3] The atm forum.

[4] K. Varadhan. Bgp ospf interaction. Technical Report RFC 1403, IETF, 1993.

[5] R. Housley, W. Ford, W. Polk, and D. Solo. Internet x.509 public key infrastructure certificate and crl profile. Technical Report RFC2459, IETF, 1999.

[6] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22, 1976.

[7] R.L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21:120–126, 1978.

[8] R. Rivest. The md5 message-digest algorithm. Technical Report RFC1321, IETF, 1992.

[9] E. Rescoria. Diffie-hellman key agreement method. Technical Report RFC 2631, IETF, 1999.

[10] Simson Garfinkel. *PGP: Pretty Good Privacy*. O'Reilly & Assoicates, Inc., 1995.

[11] Alan O. Freier, Philip Karlton, and Paul C. Kocher. The ssl protocol, version 3.0. Technical report, IETF, 1996.

[12] Christoph L. Schuba, Ivan V. Krsul, Markus G. Kuhn, Eugene H. Spafford, Aurobindo Sundaram, and Diego Zamboni. Analysis of a denial of service attack on tcp. *Proceedings of the IEEE*, 1997.

[13] S. Kent and R. Atkinson. IP authentication header. RFC 2402, Internet Engineering Task Force, November 1998.

[14] Alan Schwartz and Simson Garfinkel. *Stopping SPAM*. O'Reilly and Associates, 1998.

[15] MAPS. Maps realtime blackhole list. Technical report, MAPS, 2001. http://www.mail-abuse.org/rbl/.

[16] John Kohl and B. Clifford Neuman. The kerberos network authentication service. Technical Report RFC-1510, 1993.

[17] W. Yeong, T. Howes, and S. Kille. Lightweight directory access protocol. Technical report, IETF, 1995.

[18] S. Kent and R. Atkinson. Security architecture for the internet protocol. RFC 2401, Internet Engineering Task Force, November 1998.

[19] S. Kent and R. Atkinson. IP encapsulating security payload. RFC 2406, Internet Engineering Task Force, November 1998.

[20] T. Dierks and C. Allen. The TLS protocol version 1.0. Technical report, (IETF.

[21] William Stallings. *Protect Your Privacy: A Guide for PGP Users*. Prentice Hall PTR, 1995.

[22] PGP International. The international PGP home page. Technical report, PGPi, 2002.

[23] Linux Documentation Project. Linux documentation project. http://linuxdoc.org.

[24] D. Maughan, M. Schertler, M. Schneider, and J. Turner. Internet security association and key management protocol (isakmp). Technical Report RFC-2408, 1998.

[25] D. Harkins and D. Carrel. The internet key exchange (ike). Technical Report RFC-2409, 1999.

[26] D. Eastlake. Domain name system security extensions. Technical Report RFC2535, IETF, June 1999.

[27] D. Eastlake. Dsa keys and sigs in the domain name system (dns). Technical Report RFC2536, IETF, March 1999.

[28] D. Eastlake. Rsa/md5 keys and sigs in the domain name system (dns). Technical Report RFC2537, IETF, March 1999.

[29] D. Eastlake and O. Gudmundsson. Storing certificates in the domain name system (dns). Technical Report RFC2538, IETF, March 1999.

[30] D. Eastlake. Storage of diffie-hellman keys in the domain name system (dns). Technical Report RFC2539, IETF, March 1999.

[31] D. Eastlake. Detached domain name system (dns) information. Technical Report RFC2540, IETF, March 1999.

[32] D. Eastlake. Dns security operational considerations. Technical Report RFC2541, IETF, March 1999.

[33] S. Kent. Privacy enhancement for internet electronic mail: Part ii: Certificate-based key management. Technical Report RFC1422, IETF, 1993.

[34] Verisign. Verisign inc. - www.verisign.com. http://www.verisign.com/.

[35] William Stallings. *SNMP, SNMPv2 and RMON: Practical Network Management*. Addison Wesley, 1996.

[36] D. Harrington, R. Presuhn, and B. Wijnen. An architecture for describing SNMP management frameworks. Technical report, IETF, 1998.

[37] D. Levi, P. Meyer, and B. Stewart. SNMPv3 applications. Technical report, IETF, 1998.

[38] University of Twente. The simpleweb - IETF network management RFCs by number.

[39] TINA-C. Tina-c home page. http://www.tinac.com.

Chapter 8

Equipment Choice

One of the inevitable tasks of network design is selection between rival manufacturers of "basically" identical equipment. Although the items of equipment (router, switch, hub, transmission system, etc) do provide basically the same function as each other, in many cases the little differences can amount to a lot. Then there is the matter of price, and also the range of capacities dealt with in one manufacturer's catalog might be quite different from that in another's catalog.[1]

These issues are genuine and can be crucial in the evaluation of which supplier should be preferred in a large project, however, it is not easy to provide a lot of advice concerning them – mainly because the range of choices cannot be guessed until they are laid on the table.

However, there are some issues which are *generic*; that is to say, these issues arise over and over again, no matter which suppliers tender for the project.

Even here it is a brave thing to do to offer advice about which choice to take, so, rather than emphasizing the decision, we shall emphasize the decision process. So, even if the conclusions we draw today are quickly invalidated by changes in technology, hopefully the methods of evaluation will have some longer term value.

In particular, in Section 8.2, we set out an *economic model of layering*, the purpose of which is to identify when a layer is economically justified. In the next section we briefly survey the types of equipment under consideration and in the final section we use the economic model of layering to draw some tentative conclusions regarding some particular choices facing network managers today.

8.1 Categories of Equipment

We need to understand the range of technical choices available for building networks. Here is a list of equipment types which arise in network projects:

1. transmission systems
   (a) Purchased
      (i) optical fiber
      (ii) point-to-point
         - microwave systems
         - spread spectrum: licensed or unlicensed.
      (iii) broadcast radio system
         - to access mobile units
         - to access fixed units
      (iv) access cable
         - multipair cable
         - ADSL
         - hybrid-fiber coaxial cable (HFC)

- optical fiber to the home

(v) satellite (part or whole)

(b) Rented

(i) digital circuit

(ii) packet service – point-to-point

(iii) packet service – switched (e.g. Frame Relay, ATM), ...

(iv) Internet access

(v) shared access to local access cable

(vi) shared access to local access radio systems

2. Routers

3. Switches and Hubs

4. Radio Antennae and transmitters

5. Network Management and Control Equipment

6. Terminal Equipment

The choice of which types of equipment to use, and how to make use of the chosen equipment (e.g., where to put the antennae) in some cases requires a certain amount of design skill, which we will consider in the next chapter. In particular, the issue of where to place base stations in a radio access network is considered in Section **??** and the questions which arise in designing cable access networks are considered in Section **??**.

8.2 A Cost Model of Switching and Transmission

This section is concerned with models of the cost of core networks.

Definition 8.1 *The* raw transmission cost *of a transmission system is defined as the cost per transmitted bit/s (cpsbps).*

Definition 8.2 *The* raw switching cost *of a switching system is defined as the cost per switched bit/s (cptbps).*

For example, suppose a 1.6 Terabit/s optical fiber transmission system costs $8,000.00 for the terminations on both ends. The raw transmission cost of this system is $8000 \times 100 / 1.6 \times 10^{-12} = 2.5 \times 10^{-7} = 0.0000005$ cents per transmitted bit/s. Now consider a switch or router, or switch-router with 32 OC-3 ports (155.52 Mbit/s). Suppose this switch costs $40,000.00. The raw switching cost of this switch is $\$40000/(32 \times 155.52 \times 10^6) \approx 0.0008$ cents per bit/s.

As networks increase in size and throughput, their cost becomes more and more a function of the *raw switching and transmission cost*. As a consequence:

> *The asymptotic cost of a core network, per unit of shipped traffic, is determined by the raw switching and transmission costs.*

In large national networks, the cost of *installing* fiber probably dominates the cost of the terminations at the moment, however this cost is basically a fixed cost, independent of traffic volume. Thanks to the Internet, and a succession of new services which consume higher and higher volumes of bandwidth, we do have steadily growing traffic volumes. So, when we consider broad architectural decisions, the cost of installing fiber is not relevant – it is the routing/switching and the transmission cost which is most relevant, and of these it seems likely that the switching/routing component will dominate.

On the other hand, access network costs should not be expected to conform to this principle because it is difficult to achieve high levels of utilization, and therefore, of efficiency, in access networks.

Example 8.1. The Cost of a download

Suppose we dial up and download a 50 Megabyte file. How much of our ISP usage costs can be ascribed to routing/switching and transmission costs of the networks we are using?

A significant part of the cost should be ascribed to the cost of installation of the ducts, cable and fibers used in the telecommunication network that we are using. These are fixed costs, however, and will therefore gradually decline as a component of incurred cost.

The more intesting aspect of cost is the traffic dependent cost. If the path followed traverses 15 hops, for example, which would not be unusual, we would incur $15\times$ the transmission cost and $15\times$ the switching cost just indicated. These costs have been expressed, so far, in cents per bit/s. To work out how much this represents as a *cost per bit* for individual users, let us assume that 20% of these equipment costs are returned, in revenue, by means of charges passed on to users, each year and that the utilization level of the switching and transmission equipment is 20%. There are approximately 32 million seconds each year, so, under these assumptions, and, further, assuming that the switching has a cost of κ cents per bit per second, the cost per bit of switching should be

$$\frac{15\times 0.2/0.2}{32000000}\times\kappa\approx 5\times 10^{-7}\kappa$$

cents per bit. Using the above estimate of $\kappa = 0.016$, this suggests that a 50 Megabyte file incurs switching costs of

$$5\times 10^{-7}\times 0.0008\times 50\times 10^{6}\times 8 = 200\times 0.0008 = 0.16 \text{ cents.}$$

Transmission costs for the same download would be much less than this.

The switching (and transmission) costs accounted for here are actually only appropriate to account for the core network. The switching cost incurred in the access component of the Internet can reasonably be expected to be much higher than that in the core of the network. For example, there is no reason to expect that a 1.6 Terabit/s transmission system can be filled to anywhere remotely like its capacity except in the core network. For this reason, lower capacity transmission systems must be used in the access part of the network, and therefore costs will be much higher.

Suppose, for example, that of the 15 hops over which the download takes place, 5 can be said to be in an access network where switching is $100\times$ more expensive and transmission is $1000\times$ more expensive. Then, routing/switching will still be the dominant cost which will be approximately $\frac{500}{15}$ times as much, i.e. \$0.15.

More accurate estimates of switching and transmission costs associated with real networks are not easy to characterise and are likely to be highly dependent on the individual situation.

The point of this sort of study is not so much to estimate the cost of a single download but rather to estimate how the cost of providing this type of service can be expected to change as technology costs change. □

Suppose we wish to compare a network, *A*, with layers 1 and 3, only, to a network, *B*, with layers 1, 2 and 3. Or, following the same lines of argument, suppose we wish to compare a network, A, with layers 1, 2 and 3 with a network, *B*, with layers 1, $2+2'$, and 3. The layer $2'$ is *additional layer 2 switching* (e.g., as in MPLS). In other words, the layer 2 switching might alread be available in network A but in network B we make much more use of it than in network A.

For simplicity of notation, let us concentrate on the first case, i.e. network A has layers 1 and 3 only, and network B has layers 1, 2 and 3. Suppose the ratio of the cost of layer 2 switching to layer 3 switching (or routing) is R and the length of layer 3 paths in A is L_A while in network B it is L_B.

We expect $R < 1$, i.e. the "new" layer is cheaper than the old one, and that $L_B < L_A$, so the network with the additional layer has shorter paths in layer 3. If this wasn't the case, the additional layer would only add to cost. It is the bypassing of layer 3 functionality which reduces the cost of network *B*. However, there are also costs incurred in adding this new layer, and we need to be careful that these costs are adequately compensated for by the reduction in lengths of the layer 3 paths.

Let us denote the increase in cost due to the extra overheads required by each packet by O_h. For example, if the average packet length is 500 bytes, before the extra layer is added, and the addition layer adds 8 bytes to the total header length, we would set $O_h = 12\%$.

Finally, let us denote the total cost of network A by C_A and the total cost of network B by C_B and let us suppose that the introduction of the additional layer in network B incurs a once-only cost of C. Then the ratio of the cost of

B to the cost of A is:

$$\begin{aligned} R_{B/A} &= \frac{C_B}{C_A} = \frac{(O_h R C_A + C_A L_B/L_A)}{C_A} + C/C_A \\ &= O_h R + L_B/L_A + C/C_A. \end{aligned} \tag{8.1}$$

Explanation

The term inside the brackets on the RHS of the first line above can be explained as follows. RC_A denotes the cost of the additional layer 2 switching in network B. This has been multiplied by O_h to take account of the extra overhead required in packets because of the layer 2 switching. Network B still needs to have some layer 3 routing as well. The cost of this is accounted for in the second term inside the brackets: $C_A L_B/L_A$. We do not need layer 3 switching at every hop – only in the proportion L_B/L_A of hops. Then, because we seek the *ratio* of the cost of network B switching to the cost of network A switching, all of this must be divided by C_A.

In an extreme case, where layer 3 paths reduce almost to zero length, the ratio of switching cost approaches $O_h R$, or, nearly, R, the ratio of switching cost in layer 2 to routing/switching cost in layer 3.

Note that if $R \geq 1$ there is no point in introducing an additional layer – the layer 2 switches must be cheaper than layer 3 router/switches in order to even contemplate bypassing layer 3!

The layers in this simple cost model of layering do not necessarily correspond to the traditional concept of layers 1, 2 and 3. Let us now consider some examples. In the first example the layers in question *do* correspond to the traditional concepts of layers 1, 2 and 3. In the second they do not.

Example 8.2. Multi-Protocol Label Switching (MPLS) with TCP/IP over ATM

MPLS was described in Subsection 5.4.1. In this example we consider an implementation in which the underlying switching layer uses asynchronous transfer mode (ATM). Let us suppose that the switching cost in the ATM network is 0.0005 cents per switched bit per second and the routing cost in layer 3 is 0.003 cents per routed bit per second. Thus, $R = 0.16$.

The additional overhead of packing IP packets into ATM cells is the 5 bytes of ATM header per ATM cell (ignoring the minor issue of unfilled ATM cells), and so $O_h = 53/48 = 110\%$. Let us suppose that paths through the network without MPLS are 16 hops in length and that by the use of MPLS layer 3 paths can be reduced in length to 8 hops.

Also, for simplicity, let us assume that the setup cost of the additional layer can be neglected. then, the network with MPLS which uses ATM switching to bypass routing will be lower in cost by the ratio

$$1.1 \times 0.16 + \frac{8}{16} = 0.676$$

so the network with ATM will be two thirds of the cost per switched bit of the network without ATM. □

It should be kept in mind that these savings in switching cost, highly significant though they appear to be, will only occur when the network is operating near capacity. If the network is lightly loaded, efficiency advantages due to a better approach to switching or routing are not so important.

Example 8.3. IP over SONET / SDH

SONET/SDH add-drop multiplexors and cross-connects are capable of switching very high bandwidths. The complexity of this switching process is much lower than either ATM switching or IP routing and it is to be expected that over time the number and bandwidth of the ports connected to a cross-connect will both steadily increase, with a less than proportionate increase in cost, leading to lower and lower costs per switched bit per second.

On the other hand, by the nature of SONET/SDH cross-connects, the *efficiency* of the SONET/SDH switched bandwidth could be more of a problem. In the case of SONET/SDH there is an overhead of about 10% which can be avoided by using a very simple protocol directly on the optical fibers. However, this is not the only additional overhead incurred. If SONET/SDH switching is used extensively, the utilization level of the switched SONET/SDH channels will be somewhat lower than would be the case if the IP traffic was carried directly on the optical fibers.

Suppose the utilization level of optical fibers would be 50% if the IP traffic was carried directly on the fibers and that the extensive use of SONET/SDH cross-connects might mean that IP traffic is carried on channels of a

capacity one tenth, on average, of that of the whole optical fibers. The standard deviation to mean ratio of the traffic in these smaller links can be expected to be larger by the ratio $\sqrt{10} \approx 3$.

The overhead to allow for random variation should be set as a certain number of standard deviations above the mean. Since the standard deviation to mean ratio is increased by a factor of 3, in this scenario, instead of 50% utilization, we will need to adopt 25% utilization. So, on this account, we have an additional overhead factor of 2.

However, the situation is really worse than this because not only are the switches utilized at a significantly lower rate, but the optical fibers themselves are utilized at a lower rate. This is actually true in the previous example also, but perhaps not so significant. But in the present case the significance of this effect is much greater because we are talking about, potentially, a much larger overhead.

Let us suppose that the ratio of SONET to IP routing cost of switching per switched bit per second is 0.05, $L_A = 10$, and $L_B = 20$. Then, applying (8.1), we find that the network *with* SONET/SDH will be cheaper by the ratio

$$1.1 \times 2 \times 0.05 + 10/20 = 0.61,$$

that is to say, the SONET/SDH equipment, used in the right places, can save 40% of the switching cost in this network. □

Issue: will R remain significantly different from 1? Answer: this question must be answered by reference to the rate at which network traffic is growing and the rate at which CPU speed is increasing. Is it possible that IP routers can operate at line speed for large and larger switches? This depends critically on the technology used to make routers go faster. In a sense, the key element which allows a router to go faster is the cache. The concept of cache is very simple but surprisingly powerful, and seems to be surprisingly effective in all sorts of places – inside computers, in proxy servers, in individual workstations, and in routers. Whether router cache can keep ahead of the increasing load of traffic, including more and more new connections, is unclear at this stage.

8.3 Switching and Routing Choices

There are certain choices which arise over and over again, for the managers and technical staff who are deciding on the architecture and configuration of new networks: switches vs hubs; ATM vs switched Ethernet; SDH vs raw fiber; Voice over IP vs segregated telephony switching. Without attempting to prescribe what the decisions should be in each case, let us survey and explore the *methods of evalution* of these choices.

8.3.1 Layer 2 Switching Choices

Ethernet switches have the potential to increase the capacity of an Ethernet LAN considerably. Higher speeds of operation can also increase LAN capacity.

The higher the transmission speed of an ethernet LAN, the *shorter* the maximum distance which is allowed between the hub and the host. For this reason, even though high speed hubs reduce in cost to the same level as the lower speed hubs, there may be situations where lower speed hubs are preferable. Also, the maximum sustainable transmission speed to or from a host depends on the hardware and software in the host as well as the transmission speed of the LAN, so in some cases there would be no point in increasing the LAN speed.

An ATM switch and an ethernet switch provide comparable functionality – they can both switch IP packets. However, an ethernet switch will usually be cheaper and an ATM switch provides a number of additional features. In particular, an ATM switch can provide network-wide routing, and has a signalling protocol (PNNI – See §5.1.7) capable of supporting wide-area networking of arbitrary complexity.

ATM switching also incorporates in an integral manner facilities for ensuring Quality of Service in a range of categories. This feature of ATM technology has absorbed a considerable amount of time and effort of standardisation bodies, academic researchers, and also industrial developmen (by ATM switch manufacturers). However, since the vast majority of traffic flowing through ATM switches is IP traffic channelled through them as a wholesale bandwidth transport and delivery network, the PNNI facility is in reality the main feature of ATM switching technology which distinguishes it from alternatives which could perform nearly the same tasks.

ATM facilitates LAN emulation and switching which can be dynamically set up to bypass routers – which reduces latency and lowers load on routers – but in a local network of sufficiently small size, this can also be

achieved with ethernet switches. If the network under study become sufficiently large that ethernet switching is difficult to manage and ATM may be the best solution. This might be the case, for example, if the network includes paths of 3 or more hops. When MPLS is widely available, routers which implement IP (over IP) MPLS will compete with ATM in this area of its applicable domain.

8.3.2 Layer 1 Switching Choices

What is Layer 1 switching? This perculiar term is used here to refer to *cross-connects*, of which we need to distinguish two varieties: SONET/SDH cross-connects and optical cross-connects. Some cross-connects offer both types of interconnection in the one peice of equipment. A pure optical cross-connect would be expected to switch one wavelength, in one optical fiber, to one other wavelength in another (or the same) fiber. This type of equipment can be expected to achieve costs per switched bit/s lower than any other type of switch. The applications of such switches are to rather high capacity networks, where multiple, high usage, optical fibers are in simultaneous use. This would seem to apply mainly to carriers, at the moment, although it is conceivable that large industrial concerns might require similar facilities under special conditions.

If optical fiber to the home becomes a viable access technology in the future, which seems likely, a very considerable expansion in installation of optical fibers and their terminal equipment, and hence also of optical switches will be expected.

The SONET/SDH cross-connect facility at present has a key role in carrier networks. These switches provides the lowest cost per switched bit/s aside from optical switches and the port size and multiplicity of SONET/SDH cross-connects lies in a range of considerable utility.

Exercise 8.1. Choice of Layers

Suppose you are the designer of new network which will in the near future be required to carry considerable volumes of IP traffic between locations over a wide area, of sufficient diversity that it is clear that at least 20 switching/routing nodes will be needed.

You have a choice of the technologies indicated in Table 8.1

Switch type	O_h	R	L_B/L_A
IP Routing	1	1	1
ATM Switching	1.1	0.5	0.4
SONET/SDH cross-connect	2	0.1	0.4
Optical cross-connect	4	0.01	0.4

Table 8.1: Switching Alternatives in a high capacity network

The first column here is the type of switching equipment. The second column is the overhead introduced by this type of equipment – these overheads should be interpreted as applying to the layer below, whatever that happens to be. For example, if the only layers are the optical cross-connect layer, the ATM layer and the IP routing layer, the total additional overhead due to all layers, relative to an IP-only architecture would be $4 \times 1.1 = 4.4$.

The value of R, on the other hand, should be interpreted as relative to the IP switching function. Thus, optical cross-connects are assumed to provide a cost per switched bit/s one hundredth of the raw switching cost of IP routing.

Finally, the values of L_B/L_A are assumed to apply to a comparison between the layer in question relative to the layer immediately above, if it is in use, or, otherwise, to the layer in question relative to whatever layer is immediately above.

Questions:

(a) If at most three layers can be used, one of which must be IP, which should be selected?

(b) Suppose each layer will be introduced only if it leads to at least a 20% reduction in cost per switched bit/s. Which layers should be introduced?

□

Exercise 8.2. Routing vs Switching vs hubs

One of the basic trade-offs we need to consider when designing networks is that between hubs, switches, and routers. Give three examples in which, respectively, hubs should be used in preference to switches, switches should be used in preference to hubs or routers, and, lastly, routers should be used in preference to switches.

Your answer should take the form of a short essay (no more than three pages in length). Make sure to provide an explanation of any assertion you make concerning your example, and, in particular, you should try to explain clearly *why* your example fits the bill for the particular case. □

References

[1] R.H. Deng, L. Gong, A.A. Lazar, and W. Wang. Practical protocols for certified electronic mail. *Journal of Network and Systems Management*, 1996.

Chapter 9

Design

This chapter is about *design*. "Design" is a word so over-used that the meaning has tended to wear rather thin. However, we shall approach the topic of design from a very simple point of view, so that many of the problems we face can be tackled by a very simple, familiar technique: the concept of *present value* and the method of present value analysis as it applies to network design. In this way we can hope to be able to make sensible decisions concerning the planning of and the design of networks for small and large organizations.

Now that we know, from Chapter 3 how to choose the right *size* of link, we can tackle the larger task of *planning* the installation of equipment over a number of years, taking into account growth, changes in costs of technology, and the cost of borrowing.

Our main tool will be the concept of *Present Value*. The reason for this is that in many cases design reduces to a series of *choices*. The fact that certain choices affect other choices complicates matters somewhat, but certain choices can also simplify the remaining problems.

Traditional network design is rather concerned about the speeds, quantities, or sizes of certain components. However, modern transmission systems come in a rather coarse range of sizes. If 155 Mbit/s is not enough, the next option might be a 1.6 Terabit/s. The standard transmission rates in both the SONET standard and the SDH standard are multiples of the SONET base rate of 51.84 Mbit/s [1]. However, not all multiples of this base rate are actually used. the OC-2 rate is not actively used, for example. Since transmission systems capable of carrying 1.6 Terabits/s, formed as 160 OC-192 systems, can now be carried on a single optical fiber [2], the cost model for a single link is no longer remotely similar to the "classical" model of linear increase with capacity.

9.1 Algorithms

In this section some classical network design algorithms will be reviewed. These algorithms have mostly been known for some time and are either extremely easy to implement or implementations are available from many sources.

9.1.1 Minimal Spanning Tree

[3, Chapter 23]

The minimal spanning tree for connecting a set of nodes is the tree network which has minimum possible total length. It is usually unique but it is also possible that two or more networks could have exactly the same cost.

There are two variations upon this problem. In one case, the problem starts with a set of n nodes with distances given from each node to every other node. For example, if each node is given geographical coordinates, the distances between any two nodes can be calculated. The problem is then to choose $n-1$ pairs of nodes which will be joined together, by links, to create a connected graph.

In the second variation, instead of starting with a geographical context for the nodes, we are given an existing connected graph, in which each link has a *distance* or *cost*. The costs in this case do not necessarily correspond to geographical distances. The problem is then to choose, again, $n-1$ links, which will make a connected graph. In this case, the links must come from the links in the existing graph.

Problems of the first sort can be converted to problems of the second by creating a complete graph in which the links are all labelled by the geographical distance traversed.

The minimal spanning tree problem and algorithms for its solution is a standard item in the syllabus of algorithms and data structures [3, Chapter 26] [4, Section 9.5]. There are two algorithms with very similar performance for this problem: Prim's algorithm and Kruskal's algorithm.

Kruskal's algorithm finds the minimum cost spanning tree by starting with an empty forest and adding trees (links). It proceeds by selecting at each stage the least cost link which does not introduce a loop into the graph so far chosen. Since every set of n links has at least one loop, and no graph with fewer than $n-1$ links can connect all the nodes, the algorithm always requires precisely $n-1$ steps.

Example 9.1. Find a Minimal Spanning Tree

Consider the graph in Figure 9.1.

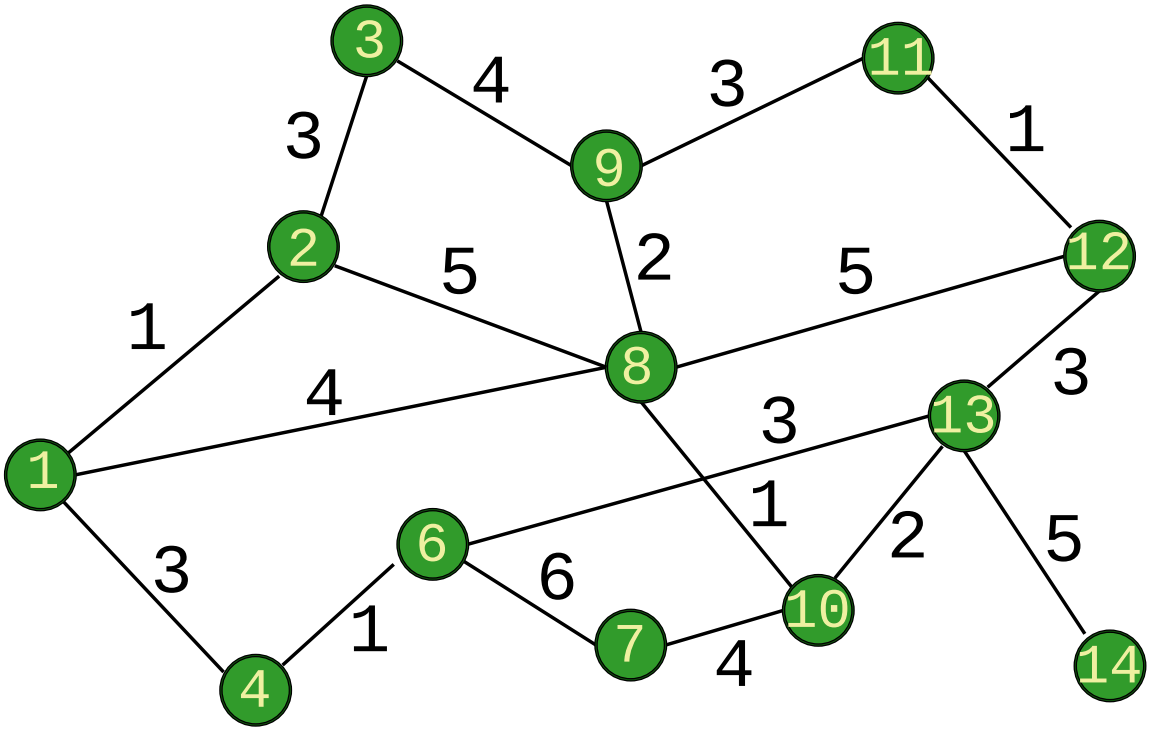


Figure 9.1: A Minimal Spanning Tree Problem

The first four steps of the algorithm will add the links of length 1, the next steps the links of length 2, then the links of length 3, except for the one between nodes 9 and 11, which creates a loop. At this stage, the network constructed is depicted in Figure 9.2. Note that instead of leaving out the link from 9 to 11 we could have left out the link from 12 to 13. Either choice works just as well with all the subsequent decisions.

Finally, a link of length 4 is added to connect node 7 to the network and a link of length 5 is added to connect node 14 to the network. This produces the network shown in Figure 9.3.

If we had chosen to leave out the link between 12 and 13 instead of between 9 and 11, at the earlier step discussed above, the subsequent steps would be exactly the same and the resulting network would be the one shown in Figure 9.4. The cost of this network is exactly the same as the one in Figure 9.3.

□

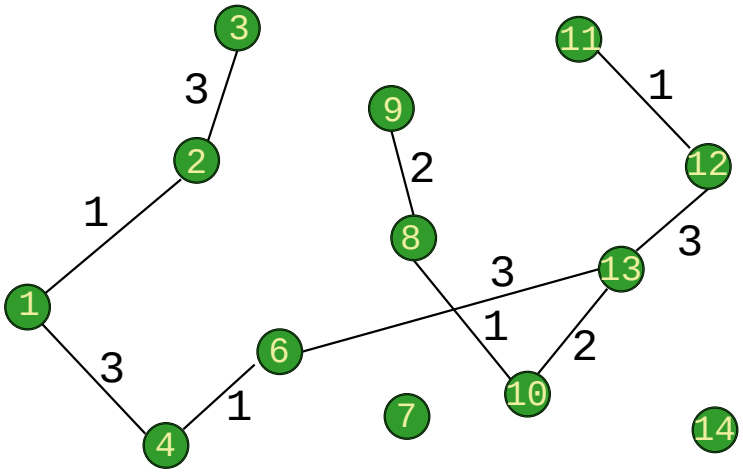


Figure 9.2: A Minimal Spanning Tree Problem

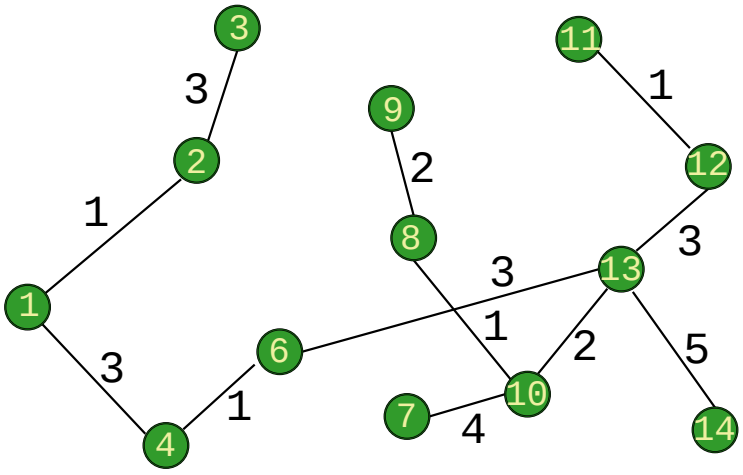


Figure 9.3: A Minimal Spanning Tree Solution

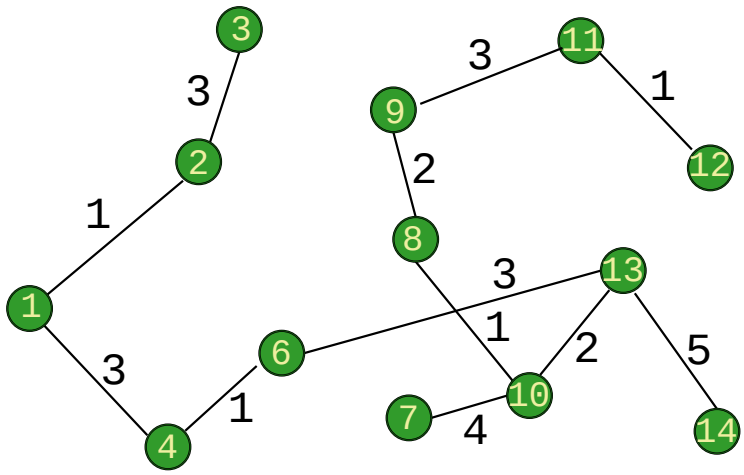


Figure 9.4: An Alternative Minimal Spanning Tree Solution

Exercise 9.1. Find a Minimal Spanning Tree

Replace the link costs in Figure 9.1 by the difference between 10 and the link costs shown, i.e. if the link cost in Figure 9.1 is x, replace this cost by $10 - x$, and then re-solve the problem. □

9.1.2 Maximum Flow

[3, Chapter 26]

The *Max-flow algorithm* is another classical network algorithm. We view the network as a network of pipes and we take as our goal the pumping of as much water as possible from one specified host, the source, to another specified host, the sink. Links may be directed or undirected, and the network can be made up of a mixture of directed and undirected links.

The algorithm proceeds as follows. In order to explain this algorithm effectively, we shall use an example based on Figure 9.1.

1. Find the path from the source to the destination with maximum value for the minimum link capacity, of any links along the path, *if such a path can be found*. If no such path can be found, the algorithm has completed and the max-flow has been found. A small variation of Dijkstra's algorithm can be used to find the path with the maximum value for the minimum link.

   For example, if we apply this step to the network of Figure 9.1, we find that the best path (the one which can carry the most flow) is the one which goes from node 1 through node 8 to node 12.

2. Add the identified flow to a graph of all the flows so far found. The result, at the first step, applied to the network in Figure 9.1, is shown in Figure 9.5.

3. Remove the flow just found from the capacity values of the links along the path used and add capacities to these links in the *reverse direction to the flow*, to indicate that by undoing this decision to carry this flow on a certain link, a flow can virtually be carried in the reverse direction.

   For example, after the first step of this algorithm starting with the network in Figure 9.1, we come to the network depicted in Figure 9.6.

4. Go back to Step 1.

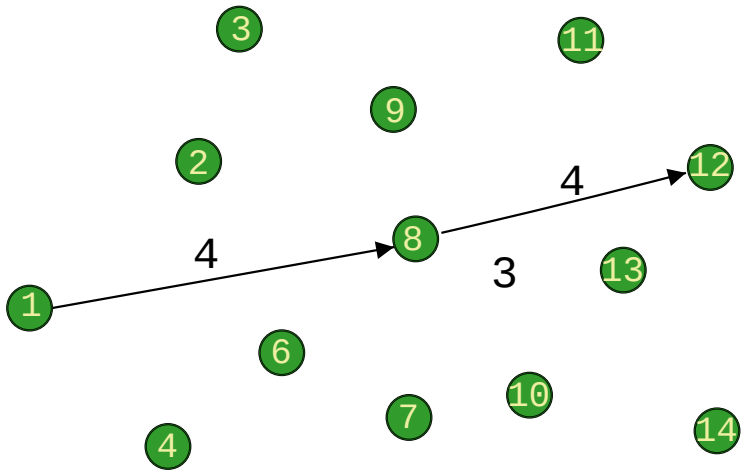


Figure 9.5: The flow which has been allocated at the end of Step 1 of the Max-flow algorithm

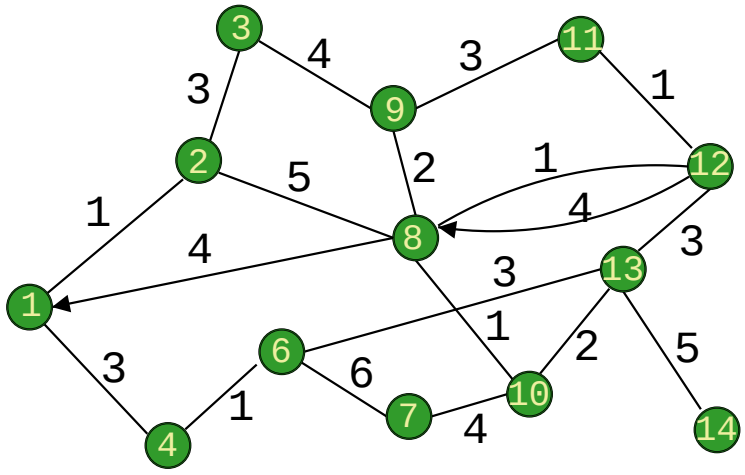


Figure 9.6: The network after Step 1 of the Max-flow algorithm

Exercise 9.2. The Max-flow Algorithm

Continue the application of the max-flow algorithm to the network of Figure 9.1 to work out the maximum possible flow. □

9.1.3 Linear Programming

Many network problems, including the ones discussed up to this point in this Chapter, can be framed as *linear programming* problems.

A linear programming problem is the task of finding an assignment to a collection of variables, $x_1, x_2, \ldots, x_n$ say, such that a certain objective function, e.g.

$$\sum_{i=1}^{n} c_i x_i,$$

is maximised while a collection of constraints

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i, \qquad i = 1, \ldots, m,$$

also hold.

Example 9.2. Shortest Path Problem as Linear Programming

In the shortest path problem our objective is to minimise the length of the path between a certain origin and a certain destination. Let us see if we can formulate this problem as a linear programming problem.

For the variables, x_i, $i = 1, \ldots, n$, let us choose the "amount" of link i which forms part of the optimal path. Hence, in the optimal solution, each x_i is either zero of one. We can't include such a constraint in the definition of the problem without going beyond the definition of linear programming, however this might not matter. If we merely constrain the x_i to lie *between* 0 and 1, it might turn out that the optimal solutions to a certain carefully selected linear programming problem will automatically enforce 0 or 1 solutions.

The objective function can be stated relatively easily as

$$\text{Maximise } -\textstyle\sum_{i=1}^{n} c_i x_i,$$

in which the coefficients c_i are the link costs.

We have already noted the constraints that $0 \leq x_i \leq 1$, $i = 1, \ldots, n$. In addition, we need to be sure that *a path* is included in the collection of links i with $x_i = 1$. Here is how to do this. First, enumerate all the possible paths. We will need a separate algorithm to do this. Let m denote the the number of possible paths and let

$$a_{ij} = \begin{cases} 1, & \text{if link } i \text{ is in the path } j, \\ 0, & \text{otherwise,} \end{cases}$$

$i = 1,\ldots,n$, $j = 1,\ldots,m$. Now let us introduce some more variables which denote the traffic on a path: y_j, $j = 1,\ldots,m$. These are related to the traffic on links by the constraints

$$\sum_{j} a_{ij} y_j = x_j, \qquad j = 1, \ldots, m,$$

and in addition, of course, we require $0 \leq y_j \leq 1$, $j = 1,\ldots,m$. The constraint that *a path* succeeds in linking the origin and the destination can now be stated as:

$$\sum_{j=1}^{m} y_j = 1.$$

This last constraint forces the choice of the value for the variables x_i to be 1 rather than somewhere between 0 and 1. □

This is not meant to provide a sensible way to formulate and solve a minimum path length problem. However, it is rather interesting to see that the important shortest-path problem can be stated as a linear programming problem. In fact, there are several quite natural variations of the shortest path problem. We have already alluded to some of these: finding all the shortest paths to a certain node, all the paths from a certain node, and finding all the paths between every node and every other node. These variations can also be stated as linear programming problems.

The linear programming algorithm is not particularly effective in solving these problems. Rather, the relationship between these problems is used the other way around. We should use the fact that certain linear programming problems can be solved very quickly and effectively by algorithms like Dijkstra's algorithm to alert us to the fact that we may be able to solve linear programming problems arising in networks by means of much faster algorithms.

Exercise 9.3. The Maximum Flow Problem as a Linear Programming

Formulate the Maximum Flow Problem as a Linear Programming Problem. □

We return to the application of linear programming to network design in §9.5.

9.1.4 Integer Programming and Mixed Integer Programming

If we add the constraint that the variables, x_i, must be integers, the resulting problem is termed an *Integer Programming Problem*. If some variables are required to be integers and others are allowed to take a continously varying range of values, the problem is termed a *Mixed Integer* problem.

The key to solving Integer Programming and Mixed Integer Programming problems is to find some way to search through the enormous range of possible solutions, looking for the optimum choice, without checking each possible choice one by one. The key to being able to do this is that even when the variables are constrained to be integers, an ordinary linear programming problem can be used to guide or constrain the possible choices.

It is natural to apply the integer programming model to network problems because the choices that must be made in network design are very seldom nowadays from a continuous range of values. As discussed above, for example, the choices of transmission speed to use over an optical fiber jump from 0 to 155.52 Mbit/s to 466.56 Mbit/s and so on. In the case of leased line facilities, on the other hand, a more smooth range of choices may be applicable.

9.1.5 Non-linear Optimization

Another variation of the linear programming problem occurs by replacing the linear objective function by a non-linear objective function, or by replacing the linear constraints by non-linear constraints, or both.

For example, the performance constraint which states that loss levels should be below 10^{-3}, can be conveniently restated by requiring that link capacities should be at least 3 standard deviations above mean traffic levels. When a network design problem is formulated as an optimization problem, there is likelyh to be a variable corresponding to the link capacity, and this variable appears in this constraint in a linear fashion. However, there is likely to be another variable in this constraint, the mean traffic level. This variable appears in the constraint in a linear and a non-linear fashion, because although the mean traffic level appears directly, in a linear fashion, it appears indirectly, by implication, in the *standard deviation*.

9.1.6 Travelling Salesman Problem

One more classical network design problem should be considered: the *Travelling Salesman Problem*. Given a network, with undirected links, each with a positive cost, the travelling salesman problem is to find a circuit, i.e. a path through the entire network which visits each node exactly once, which has the least total cost, where the cost is defined as the sum of the link costs.

This problem is *NP Complete*. Generally speaking, the term NP Complete is taken as meaning *very hard to solve*, especially for large problems. As far as we can tell, none of the NP Complete problems in computer science have an algorithm by means of which the stated problems can be solved unless the time taken to find the solution increases faster than any polynomial with the size of the problem.

Despite this, there are actually many NP Complete problems for which the known best algorithmic solutions are actually quite satisfactory. This appears to be true of the Travelling Salesman problem itself – i.e. although there

is no known, simple, algorithm which is able to solve this class of problems in a time bounded by a polynomial in the complexity of the prblem, there are some very good algorithms which make no claim to be able to solve the problem in question in polynomial time.

These algorithms – Max-flow, shortest path, minimum spanning tree, linear pogramming, integer and mixed-integer programming, nonlinear programming, and the travelling salesman problem – comprise a rich collection of models and their solution. It is difficult to say where one or other of these algorithms will crop up next. It is just wise, if possible, to be equipped to apply one of these techniques when appropriate.

9.2 Present Value Analysis

The *Present Value* concept allows us to take into account the *cost of (borrowing) money*. This concept can also, in a rough sort of way, be used to take into account the reasonable expectation that costs of technology will reduce.

Here is how it works. Suppose we are contemplating purchasing, installing, and maintaining a network of switching and transmission equipment and we have two alternative plans which we wish to compare to work out which we should adopt. In the first plan, Plan A, we will need to spend $50,000.00 in the first, second and third years, $40,000.00 in the fourth, fifth and sixth years, and $20,000.00 every year, for the forseeable future, thereafter.

In Plan B, we can cater for the same need by spending $300,000.00 in the first year, $40,000.00 in the second year, and nothing thereafter.

How can we decide between these alternative plans? The money for either of these plans will come partly from cash flow of the business and partly from borrowing.

For simplicity, let's assume that all the money, in both cases, has to be borrowed, and let's work out the total cost over a long period of time, for example, 10 years. Let's suppose that the interest rate is 10%, charged annually at the end of the year in which the expenditure takes place. (In a more detailed study, the time when the expenditure occurs, during the year, would also need to be taken into account).

The cost of the both plans over ten years is calculated in Table 9.1. So the first plan is better even though it

	Plan 1				*Plan 2*			
Year	**Expend**	**Multiplier**	**Cost ($,000)**	**Cumulative**	**Expend**	**Multiplier**	**Cost ($,000)**	**Cumulative**
				0				
1	50	2.59	129.69	129.69	300.00	2.59	778.12	778.12
2	50	2.36	117.90	247.58	40.00	2.36	94.32	872.44
3	50	2.14	107.18	354.76		2.14	0.00	872.44
4	40	1.95	77.95	432.71		1.95	0.00	872.44
5	40	1.77	70.86	503.58		1.77	0.00	872.44
6	40	1.61	64.42	568.00		1.61	0.00	872.44
7	20	1.46	29.28	597.28		1.46	0.00	872.44
8	20	1.33	26.62	623.90		1.33	0.00	872.44
9	20	1.21	24.20	648.10		1.21	0.00	872.44
10	20	1.10	22.00	670.10		1.10	0.00	872.44

Table 9.1: Comparison of Costs of Plans A and B

entails greater total expenditure over the ten years.

If we considered these two plans over eleven years instead of ten, the numbers would be different, but the decision would be the same.The numbers would change because we would have an extra year of interest to pay, and so the cost column would be multiplied by 1.1 for both plans. In addition, we would need to add another expenditure in the *expend* column for Plan A. However, this would make little difference to the comparison.

In fact, no matter how many additional years we considered, Plan A would still turn out to be better because even though those extra expenditures would be added to the cost of Plan A, Plan B would be getting worse because of the interest due on the heavy loans taken out at the start.

The choice of the number of years to consider, in this case is somewhat arbitrary, so long as its more than about six years. Since the choice of the ending year for our calculations is somewhat arbitrary, a different way of doing the calculations is usually adopted. Up to now, we have been comparing the total costs of our alternative plans by considering the financial outcome at the *end* of the period under consideration.

In a sense, we have been comparing the two plans in year 10 dollars, or year 11 dollars, etc, depending on when we end our plan. However, we can measure these quantities of money in a consistent quantity, namely year 1 dollars, by a very simple adjustment. Year 1 dollars are effectively worth 1.1^{10} times as much as year 10 dollars. If a year 1 dollar is invested at the interest rate assumed in this example, i.e. 10% p.a., after 10 years it will be worth 1.1^{10} dollars (year 10 dollars). Or, putting this another way, if we borrow a year 1 dollar, and pay nothing off the loan for ten years, by the time we reach year 10, our debt will be 1.1^{10}.

So let us compare the two plans by comparing the number of dollars which would have to be borrowed in year 1 to produce the same result at the end of the period.

If we do this for Plan A we find that the number of year 1 dollars which is equivalent to our entire expenditure plan is $258,350.00 while when we do this for Plan B we find that the entire plan is equivalent to $336,360.00.

Making the comparison in this way doesn't make any difference to the decision, but it means that we can do the calculations in a way which doesn't cause all our numbers to be revised every time the plan is extended for another year or two. In fact, what we should do is to carry out the calculations in year 1 dollars from the start. So, Plan A will cost $50+50/(1+0.1)+50/(1+0.1)^2+\cdots = \$258,350.00$ (year 1 \$) and Plan B will cost $100+55/1.1 = 336,360.00$ year 1 \$ The detailed calculations are shown in Figure 9.2

	Plan 1				*Plan 2*			
Year	**Expend**	**Multiplier**	**Cost ($,000)**	**Cumulative**	**Expend**	**Multiplier**	**Cost ($,000)**	**Cumulative**
				0				
1	50	1.00	50.00	50.00	300.00	1.00	300.00	300.00
2	50	0.91	45.45	95.45	40.00	0.91	36.36	336.36
3	50	0.83	41.32	136.78		0.83	0.00	336.36
4	40	0.75	30.05	166.83		0.75	0.00	336.36
5	40	0.68	27.32	194.15		0.68	0.00	336.36
6	40	0.62	24.84	218.99		0.62	0.00	336.36
7	20	0.56	11.29	230.28		0.56	0.00	336.36
8	20	0.51	10.26	240.54		0.51	0.00	336.36
9	20	0.47	9.33	249.87		0.47	0.00	336.36
10	20	0.42	8.48	258.35		0.42	0.00	336.36

Table 9.2: Comparison of Costs of Plans A and B using Year 1 dollars

This process of comparing plans on the basis of their cost measured in equivalent money at the start of the period is know as *Present Value Analysis*.

Example 9.3. Dimensioning a Link

Let us reconsider the link which was *analyzed* in §3.3. We now wish to estimate the appropriate capacity for this link under the assumptions that:

(i) traffic is doubling every two years (and the variance of traffic increases at a similar rate), starting at the present (year 2000, let's say) rate of 8 Mbit/s for the mean, and 4 Mbit/s standard deviation;

(ii) the link capacity can be selected as 10 Mbit/s, 55 Mbit/s, 165 Mbit/s, or 660 Mbit/s at a cost of $100,000.00, $120,000.00, $200,000.00, or $250,000.00 respectively;

(iii) when an upgrade is required, the cost of the link which was previously installed will need to be completely written off.

(iv) a dollar now is worth 10% more than a dollar one year later.

In order to solve this problem, we shall create a *table* of expected outcomes, year by year, for 8 years. See Table 9.3. The precise number of years to consider is not critical, so long as it goes far enough into the future to reflect the effect of decisions made in the next few years.

year	**traffic mean**	**traffic variance**	**traffic standard deviation**	**Plan A Capacity**	**Plan B Capacity**
1	8	16	4	165	?
2	11.2	22.5	4.76	165	
3	16	32	5.66	165	
4	22.4	45	6.73	165	
5	32	64	8	165	
6	44.8	90	9.51	165	
7	64	128	11.31	165	
8	89.6	180	13.45	165	
9	128	256	16	165	

Table 9.3: Table of Traffic

At the end of the 8 year period, i.e. at the start of Year 9, the traffic offered to the link will have doubled every two years to reach $16 \times 8 = 128$ Mbit/s. The largest item in our transmission equipment supply list caters for this level of traffic.

The variance is also expected to double every two years, so that by 2008 the standard deviation will be 16 Mbit/s. Clearly the 165 Mbit system will be adequate in Year 9, but the 55 Mbit/s system will not be able to cope. Let us call Plan A the approach where the 165 Mbit/s system is purchased at the start of the period, as depicted in Table 9.3.

The next approach, called Plan B, will be to start with a minimal transmission system in the year 2001 and replace this with the 165 Mbit/s system at the latest possible time.

It may or may not be necessary to consider a Plan C. We shall see.

In the first year, the traffic is 8 Mbit/s and the standard deviation is 4 Mbit/s, so the 10 Mbit/s system will clearly be inadequate, while the 55 Mbit/s system will be quite satisfactory. So, in Plan B, the 55 Mbit/s system will be installed in Year 1.

The estimated offered traffic, and its estimated standard deviation at the start of each year are set out in Table 9.4, together with the installations made in each year, and their cost, according to the Plans A and B. The decision as to whether the existing transmission system has sufficient capacity, or whether it needs to be upgraded, is made by adding two standard deviations to the mean and comparing against the link capacity. If the link capacity is the smaller number, the link needs to be upgraded.

The table shows that the plan where a large link is selected for installation in Year 1 is preferable. If a Plan C were to be considered, it would need to include *two* link upgrades – otherwise it couldn't be as good as Plan B, which is the least cost plan which has one link upgrade and finishes with a 165 Mbit/s system in year 9. So, clearly, Plan C does not need to be considered. Plan A is optimal.

□

Exercise 9.4. Planning a Pt-to-Pt link

Choose a plan for installing transmission equipment on a point-to-point communication link under the following assumptions:

(i) traffic is 6 Mbit/s at present, with standard deviation of 4 Mbit/s;

(ii) mean traffic and variance of the traffic is doubling every three years;

(iii) the link capacity can be selected as 10 Mbit/s, 55 Mbit/s, or 165 Mbit/s, at a cost of $100,000.00, $120,000.00, or $150.000.00 respectively;

year	mean traffic	stdev	Plan A				Plan B			
			link	cost	disc.	disc. cost	link	cost	disc.	disc. cost
1	8	4	165	200,000	1	200,000	55	120,000	1	120,000
2	11.31	4.76	-	0			-	0		
3	16	5.66	-	0			-	0		
4	22.63	6.73	-	0			-	0		
5	32	8	-	0			165	200,000	0.68	136,600
6	45.25	9.51	-	0			-	0		
7	64	11.31	-	0			-	0		
8	90.51	13.45	-	0			-	0		
9	128	16	-	0			-	0		
total disc. cost						200,000				256,600

Table 9.4: A table of outcomes, year by year

(iv) when an upgrade is required, the cost of the link which was previously installed will need to be completely written off.

(v) a dollar now is worth 12% more than a dollar one year later.

The two sites and the alternative connecting links are depicted in Figure 9.7.

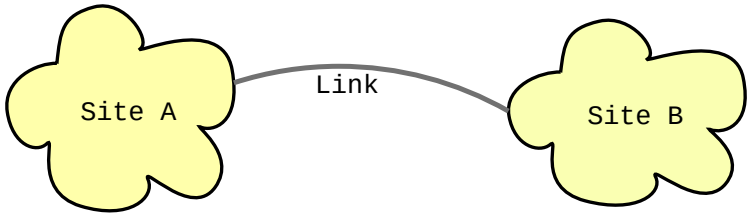


Figure 9.7: Planning a Link

Consider also an alternative case (Case (b)) in which the initial traffic offered to the link is 10 Mbit/s instead of 6 Mbit/s. Also assume that the standard deviation of this traffic is 10 Mbit/s at the outset and that the variance of the traffic grows at the same rate as the mean, i.e. doubles every three years.

Note: in saying that mean traffic doubles every three years, it should be taken also that the mean traffic increases in the ratio $\sqrt[3]{2}$ each year. If traffic was increasing in the ratio x over each successive group of m years, we would normally assume that traffic was increasing in the ratio $\sqrt[m]{x}$ over each year. Of course, all such parameters are merely estimates and growth is unpredictable and varies from year to year randomly, so that the assumption that growth is distributed over successive years as uniformly as possible is only an approximation. However, given the imprecision of all the other considerations, this assumption is probably reasonable.

Explain your conclusions carefully and provide full workings for both cases in your answer. □

9.3 Planning

Let us now consider a more complex situation.

Example 9.4. Linking two Sites

Suppose an organization is located at two sites, A and B, and needs to establish a communication facility between these two sites, as depicted in Figure 9.8. It has a choice between

(i) a dedicated microwave link,

(ii) a leased line from a telecommunications company, and

(iii) a tunnel through the Internet, and

(iv) a combination of the above.

The traffic to be carried is estimated to be 8×64 kbit/s now, and growing at the rate 10% per year of *telephone traffic* plus 160 kbit/s now, and growing at the rate of 50% per year of *TCP/IP* traffic. We shall assume that the standard deviation of the TCP/IP traffic is 160 kbit/s at the start of the period and the variance is growing at the same rate as the mean of this traffic.

The variance of telephone traffic is determined by the fact that call arrivals form a Poisson process. It follows (this is not necessarily obvious) that the number of calls active at any time is also Poisson distributed, and therefore the variance of the number of calls which are active at a certain time is the same as the mean number of calls active at this time, i.e., in the present case, 8. It follows that the standard deviation of the traffic due to telephone calls is $64 \times \sqrt{8}$ kbit/s at the start of the period and growing at the rate 10% every *two years*. (Telephone traffic is discussed in more detail in Subsection 3.3.3.)

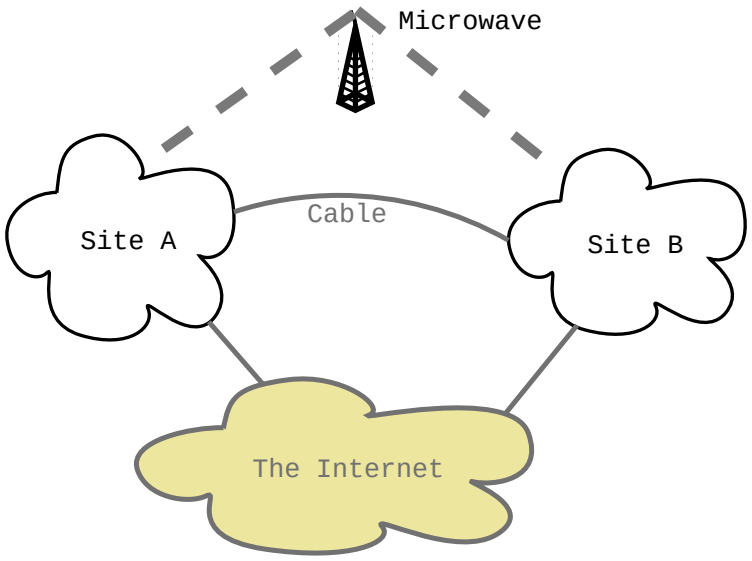


Figure 9.8: Equipment Alternatives for a Link

The telecommunications company link costs and availabilities are given in Table 9.5, the costs and availabilities of a dedicated microwave link are given in Table 9.6 and the costs and availabilities of an *Internet tunnel* are given in Table 9.7. Upgrading the telecommunications company link costs nothing, upgrading the microwave link requires a complete write-off of the cost of the existing link, and upgrading the Internet link is also cost-less.

Let us assume that all technical options are capable of carrying a mixture of telephone and TCP/IP traffic. This would not be the case at present, however it is not unlikely to be realistic in the near term future.

We shall aim to achieve an availability of 99.9% (possibly at reduced capacity, but not at a capacity below 50% of what is necessary), and to ensure that loss of telephone traffic is less than 2 % (when all systems are operational), and nominal loss of TCP/IP traffic is less than 1 % (when all systems are operational).

Also we shall assume that the discount rate for money is 10 % (so a dollar now is worth 10% more than a dollar at the same time next year) and that our aim is to minimize the present value of the cost of your entire network implementation plan, while always meeting all the performance standards which have just been discussed. We shall use a forward plan of 8 years when estimating costs.

capacity (kbit/s)	cost (k$/year)	availability
x	$x/6$	99.9%

Table 9.5: Leased line costs and availabilities

capacity	installation cost (k$)	availability
1 Mbit/s	50	99%
10 Mbit/s	100	99%
45 Mbit/s	200	99%

Table 9.6: Microwave link costs and availabilities

capacity (kbit/s)	cost (k$/year)	availability
64, growing at 50 % per year	2	99%

Table 9.7: Internet tunnel costs and availabilities

The traffic for Example 9.4 over the period of the study (starting, arbitrarily at the year 2000) is depicted in Table 9.8 and the minimum link size to be able to carry this traffic is depicted in Table 9.9. The minimum required link capacity was calculated as the mean plus two standard deviations.

Two reasonably obvious solutions are immediately suggested: Case A. Purchase a microwave link and back it up with the leased line *and* the Internet link; and Case B. Just use the Internet connection with the leased line, supplying the remaining capacity *and* the backup facility (actually, the leased line is sufficiently reliable that it doesn't need to be backed up). The backup capacity in each case does not need to be more than half the required capacity of the link, although in the latter case, where the leased line is both the primary and the backup (except for the contribution made by the Internet tunnel), the leased line will have to carry most of the traffic anyway.

The required capacity on the leased line in these two cases are shown in Figure 9.10. In Case A, where there is a Microwave link *and* an Internet tunnel in use in addition to the leased line, and the microwave link is capable of carrying all the traffic, the leased line only needs to carry half the nominal traffic minus the traffic carried by the Internet.

Since this could be a bit confusing, lets use a bit of algebra to make the formula for the required leased line capacity a bit more explicit. Suppose the nominal required capacity (as on the right of Table 9.9), called T, and the capacity of an Internet tunnel at this time is I. Then the capacity required in the leased line will be $T/2-I$.

There is an alternative interpretation here, incidentally, which is that the backup facility should only need to have a capacity to backup half the *mean* of the traffic. This would cost a bit less, but then the performance would be less satisfactory under failure conditions.

In Case B, the leased line has to carry all the nominal required capacity except for the Internet tunnel.

Finally, discounted costs of equipment are shown in Figure 9.12. It is clear from this table that Case A is more attractive.

□

Exercise 9.5. Equipment Selection for a Pt-to-Pt link

A company needs to choose and install transmission facilities to connect two sites. The traffic to be carried on this link is made up of telephone calls, 12 Erlangs at the moment and growing at the rate 10 % per year, and

year	mean, tel tr (kbit/s)	var tel tr (kbit/s)	mean tcp (kbit/s)	var tcp $(kbit/s)^2$	total mean (kbit/s)	total var $(kbit/s)^2$	stdev (kbit/s)
2000	512	32000	160	25600	672	57600	240
2001	563	35200	240	38400	803	73600	271
2002	620	38720	360	57600	980	96320	310
2003	681	42592	540	86400	1221	128992	359
2004	750	46851	810	129600	1560	176451	420
2005	825	51536	1215	194400	2040	245936	496
2006	907	56690	1823	291600	2730	348290	590
2007	998	62359	2734	437400	3731	499759	707
2008	1,098	68595	4101	656100	5198	724695	851

Table 9.8: Traffic for Example 9.4, year by year

year	mean traffic (kbit/s)	stdev traffic (kbit/s)	reqd capac (kbit/s)
2000	672	240	1152
2001	803	271	1346
2002	980	310	1600
2003	1221	359	1940
2004	1560	420	2400
2005	2040	496	3031
2006	2730	590	3910
2007	3731	707	5145
2008	5198	851	6901

Table 9.9: Required Link Capacities for Example 9.4, year by year

year	reqd capac (kbit/s)	Internet Capacity kbit/s	Leased Line (Case A) kbit/s	Leased Line (Case B) kbit/s	Undiscounted Cost (Case A)	Undiscounted Cost (Case B)
2000	1152	64	512	1088	$187333	$183333
2001	1346	96	577	1250	$98149	$210298
2002	1600	144	656	1456	$111352	$244705
2003	1940	216	754	1724	$127648	$289297
2004	2400	324	876	2076	$147978	$347957
2005	3031	486	1030	2545	$173618	$426237
2006	3910	729	1226	3181	$206322	$532144
2007	5145	1094	1479	4052	$248530	$677311
2008	6901	1640	1810	5260	$303685	$878745

Table 9.10: Costs in Cases A (Uses Microwave Link) and B (No Microwave Link)

year	coeff.	Undisc. A	Disc. A	Cumul A	Undisc. B	Disc. B	Cumul B
2000	1.00	$187333	$187333	$187333	$183333	$183333	$183333
2001	0.91	$98149	$89226	$268448	$210298	$191180	$357133
2002	0.83	$111352	$92027	$344503	$244705	$202235	$524270
2003	0.75	$127648	$95904	$416558	$289297	$217353	$687570
2004	0.68	$147978	$101071	$485591	$347957	$237659	$849895
2005	0.62	$173618	$107803	$552528	$426237	$264659	$1014228
2006	0.56	$206322	$116463	$618268	$532144	$300381	$1183785
2007	0.51	$248530	$127535	$683714	$677311	$347568	$1362142
2008	0.47	$303685	$141671	$749805	$878745	$409941	$1553383

Table 9.12: Discounted Costs and Total Discounted Costs in Cases A (Uses Microwave Link) and B (No Microwave Link)

TCP/IP traffic at the level 100 kbit/s at the moment, with a standard deviation of 100 kbit/s, and growing at the rate of 25 % per year (both the mean and the variance).

The link must be maintained with an availability of 99.9%.

It has the following alternative technical alternatives:

- a microwave link, with costs as indicated in Table 9.14,
- a leased line, with costs as indicated in Table 9.13, or
- an Internet tunnel, which costs k$10.00 per year to lease and maintain, and has a capacity of 100 kbit/s at the moment which is growing at 25% per year. The availability of the Internet tunnel is estimated as 99%.

You should use a discount rate of 10% to evaluate the various alternative combinations and decide on a plan which carries the traffic and meets the availability target while achieving lowest possible cost. Please note that the availability target should be achieved in such a way that when there are no failures, performance should be completely adequate and 99.9% of the time the transmission capacity is at least half of the offered traffic. □

capacity (kbit/s)	cost (k$/year)	availability
x	$x/100$	99.9%

Table 9.13: Leased line costs and availabilities

capacity	installation cost (k$)	availability
1 Mbit/s	50	99%
10 Mbit/s	100	99%
45 Mbit/s	200	99%

Table 9.14: Microwave link costs and availabilities

Example 9.5. Major Network Upgrade

Suppose you are the manager of a middle-ranking telecommunications company, with turnover of approximately $10,000,000,000.00 per year. the network under your management has an asset value, in your Balance Sheet, of $40,000,000,000.00. Three quarters of this is associated with access networks and of the remaining $10,000,000,000.00, three quarters is made up of ageing telecommunications transmission and switching equipment. The switching equipment, in particular, will need to be replaced at some stage in the next 10 years.

The question is not "If?", but "When?". Actually, not "When?", so much as "How much now, how much next year, and how much the year after that?" We will not concern ourselves at all with the access network. No doubt the access network will require consideration, and probably significant upgrades, but in this example we concentrate exclusively upon the core of the telecommunications network. Furthermore, we are *only* considering the way in which voice services are carried. The question of what technology should be used for carrying Internet services is considered, for the purposes of this example, as a separate matter. In this example we shall assume that either Internet networking technology does not change character radically over the next 5 years or, if it does change character, it does so by become cheaper and more efficient even more rapidly than what is currently available.

The issues which dictate the need for change and which need to be taken into account in the decisions to be made are the following.

Issues:

(i) traffic efficiency – an IP network is more efficient than a switched network; in particular, voice communication can easily be carried in 1/8th the bandwidth when carried as IP traffic rather than on a circuit-switched network;

(ii) equipment cost – IP hardware is a mass-produced consumer product, an off-the-shelf item, whereas the traditional telephone switching equipment is highly specialised – made to order; cost reductions in the order of 10 to 100 times, as measured in dollars per switched bit/s are to be expected;

(iii) maintenance – existing equipment is highly reliable and lasts for years with regular maintenance; new equipment may actually require more maintenance, simply because of the need to break new ground; however, replacement cost of individual components can be expected to follow the same pattern as production ocsts of this equipment – 10 to 100 times lower costs. For this reason, maintenance costs of an IP network must be expected to reduce at a faster rate than existing networks;

(iv) new services – an all IP network has the potential to provide a great many new services without the need for expensive specialised hardware;

(v) transition – the transition to an all-IP network will inevitably lead to unforseen costs and unforseen problems; actually, there are also some forseeable additional costs which apply only during the transition, namely the cost of gateways which connect the voice over IP traffic with existing traffic. The cost of these gateways will rise steadily as voice over IP traffic increases, reaching a peak at the half-way point; these transition costs will work against the introduction of voice over IP;

(vi) non-IP technology – it is possible that at some stage in the near-term future a technology might arise which revises dramatically our current views concerning the future of networking. In an extreme case, IP networking might all-of-a-sudden become old-hat. In such a case, an enormous expenditure on switching to an IP network for the purpose of cost advantages which take ten years to prove themselves might turn out to be a dramatic mistake;

(vii) revenue – revenue per transmitted and switched bit/s will inevitably fall. Customers expect the cost advantages of new technology to be passed on to them and competition between major telecommunication carriers is now sufficiently intense to ensure that this happens. If any perception of a deal between carriers is sensed, government intervention is likely to re-establish pressure on telecommunication prices. It would be reasonable to expect real price reductions of in the order of 10% per annum as measured in cents per end-to-end bit/s.

Let us now revisit these issues, trying all the while to arrive at some answers to the problems we face.

Issues Revisited:

(a) Considering the last issue, (vi), first, it is clear that we should *invest now in research* into future possible networking technology. The brief of these researchers should be to lay aside any pro-IP (or anti-IP, pro-ATM, ...) bias or prejudice that they may be prey to and try to envision a world of high speed networking that might

follow *after* a world network constructed using today's technology. Technology which needs to be considered for use in such a network naturally includes wireless transmission, optical fiber to the home, optical switching, and optical computing. This research should be undertaken for the purpose of informing investment decision-making, and therefore it is essential that the researchers have a good working relationship with whoever it is that makes these decisions.

The concept that investment in *today's* new technology should be moderated on account of the possibility that this investment could be invalidated by the advent of new technology or changes in costs should be adopted as a matter of over-riding principle. This just makes financial good sense, and is probably best stated in terms which apply universally:

> *Postpone changes in investment strategy until the benefits of the proposed investment will be significantly compromised by further delay.*

(b) Now let us consider Issue (i). Is efficiency really important? The actual switching and transmission costs are only a small fraction of the cost of providing services. In reality, the major contributors to the provision of service are: servicing capital (i.e. gaining an adequate return on the huge nominal cost of the existing network), maintenance, and overheads.

However, we have to prepare for the advent of two contingencies: the first of these is the arrival of a competitor without the huge sunken cost of an existing network, high maintenance costs, and heavy overheads that burden existing telecommunications companies; such a competitor could have a huge impact on profitability; to protect against this contingency, we need to steadily reduce the all three of these seemingly perennial burdens relentlessly and without limit.

The second contingency that we should be wary of is the arrival of a new service that requires very large quantities of bandwidth. There is every reason to believe that this will happen. It has already happened, with the arrival of the world-wide-web. This particular arrival is still expanding its impact on networks. There is, in addition, the possibility that a service with a different character, even more dramatic in its impact, could add to the accelaration of growth of Internet traffic. If this happens, our networks could be quickly projected into a new operating region where traffic related costs do begin to form a major component of our costs. Even at present growth levels, networks are expanding sufficiently rapidly that traffic efficiency is of more importance all the time.

For both of these reasons, efficiency of our networks should remain a focus of attention. The technology with the lowest cost per switched and transmitted bit/s should be chosen and cultivated. This cost per switched bit/s should be assessed in a manner which includes capital expenditure *and maintenance* expenditure. In many cases, maintenance costs may dominate capital costs, although if this is the case, it increases the pressure to seek more cost effective maintenance procedures.

A Scenario

The difficulty we face in this example is primarily caused by uncertainty concerning the future. However, there are certain aspects of the future that are actually clear:

1. If voice over IP terminal devices become sufficiently cheap and ISP's are able to provide sufficient end-to-end performance, for a voice-class of IP traffic, that voice over IP traffic provides as good or better reliability, loss, and delay as the telephone network, then users will switch to it in large numbers;

2. Voice over IP terminal devices will become nearly as cheap as telephones within 3-5 years;

3. The Internet at large will be capable of providing voice-grade service for voice-class IP traffic over a great proportion of paths within 3-5 years; performance weaknesses for some paths might persist for a considerable time; (See §5.3.3 for further justification of this item);

There may be some controversy concerning the timing of these developments, but probably not concerning their eventual occurrence. It follows that the transition to a voice over IP future can occur without any alteration or upgrading of the existing telephone network. The traffic of the entire telephone networks of the world can probably

be absorbed into the Internet under the general heading of *normal growth* without disrupting unduly the gradual, or not so gradual, expansion of this network.

From this point of view, the task facing a telecommunications company can be broken down into two separate tasks:

Internet: It is important to incorporate the necessary improvements to the part of the Internet managed by this company so that item 3 will hold for our customers. In particular, the DiffServ architecture or something with similar objectives should be incorporated into our Internet services at the earliest opportunity. It is crucial, in order to retain customers, that our Internet services can be justifiably marketed as providing the best available service and the *appropriate* quality of service for each specialised class of service that we wish to cater for;

Telephony: The telephone network may well become little more than a vestige of its past glory in a relatively short period of time. The best items should be targetted for delivery to museums now. Retraining of existing staff should be initiated as a matter of urgency. It will probably be necessary to encourage late adopters to switch to the new way of handling voice over IP simply because in the not too distant future, the cost of maintaining a separate telephone network will be a burden we don't wish to bear.

Note 1: We have been able to formulate our decisions and our plans without reference to any quantities at all. This is not unusual and should come as no surprise.

Note 2: Something which might be considered is the setting up of a physically separate *private Internet*, just for our own customers to communicate with each other, and thereby obtain greater certainty of good reliability, loss performance and delay performance. However, the very significant advantages of large traffic aggregates strongly suggests that physical separation of one class of traffic from the main body of Internet traffic would only increase the cost of guaranteeing performance.

Note 3: Privacy of communication is a service which could conceivably be provided to voice over IP users in the future. In fact, there is every likelihood that this service will be provided by the terminal equipment. If such a service were to be provided, as an optional extra, it would most likely be provided "nearly" end-to-end by equipment installed at points in the network close to the terminals. Thus, the need for this type of service is largely independent of other issues and, in particular, has no impact on our strategy for switching to an IP-dominated network future.

□

9.4 Design for Service Protection

An interesting question arises in the design of networks for high reliability. We have already seen in Chapter 2 that high reliability networks can be created by using rings, and in Subsection 2.4.1 we discovered a rough rule of thumb for what the diameter of these rings should be. But there is still a question about what *capacity* these rings should have.

The capacity should be determined by the requirement that when a *single failure* occurs, the spare capacity in the network should be sufficient to carry the traffic which has to be redirected [5, 6].

The use of any network in which paths are longer than one hop has the potential to increase path length, and therefore increase cost. However, because transmission capacity is so dramatically cheaper in very large bundles, it makes a lot of sense to tolerate the awful efficiency of a network design with long paths if it can help to get more use out of the very large systems which are currently being installed.

Assuming completely homogeneous distribution of traffic over all *Origin Destination pairs* (O-D pairs) , the average path length in a ring network with n nodes is

$$A_L = \begin{cases} \frac{2\times1+2\times2+\ldots2\times(n/2-1)+n/2}{n-1} = \frac{n^2}{4(n-1)}, & n \text{ even}, \\ \frac{2\times1+2\times2+\ldots2\times((n+1)/2-1)}{n-1} = \frac{n+1}{4}, & n \text{ odd}. \end{cases}$$

and the total number of OD pairs is $\frac{n(n-1)}{2}$ so the average load on a link (measured in multiples of the load *to or from* an arbitrary O-D pair) is

$$\frac{A_L \times \frac{n(n-1)}{2}}{n} = \begin{cases} \frac{n^2}{8}, & n \text{ even}, \\ \frac{(n-1)(n+1)}{8}, & n \text{ odd}. \end{cases}$$

Clearly, there is a reasonably low limit on how large rings should be allowed to become since, basically, their efficiency is proportional to $\frac{1}{n^2}$, for large n. The cost advantage of using large modules of capacity rather than small ones over-rides other considerations at least until ring networks exceed 6 links and nodes by a significant degree.

Now let us consider how reconfiguration traffic might be accommodated. Suppose a failure occurs. We want to be able to reconfigure the traffic flowing through the failed link to pass along alternate paths. Since we are assuming that the load between every O-D pair is the same, we can pick an arbitrary O-D pair and see how we have to reconfigure the traffic to see the general picture.

Let us assume that the link from E to D in Figure 9.10 has failed. There is only one way to reconfigure the traffic – it has to go around the ring in the opposite direction. For example, the traffic starting at E and going to D will now have to go via F, A, B, and C. This will increase the load on every other link in the ring. On the other hand, traffic going through the link from E to D which starts at A and finishes at D (let's assume that half of this traffic normally goes via F and the other half goes via B), will be reconfigured to go through B and C. The reconfiguration load in this case is added to the links AB, BC, and CD, but does not affect FA and EF.

On reflection, there is one link which plays a part in every single item of reconfigured traffic: the link from A to B. It follows that in order for the ring network to be able to carry all the reconfigured traffic, the link from A to B must have *twice* the capacity that it would have in order to be able to only carry the original traffic.

Since the traffic is homogeneous and every link is exactly the same as every other, we can conclude that for the ring to be able to carry reconfiguration traffic as well as the base load, its capacity must be doubled.

Any degree of inhomogeneity of traffic will increase the capacity levels required in order to completely protect against a single link failure. Consider the extreme example where only one O-D pair has any traffic at all. In this case the minimum capacity on the ring to carry the base load is provided by equipping only this one link, whereas in order to protect against a single link failure, all the other links need to have their capacity brought up to the same level. This represents an n-fold increase of capacity.

If we consider networks with *more than one ring*, as in Figure 9.11, the situation becomes a little more attractive. Suppose, for example, that the network is completely symmetric, the traffic is homogeneous and consequently, *every link is in two rings*. Under these circumstances, half the reconfiguration traffic from any link which fails can be carried on *one* ring that it is part of, and the rest can be carried on the *other* ring. Hence, the total capacity required on all links, under these assumptions, to protect against the failure of any *single link*, is an extra 50%, throughout the network.

Again, any departure from homogeneity will mean that *more capacity* is required to protect against all single link failures. (On the other hand, in a network with inhomogeneous traffic which is protected against all single link failures, there may be some failure situations where two links fail which and yet the network is capable of carrying all the failure affected traffic).

Having to install an extra 50% of capacity to protect against failures is still a steep price to pay. In sufficiently large networks an additional consideration works in our favour.

First of all, in a large network, an increasing proportion of traffic which passes through a ring is *transit traffic*, that is to say it started elsewhere and it is going elsewhere. This traffic can be redirected, when a failure occurs, to pass through a variety of different rings. More important, though, is the fact that in a large network the traffic streams passing through each individual link come from and go to a great variety of locations, and hence, when this traffic has to be reconfigured after a failure it should be easy to spread this traffic over considerable diversity of paths. It might even be useful to reconfigure traffic which is initially local to a certain ring to traverse a much longer route which passes off that ring, if there is insufficient capacity on the other links of the ring.

In a sufficiently large network, the additional capacity required to protect against any single link failure will probably reduced to 10% or less. However, for such networks, the likelihood of several links having failed at once will become significant and protection against more complex failure conditions needs to be considered.

What is a reasonable level of spare capacity to build in to a network? In the case of a small campus network, probably we should have twice or more than twice the capacity needed, in most places. The additional cost is unlikely to be prohibitive and planning for growth will tend to force high levels of available bandwidth anyway. For large networks, again, methods for putting huge bit rates down optical fibers are becoming available [2], and growth rates are expected to be high for the foreseeable future, so it also makes sense to have quite significant quantities of un-utilized bandwidth on hand.

Example 9.6. Service Protection in a Campus Network

Figure 9.9: A Ring Network

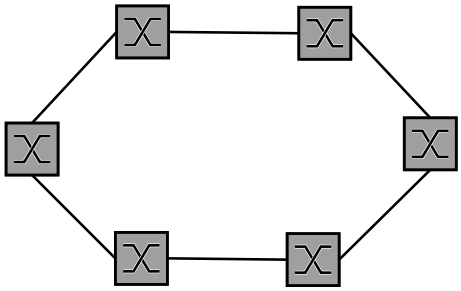


Figure 9.10: A Ring Network (labelled)

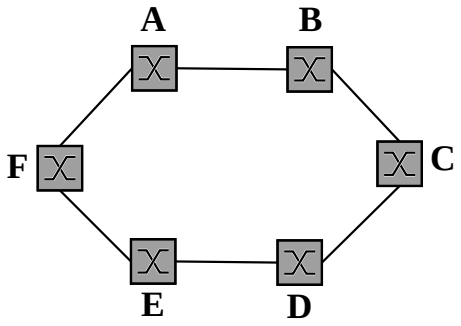
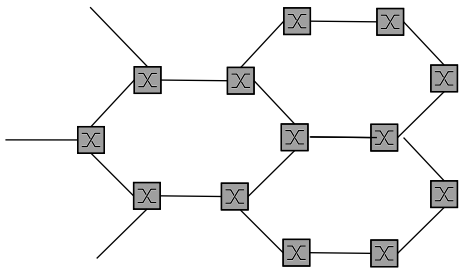


Figure 9.11: A Ring Network with many rings



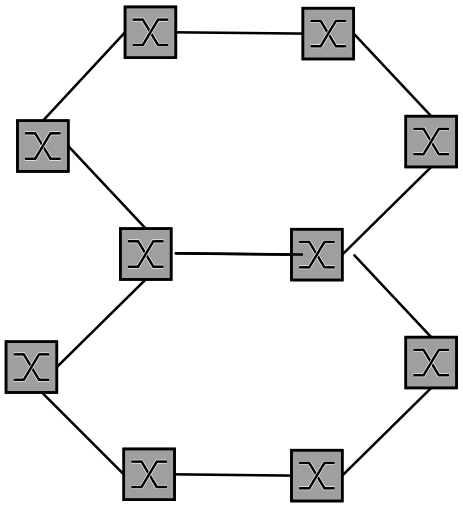
A campus network is perhaps the smallest of our examples which requires explicit attention to the issue of service protection. Campus networks are usually, nowadays at any rate, built around a central ring network made up of optical fibers. These optical fibers connect together a series of switches which are capable of reconfiguring the traffic on the ring. Typically switching is used in preference to routing at these locations because it is faster and cheaper. A small number of routers (e.g. two) should be sufficient for the entire campus, however, for special reasons it is likely that several dozen other routers exist in various laboratories and back rooms. After all, every Unix workstation has the capability built-in.

The switches may be ATM based, or Ethernet. At any rate, some mechanism for dynamically reconfiguring these switches is likely to be present, if only to lighten the load on the main router(s). □

Exercise 9.6. Design of a Ring Network

Consider the network depicted in Figure 9.12. Determine an accurate estimate of the appropriate capacity, as a multiple of the basic traffic demand across an O-D pair, for each link. Assume that traffic demand is completely symmetric and that the network should be able to carry all the traffic even if one link fails. □

Figure 9.12: A network with two Rings



9.5 Design Optimization

Forget for the moment that the shortest path is the obvious way to route traffic (and recall that obvious is not always best) and let us pose the following problem, in the context of network as depicted in Figure 9.13.

Example 9.7. Design as Optimization

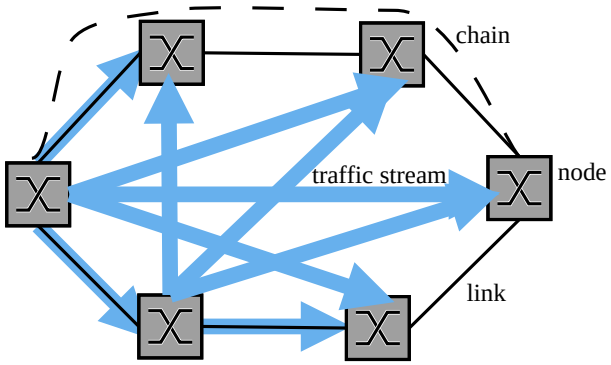
Find the routes (chains, paths) for each traffic stream such that

1. Traffic on each link $\leq$ link capacity;
2. Total link capacity is minimized.

Let's number the "feasible" chains $1, \ldots, N_C$, the nodes $1, \ldots, N_N$, the traffic streams $1, \ldots, N_S$, and the links $1, \ldots, N_L$. Let $\iota_{\ell c}$ denote the link-chain incidence matrix, so

$$\iota_{\ell c} = \begin{cases} 1 & \text{chain } c \text{ contains link } \ell \\ 0 & \text{otherwise.} \end{cases} \tag{9.1}$$

Figure 9.13: A network with traffic streams and chains



and let γ_{cs} denote the chain-stream incidence matrix, so

$$\gamma_{cs} = \begin{cases} 1 & \text{stream } s \text{ can be carried on chain } c \\ 0 & \text{otherwise.} \end{cases} \tag{9.2}$$

Also, let κ_ℓ denote the capacity of link ℓ, T_s denote the traffic on traffic stream s and τ_c denote the traffic carried by chain c

In this notation, the constraints can now be written:

$$\sum_c \tau_c \gamma_{cs} = T_s, \qquad s = 1, \ldots, N_s,$$

(i.e. the chains carry all the traffic on the streams), and

$$\sum_c \iota_{\ell c} \tau_c \leq \kappa_\ell, \qquad \ell = 1, \ldots, N_L,$$

(i.e. links have sufficient capacity for the traffic supplied to them by the chains which use them), and the objective function is:

$$\text{Min} \quad \sum_\ell \kappa_\ell.$$

□

This basic network dimensioning problem is linear (both the objective function and the constraints are linear) and can be solved by choosing the shortest path for each traffic stream and adding up the traffic on each link.

However, there are situations where the problem statement should be modified. In particular, we usually install additional capacity, over and above the mean offered traffic, for several reasons:

(i) transmission capacity is modular (and comes in rather large modules);

(ii) the traffic is random and we should really provide link capacity $\mu + 2\sigma$ to carry traffic with mean μ and standard deviation σ, not just μ;

(iii) traffic is usually growing at such a rate that we must install additional capacity to anticipate the future need;

(iv) as in Section 9.4, we may wish to install additional capacity to allow for failures.

Depending on the network layer under consideration, different issues among (i)–(iv) are relevant. In each case, if one or more of these issues are taken into account, the formulation of the design optimization problem can be modified to reflect a more realistic statement of the real optimization problem, although in some cases the resulting problem may turn out to be so difficult to solve, that it is more useful to use a heuristic method.

Example 9.8. Design Optimization taking into account traffic variation

A fairly classical variation on the design optimization problem is to take into account the extra capacity required because traffic is random. To do this we need to model every traffic stream, and the traffic on each chain, by means of *two* parameters: mean and variance. Fortunately by assuming independence of the traffic on different traffic streams (not *such* a bad assumption), we can treat variance as an additive descriptor, so that if two traffic streams with mean and variance μ_1, σ_1^2, μ_2, σ_2^2 are both carried on the same chain, the resulting traffic will have mean $\mu_1+\mu_2$ and variance $\sigma_1^2+\sigma_2^2$. So far so good: the problem still seems to be linear. But this is where the problem starts. The link constraints now have to reflect the revised dimensioning rule.

Let us denote the standard deviation of the traffic on each traffic stream by σ_s, $s=1,\ldots,N_S$, and the standard deviation of the traffic on each chain by s_c, $c=1,\ldots,N_C$. Then the constraints, in this problem, are:

$$\sum_c \tau_c \gamma_{cs} = T_s, \qquad s=1,\ldots,N_s,$$

(i.e. the chains carry all the traffic on the streams), and

$$s_c^2 = \frac{\tau_c}{T_s}\sigma_s^2, \qquad c\in\{c:\gamma_{cs}=1\}, \quad s=1,\ldots,N_s,$$

(i.e. chain variances are the same fraction of traffic stream variances as the chain means are of traffic stream means – a very simple *splitting formula*)

$$\sum_c \iota_{\ell c}\tau_c + 2\sqrt{\sum_c \iota_{\ell c} s_c^2} \le \kappa_\ell, \qquad \ell=1,\ldots,N_L,$$

(i.e. links have sufficient capacity for the traffic supplied to them by the chains which use them), and the objective function is:

$$\text{Min} \quad \sum_\ell \kappa_\ell.$$

□

Example 9.9. Design Optimization for Service Protection

Another variation of Example 9.7 can be formulated to take into account the spare capacity required to protect against failures.

This is a complex problem. However, conceptually we can simplify this problem a little by observing that it is as if we need to repeat the unprotected design problem many times: once for each failure condition that we need to protect against. Let us therefore enumerate these failure conditions: $1,\ldots,N_f$, where condition 1 is the *failure free* condition, and each other integer less than or equal to N_f corresponds to a network in which some failure condition holds, probably in which one specific link has failed, however in the *formulation* of this problem at any rate, any combination of failures can be entered into this list. Also, we are under no inherent obligation to list *all* single link failures. If we wish, at the risk of compromising our design, we can list just a sample of failure conditions.

Failure condition f will be characterised by the links which have failed under this condition, and in the statement of the constraints in the design problem, this information will be included by means of the matrix $\phi_{\ell f}$ defined so that

$$\phi_{\ell f} = \begin{cases} 1, & \text{if link } \ell \text{ is up under failure condition } f \\ 0, & \text{otherwise.} \end{cases}$$

We need to embellish the notation from Example 9.7 to allow for the greatly increased number of conditions we now need to impose on the design. The end-to-end traffic which needs to be carried around the network will be the same under all these conditions, however, the traffic flowing on each chain will be different depending on the failure condition, hence the *chain flows* will now be denoted by τ_c^f, $c=1,\ldots,N_c$, $f=1,\ldots,N_f$ and the routing matrix by $\gamma_{cs}^f, c=1,\ldots,N_c, s=1,\ldots,N_s, f=1,\ldots,N_f$.

The constraints can now be restated:

$$\sum_c \tau_c^f \gamma_{cs}^f = T_s, \qquad s=1,\ldots,N_s,\ f=1,\ldots,N_f,$$

(i.e. the chains carry all the traffic on the streams under all failure conditions), and

$$\sum_c \phi_{\ell f} \iota_{\ell c} \tau_c^f \leq \kappa_\ell, \qquad \ell = 1,\ldots,N_L,\ f = 1,\ldots,N_f,$$

(i.e. the links which are operational have sufficient capacity for the traffic supplied to them by the chains which use them under all failure conditions), and the objective function is unchanged:

$$\text{Min} \quad \sum_\ell \kappa_\ell.$$

□

Exercise 9.7. Taking modularity into account

Formulate a design optimization problem based on Example 9.7 but taking into account the fact that link capacities are actually provided in multiples of the basic OC1 rate, of 51.84 Mbit/s.

Hint: try to reformulate the problem by changing only the objective function. □

Exercise 9.8. Formulation of Optimal Design for Service Protection *and* a Traffic Buffer

Formulate a design optimization problem in which the additional complications of both Example 9.8 and Example 9.9 are taken into account, i.e. links are dimensioned to take into account the extra capacity required to allow for traffic variation and also the extra capacity required for protection against failures.

Two cases should be considered, Case (a) and Case (b). In Case (a), when a failure occurs the requirement for a buffer against traffic variation is dropped; in Case (b), the requirement for a buffer against traffic variation holds also under failure conditions. □

As more and more of the complications and complexities of the real world are incorporated into the basic design optimization problem it becomes harder and harder to solve by mathematical optimization. However, these complications do not always make the solution harder to find by other means.

For example, consider Example 9.9 under the additional constraint that links can only be provided in very very large moduls. Perhaps, to emphasise the issue, it has been decided that the *only* link capacity to be used will be 1.6 Tbit/s, which is many time larger than the largest capacity which might be required. Under these circumstances, the optimal design is a single ring. We have thus reduced the problem to the travelling salesman problem (see §9.1.6). This problem is not necessarily that easy to solve, although there are heuristics which can solve very large problems.

However, in reality, for large networks, it is not sufficient to take into account the possibility of just one failure at a time and also we are seldom in the position of needing to design such a network from scratch. It is more likely that we will need to expand the capacity of part of a network, or to expand the geographical range of such a network. The real-world constraints in such cases are likely to restrict our choices so much that the appropriate choices are relatively obvious.

References

[1] Techfest. SONET / SDH technical summary. Internet Web Site. http://www.techfest.com/networking/wan/sonet.htm.

[2] Lucent Technologies. Lucent - product and services. Internet Web Site. `http://www.lucent.com/products`.

[3] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, 2001.

[4] Alan Weiss. *Data Structures and Algorithms*. Addison-Wesley, 1997.

[5] Sebastian Cwilich, Mei Deng, David J. Houck, and David F. Lynch. An lp-nased approach to restoration network design. In P. Key and D. Smith, editors, *Teletraffic Engineering in a Competitive World*, volume 3a of *Proceedings of the International Teletraffic Congress*, pages 633–644. International Teletraffic Congress, Elsevier, 1999.

[6] Meir Herzberg and Felix Shleifer. Optimization models for the design of bi-directional self-healing ring based networks. In P. Key and D. Smith, editors, *Teletraffic Engineering in a Competitive World*, volume 3a of *Proceedings of the International Teletraffic Congress*, pages 183–194. International Teletraffic Congress, Elsevier, 1999.

Chapter 10

Conclusion

We should now appreciate to some degree, the goals, issues, and occasional solutions in the analysis and design of modern communication networks.

The subject has changed dramatically over the past decade. More and more networks are being built. Networks are getting larger. The equipment in the networks is getting faster, cheaper, and dramatically more capable of doing the fundamental task – connecting computers together so that humans can use them to communicate with each other and with the various daemons and agents that now populate the networks. As a consequence there are more and more people getting down to the task of network design, which is the subject of this book.

One of the tasks of any author is to attempt to unify and simplify the subject matter. This author has attempted to put this task as the top priority, perhaps even exaggerating the potential to simplify and unify. The aim of this work has been to reduce everything to common sense, where necessary by recalling, bringing to mind, or creating the common sense required to achieve this result.

Hopefully, certain recurrent themes have been apparent. Let us now recall some of these themes:

(i) Performance of networks *is* an issue – the central issue – and the types of performance under consideration are: loss, delay, reliability, and security; real networks are always pitched to provide an acceptable, but not excessive, performance on each of these scales.

(ii) Design is minimization of cost subject to performance constraints.

(iii) Networks are made up of layers, and layers *also* should be included or not included as the case may be depending on whether or not they provide sufficient additional performance on one or other scale, for the cost of the equipment and maintenance of that layer.

(iv) Design, in practice, is a messy compromise dominated by financial considerations; in many cases, a budget, or a business plan, will need to be prepared and this document should show, ideally, that one path – one collection of decisions – appears to lead to a financially more attractive outcome than others, taking into account the cost of borrowing, the cost of various risks, and the ever changing levels of traffic; a simple and effective way to formulate this sort of argument is by means of the concept of *present value*, however in some cases even this much complexity is unnecessary (or confusing) in order to discern the right approach.

(v) Traffic is a useful mathematical abstraction for the service our networks provide – it can be envisioned as something like a flow of water, but it is random; this randomness of traffic cannot be ignored – it must be allowed for in the apportionment of equipment to the communication tasks we wish to carry out.

(vi) The Internet and its protocols, the entire collection of them which we can describe as the TCP/IP architecture, has succeeded where other networks and protocol suites have not, to provide an extremely attractive mass service – a service for as many customers as can afford the equipment required to join to the network. It is to be expected that the Internet will expand to provide voice services on a large scale in the near term future. The supremacy of the Internet and its protocols, and the failure of alternative networks and protocols in competition with the Internet, is a fact of life today which we regularly use to test the validity of any idea about networks.

Appendix A

Netml, A Language for Describing Networks and Traffic

A.1 Introduction

In order to be able to describe networks in a reasonably uniform way, and then supply such networks to algorithms, it is virtually essential to make use of a language for describing networks. The features such a language requires have been worked out previously [1], and are not particularly controversial.

At this point in time, it seems obvious that such a language should be based on SGML or XML.

We need the following entities in a language for networks: *nodes*, *links*, and *traffic streams*. Special cases of nodes might include: *hubs*, *switches*, and *routers*, and also, for more specialised purposes, *sources* and *sinks*; and special cases of links might include: *LAN*s, and *point-to-point links*.

There isn't a strong need to mandate a specific order of presentation for these entities in a document describing a network. A schema for such a language is presented in Figures A.1–A.8.

A.2 Nodes

The best way to learn the netml language is by example. An example is presented below, and many more are available at the netml web site, at `http://cs.sci.usq.edu.au/netml2`.

The nodes of a network are the objects used to represent switches, routers, hubs, servers, and hosts. Nodes have sub-elements to delineate their features, including: *name* (arbitrary text, unique amongst nodes), *xposition* (a number), *yposition* (another number), and *rtable*, for describing a routing table.

A.3 Links

The important sub-elements of the *link* element are *name* (arbitrary text, unique amongst links), *origin* (which should be the name of a node), *destination* (name of a different node), *capacity* (a number), *meanTraffic* (a number), and *stdevTraffic* (a number).

A.4 Traffic

The important sub-elements of the *traffic* element are *name* (arbitrary text, unique amongst traffic streams), *origin* (which should be the name of a node), *destination* (name of a different node), *mean* (a number) and *stdev* (a number).

```
<?xml version="1.0" encoding="UTF-8"?>

<schema xmlns="http://www.w3.org/2001/XMLSchema"
   targetNamespace="http://cs.sci.usq.edu.au/netml/network"
    xmlns:netml="http://cs.sci.usq.edu.au/netml/network" >

<!-- Description of Network -->

<element name="network">
   <complexType>
      <all>
         <element name="nodes" minOccurs="1" maxOccurs="1">
            <complexType>
               <sequence>
                  <element name="node" type="netml:nodeType"
                     minOccurs="0" maxOccurs="unbounded" />
               </sequence>
            </complexType>
         </element>
         <element name="links" minOccurs="1" maxOccurs="1">
            <complexType>
               <sequence>
                  <element name="link" type="netml:linkType"
                     minOccurs="0" maxOccurs="unbounded" />
               </sequence>
            </complexType>
         </element>
         <element name="trafficStreams" minOccurs="1" maxOccurs="1">
            <complexType>
               <sequence>
                  <element name="traffic" type="netml:trafficType"
                     minOccurs="0" maxOccurs="unbounded" />
               </sequence>
            </complexType>
         </element>
         <element name="displaySetting" type="netml:displaySettingType"
               minOccurs="0" maxOccurs="1" />
      </all>
   </complexType>
```

Figure A.1: netml schema, Part I

```
<!--Constraint to check link origin and destination node exists -->

   <key name="keyNode">
      <selector xpath="nodes/node" />
      <field xpath="name" />
      </key>
   <keyref name="refNodeOrig" refer="netml:keyNode">
      <selector xpath ="links/link" />
      <field xpath = "origin" />
   </keyref>
   <keyref name="refNodeDest" refer="netml:keyNode">
      <selector xpath ="links/link" />
      <field xpath = "destination" />
   </keyref>

</element>
```

Figure A.2: netml schema, Part II

A.5 Settings

In addition to the sub-elements already described, in connection with nodes, links and traffics, each of these elements may have sub-elements which define *settings* used to guide the appearance of the network when it is printed or converted into a graphic. Default values of these settings can also be included in the document, in an element called *displaySettings*. In many cases, it will not be necessary to individually adjust these settings for each node, link, or traffic stream individually.

Note that objects may be *hidden*, their *size*, and *shape* may be specified (in the case of nodes at any rate), a *label* may be specified (using the name of the element to be used as the label), and the size of the font used in the label.

A.6 Example

An example of a network, described by means of the netml languate, is presented in Figures A.9–A.15.

References

[1] R. G. Addie, K. H. Soh, and P. P. Sember. EDL – a data language for network design and analysis. *Australian Telecommunications Research*, 26(1), 1992.

```
<!-- Description of node -->

<complexType name="nodeType">
   <all>
      <element name="name" type="string" />
      <element name="xposition" type="decimal" />
      <element name="yposition" type="decimal" />
      <element name="type" type="string"
         minOccurs="0" maxOccurs="1" />
      <element name="nodeSetting"
            type="netml:nodeSettingType"
            minOccurs="0" maxOccurs="1" />
                <xsd:sequence minOccurs="1" maxOccurs="1">
                    <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
                        processContents="skip"/>
                </xsd:sequence>
   </all>
</complexType>



<!-- Description of link -->

<complexType name="linkType">
   <all>
      <element name="name" type="string" />
      <element name="origin" type="string" />
      <element name="destination" type="string" />
      <element name="capacity" type="decimal"
         minOccurs="0" maxOccurs="1" />
      <element name="unavailability" type="decimal"
         minOccurs="0" maxOccurs="1" />
      <element name="otherParameters" minOccurs="0" maxOccurs="1">
         <complexType>
            <sequence>
                  <any minOccurs="1" maxOccurs="unbounded"
                  processContents = "skip" />
            </sequence>
         </complexType>
      </element>
      <element name="linkSetting"
            type="netml:linkOrTrafficSettingType"
            minOccurs="0" maxOccurs="1" />
                <xsd:sequence minOccurs="1" maxOccurs="1">
                    <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
                        processContents="skip"/>
                </xsd:sequence>
   </all>
</complexType>
```

Figure A.3: netml schema, Part III

```
<!-- Description of traffic -->

<complexType name="trafficType">
   <all>
      <element name="name" type="string" />
      <element name="origin" type="string" />
      <element name="destination" type="string" />
      <element name="mean" type="decimal"
         minOccurs="0" maxOccurs="1"/>
      <element name="unavailability" type="decimal"
         minOccurs="0" maxOccurs="1"/>
      <element name="otherParameters" minOccurs="0" maxOccurs="1">
         <complexType>
            <sequence>
                  <any minOccurs="1" maxOccurs="unbounded"
                  processContents = "skip" />
            </sequence>
         </complexType>
      </element>
      <element name="trafficSetting"
            type="netml:linkOrTrafficSettingType"
            minOccurs="0" maxOccurs="1" />
                <xsd:sequence minOccurs="1" maxOccurs="1">
                    <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
                        processContents="skip"/>
                </xsd:sequence>
   </all>
</complexType>
```

Figure A.4: netml schema, Part IV

```
<!-- Global settings for display of node and link or traffic -->

<complexType name="displaySettingType">
   <sequence>
      <element name="nodeSetting"   type="netml:nodeSettingType"
            minOccurs="0" maxOccurs="1" />
      <element name="linkSetting"
            type="netml:linkOrTrafficSettingType"
            minOccurs="0" maxOccurs="1" />
      <element name="trafficSetting"
            type="netml:linkOrTrafficSettingType"
            minOccurs="0" maxOccurs="1" />
   </sequence>
</complexType>
```

Figure A.5: netml schema, Part V

```
<!-- Settings for display of node -->

<complexType name="nodeSettingType">
   <all>
      <element name="shape" minOccurs="0" maxOccurs="1">
         <complexType>
            <simpleContent>
               <extension base="netml:shapeType">
                  <attribute name="priority"
                     type="nonNegativeInteger" />
               </extension>
            </simpleContent>
         </complexType>
      </element>
      <element name="size" minOccurs="0" maxOccurs="1">
         <complexType>
            <simpleContent>
               <extension base="nonNegativeInteger">
                  <attribute name="priority"
                     type="nonNegativeInteger" />
               </extension>
            </simpleContent>
         </complexType>
      </element>
      <element name="colour" minOccurs="0" maxOccurs="1">
         <complexType>
            <simpleContent>
               <extension base="string">
                  <attribute name="priority"
                     type="nonNegativeInteger" />
               </extension>
            </simpleContent>
         </complexType>
      </element>
   </all>
</complexType>
```

Figure A.6: netml schema, Part VI

```
<!-- Settings for display of link or traffic -->

<complexType name="linkOrTrafficSettingType">
   <all>
      <element name="hidden" minOccurs="0" maxOccurs="1">
         <complexType>
            <simpleContent>
               <extension base="netml:yesNoType">
                  <attribute name="priority"
                     type="nonNegativeInteger" />
               </extension>
            </simpleContent>
         </complexType>
      </element>
      <element name="label" minOccurs="0" maxOccurs="1">
         <complexType>
            <simpleContent>
               <extension base="string">
                  <attribute name="priority"
                     type="nonNegativeInteger" />
               </extension>
            </simpleContent>
         </complexType>
      </element>
      <element name="width" minOccurs="0" maxOccurs="1">
         <complexType>
            <simpleContent>
               <extension base="positiveInteger">
                  <attribute name="priority"
                     type="nonNegativeInteger" />
               </extension>
            </simpleContent>
         </complexType>
      </element>
      <element name="colour" minOccurs="0" maxOccurs="1">
         <complexType>
            <simpleContent>
               <extension base="string">
                  <attribute name="priority"
                     type="nonNegativeInteger" />
               </extension>
            </simpleContent>
         </complexType>
      </element>
   </all>
</complexType>
```

Figure A.7: netml schema, Part VII

```
<!-- global type for yes or no -->

<simpleType name="yesNoType">
   <restriction base="string">
      <enumeration value="yes" />
      <enumeration value="no" />
   </restriction>
</simpleType>


<!-- global type for shape -->

<simpleType name="shapeType">
   <restriction base="string">
      <enumeration value="box" />
      <enumeration value="circle" />
   </restriction>
</simpleType>

</schema>
```

Figure A.8: netml schema, Part VIII

```
<?xml version="1.0" encoding="UTF-8"?>
<nml:network xmlns:nml="http://www.sci.usq.edu.au/staff/addie/netml">
  <nodes>
    <node>
      <name>6</name>
      <xposition>98</xposition>
      <yposition>90</yposition>
    </node>
    <node>
      <name>5</name>
      <xposition>184</xposition>
      <yposition>77</yposition>
    </node>
    <node>
      <name>4</name>
      <xposition>246</xposition>
      <yposition>116</yposition>
    </node>
    <node>
      <name>7</name>
      <xposition>87</xposition>
      <yposition>187</yposition>
    </node>
    <node>
      <name>8</name>
      <xposition>95</xposition>
      <yposition>261</yposition>
    </node>
    <node>
      <name>1</name>
      <xposition>170</xposition>
      <yposition>300</yposition>
    </node>
    <node>
      <name>2</name>
      <xposition>272</xposition>
      <yposition>257</yposition>
    </node>
    <node>
      <name>3</name>
      <xposition>321</xposition>
      <yposition>185</yposition>
    </node>
  </nodes>
```

Figure A.9: An example network, described by means of netml, Part I

```
<links>
  <link>
    <name>O0</name>
    <origin>7</origin>
    <destination>8</destination>
    <meanTraffic>122.00</meanTraffic>
    <stdevTraffic>0.00</stdevTraffic>
    <capacity>100.50</capacity>
  </link>
  <link>
    <name>O1</name>
    <origin>6</origin>
    <destination>7</destination>
    <meanTraffic>123.10</meanTraffic>
    <stdevTraffic>0.00</stdevTraffic>
    <capacity>104.10</capacity>
  </link>
  <link>
    <name>O2</name>
    <origin>6</origin>
    <destination>5</destination>
    <meanTraffic>109.20</meanTraffic>
    <stdevTraffic>0.00</stdevTraffic>
    <capacity>94.20</capacity>
  </link>
  <link>
    <name>O3</name>
    <origin>8</origin>
    <destination>1</destination>
    <meanTraffic>108.60</meanTraffic>
    <stdevTraffic>0.00</stdevTraffic>
    <capacity>87.30</capacity>
  </link>
  <link>
    <name>O4</name>
    <origin>2</origin>
    <destination>1</destination>
    <meanTraffic>83.10</meanTraffic>
    <stdevTraffic>0.00</stdevTraffic>
    <capacity>70.80</capacity>
  </link>
  <link>
    <name>O5</name>
    <origin>5</origin>
    <destination>4</destination>
    <meanTraffic>54.80</meanTraffic>
    <stdevTraffic>0.00</stdevTraffic>
    <capacity>51.80</capacity>
  </link>
  <link>
    <name>O22</name>
    <origin>3</origin>
    <destination>2</destination>
    <meanTraffic>36.90</meanTraffic>
    <stdevTraffic>0.00</stdevTraffic>
    <capacity>33.90</capacity>
  </link>
</links>
```

Figure A.10: An example network, described by means of netml, Part II

```
<trafficStreams>
  <traffic>
    <name>t1</name>
    <origin>1</origin>
    <destination>2</destination>
    <mean>6</mean>
  </traffic>
  <traffic>
    <name>t2</name>
    <origin>1</origin>
    <destination>3</destination>
    <mean>3.5</mean>
  </traffic>
  <traffic>
    <name>t3</name>
    <origin>1</origin>
    <destination>4</destination>
    <mean>9</mean>
  </traffic>
  <traffic>
    <name>t4</name>
    <origin>1</origin>
    <destination>5</destination>
    <mean>5</mean>
  </traffic>
  <traffic>
    <name>t5</name>
    <origin>1</origin>
    <destination>6</destination>
    <mean>8</mean>
  </traffic>
  <traffic>
    <name>t6</name>
    <origin>1</origin>
    <destination>7</destination>
    <mean>7</mean>
  </traffic>
  <traffic>
    <name>t7</name>
    <origin>1</origin>
    <destination>8</destination>
    <mean>6</mean>
  </traffic>
  <traffic>
    <name>t8</name>
    <origin>2</origin>
    <destination>3</destination>
    <mean>7.5</mean>
  </traffic>
  <traffic>
    <name>t9</name>
    <origin>2</origin>
    <destination>4</destination>
    <mean>9.3</mean>
  </traffic>
```

Figure A.11: An example network, described by means of netml, Part III

```
<traffic>
  <name>t10</name>
  <origin>2</origin>
  <destination>5</destination>
  <mean>25</mean>
</traffic>
<traffic>
  <name>t11</name>
  <origin>2</origin>
  <destination>6</destination>
  <mean>6.2</mean>
</traffic>
<traffic>
  <name>t12</name>
  <origin>2</origin>
  <destination>7</destination>
  <mean>3.2</mean>
</traffic>
<traffic>
  <name>t13</name>
  <origin>2</origin>
  <destination>8</destination>
  <mean>4</mean>
</traffic>
<traffic>
  <name>t14</name>
  <origin>3</origin>
  <destination>4</destination>
  <mean>3</mean>
</traffic>
<traffic>
  <name>t15</name>
  <origin>3</origin>
  <destination>5</destination>
  <mean>12</mean>
</traffic>
```

Figure A.12: An example network, described by means of netml, Part IV

```
<traffic>
  <name>t16</name>
  <origin>3</origin>
  <destination>6</destination>
  <mean>4</mean>
</traffic>
<traffic>
  <name>t17</name>
  <origin>3</origin>
  <destination>7</destination>
  <mean>2.5</mean>
</traffic>
<traffic>
  <name>t18</name>
  <origin>3</origin>
  <destination>8</destination>
  <mean>4.4</mean>
</traffic>
<traffic>
  <name>t19</name>
  <origin>4</origin>
  <destination>5</destination>
  <mean>6.5</mean>
</traffic>
<traffic>
  <name>t20</name>
  <origin>4</origin>
  <destination>6</destination>
  <mean>12</mean>
</traffic>
<traffic>
  <name>t21</name>
  <origin>4</origin>
  <destination>7</destination>
  <mean>3</mean>
</traffic>
```

Figure A.13: An example network, described by means of netml, Part V

```
  <traffic>
    <name>t22</name>
    <origin>4</origin>
    <destination>8</destination>
    <mean>12</mean>
  </traffic>
  <traffic>
    <name>t23</name>
    <origin>5</origin>
    <destination>6</destination>
    <mean>7.3</mean>
  </traffic>
  <traffic>
    <name>t24</name>
    <origin>5</origin>
    <destination>7</destination>
    <mean>7.8</mean>
  </traffic>
  <traffic>
    <name>t25</name>
    <origin>5</origin>
    <destination>8</destination>
    <mean>3.8</mean>
  </traffic>
  <traffic>
    <name>t26</name>
    <origin>6</origin>
    <destination>7</destination>
    <mean>7</mean>
  </traffic>
  <traffic>
    <name>t27</name>
    <origin>6</origin>
    <destination>8</destination>
    <mean>8</mean>
  </traffic>
  <traffic>
    <name>t28</name>
    <origin>7</origin>
    <destination>8</destination>
    <mean>4</mean>
  </traffic>
</trafficStreams>
```

Figure A.14: An example network, described by means of netml, Part VI

```
<displaySetting>
  <nodeSetting>
    <shape priority="2">circle</shape>
    <size priority="0">20</size>
    <labelfont>8</labelfont>
    <colour priority="0">green</colour>
  </nodeSetting>
  <linkSetting>
    <hidden priority="3">no</hidden>
    <label priority="3">meanTraffic</label>
    <labelfont>8</labelfont>
    <width priority="3">2</width>
    <colour priority="2">blue</colour>
  </linkSetting>
  <trafficSetting>
    <hidden priority="0">yes</hidden>
    <label priority="3">unavailability</label>
    <width priority="3">8</width>
    <colour priority="1">green</colour>
  </trafficSetting>
</displaySetting>
```

Figure A.15: An example network, described by means of netml, Part VII

Index