INNOVATIVE TECHNOLOGY SERIES

INFORMATION SYSTEMS AND NETWORKS

Communicating Systems & Networks: Traffic & Performance

Georges Fiche &
Gérard Hébuterne

HPS HERMES PENTON SCIENCE

an imprint of KOGAN PAGE SCIENCE

Communicating Systems & Networks: Traffic & Performance

This page intentionally left blank

INNOVATIVE TECHNOLOGY SERIES
INFORMATION SYSTEMS AND NETWORKS

Communicating Systems & Networks: Traffic & Performance

Georges Fiche &
Gérard Hébuterne

KOGAN PAGE SCIENCE

London and Sterling, VA

First published in France in 2003 by Hermes Science entitled *'Trafic et performances des réseaux de télécoms'*
First published in Great Britain and the United States in 2004 by Kogan Page Science, an imprint of Kogan Page Limited

120 Pentonville Road
London N1 9JN
UK
www.koganpagescience.com

22883 Quicksilver Drive
Sterling VA 20166-2012
USA

ISBN 1 9039 9635 X

British Library Cataloguing-in-Publication Data

A CIP record for this book is available from the British Library.

Library of Congress Cataloging-in-Publication Data

Fiche, Georges.
[Trafic et performances des reseaux de telecoms. English]
Systems and communicating networks : traffic and performance / Georges Fiche and Gerard Hâebuterne.
p. cm. -- (Innovative technology series. Information systems and networks)
ISBN 1-903996-35-X
1. Telecommunication--Traffic. 2. Computer networks. I. Hâebuterne, Gâerard. II. Title. III. Series.
TK5102.5F47 2004
621.382'1--dc22

2003024763

Typeset by Kogan Page
Printed and bound in Great Britain by Creative Print and Design (Wales), Ebbw Vale

Contents

Introduction

This book originates from the desire to perpetuate expertise in the field of performance evaluation, particularly for telecommunication systems and networks. It is obviously impossible to translate into a single work all the facets of an activity which encompasses many systems and a great variety of domains such as standardization, modelling, measurement, fields trials, observations etc. However, it rapidly becomes evident that performance study through its different aspects is the expression of a real and unique discipline: *performance engineering*. So it is worth writing a book whose contents is, as much as possible, the synthesis of both the theoretical and the technical knowledge which are the basis for good practice in this field.

In this respect this work aims to be both a tool for education in performance engineering, and a guide to implementing performance activity, both in the research laboratory and in industrial environment.

Research and industrial work are both demanding. The performance engineer will have to juggle with equations, as well as with equipment in the lab or in the field. His/her permanent search for efficiency, the necessity to use tractable approximations, and his/her natural trend to perform experimental measurements will not prevent him/her from mastering complex mathematical models. As a matter of fact, it is the complete mastering of the analytical tools together with their application to the whole set of system development phases (from design to operation), which will lead to maximum efficiency, by making possible the synthesis between theory and practice as required by market and industrial constraints.

Therefore, in this book, we will deal equally with elementary calculations, such as processor occupancy, the number of messages etc. as well as with more complex computations such as multiplexer dimensioning in the case of internet traffic. In the same way, we will use elementary probability calculations or classical Markov models, as well as complex methods such as Pollaczek's method for queueing systems evaluation. Lastly, again with concern for efficiency, we always shall keep in mind the actual conditions of application and particularly the order of magnitudes of the parameter values, which will allow great simplification of the models.

A tool for education and a companion hand book for performance engineering activity, this work is dedicated to the student who wishes to learn about communicating systems and networks, probability and queueing theory, as well as to the engineer and the researcher who desires to enlarge his/her competence domains to other fields, such as reliability, statistics, quality of service standardization, and methodology.

Several approached to this work are possible. Each chapter, relatively independent from the others, deals with a subject in a way as progressive as possible.

Chapters however are organized according to a logic whose motivation is again the desire to teach the basics of the job of performance engineering. The logic is as follows.

The first chapter presents the main characteristics and functions of major *telecommunication networks*. These are indeed the subject of our performance studies, whose objective is to evaluate their capacity to handle the traffic they are offered. In this respect, we then shall present in the same chapter the basic concepts of what is called *teletraffic*.

In the second chapter, we develop the generic aspects of *quality of service* (QoS), through an overview of the main performance parameters such as specified in the international *standardisation* organisms.

So, with these two chapters we have set the scene: the subject that we are studying (the telecommunication systems and networks), and the reason why we are interested in it (traffic handling and QoS). Now we have to deal with the means needed for our studies, firstly with the basic theoretical tools.

The third chapter presents the *theory of probability* and introduces analytical tools such as transform functions, which allow us to solve most of the probabilistic problems that we will face.

The fourth chapter presents the main *probability laws* which will be of continuous use in the rest of the book, in various domains such as statistic, reliability, queueing, etc.

The fifth chapter, as for probability theory, presents the theory of *statistics*. Indeed we often shall be concerned with estimation and its associated notion of risk.

The fundamental tools being thus acquired, we then look at their application to the different domains of performance. We now introduce performance evaluation techniques:

The sixth chapter presents *reliability* theory, and more generally what is called *dependability*. Here we look at the means to evaluate system performance in terms of availability, service continuity, maintenance load etc..

The seventh chapter presents *queueing theory*. Here, of course, the purpose is the evaluation of systems and network performance in terms of information transfer delays, response times, set-up times, etc.

The eighth chapter introduces *simulation* techniques, which, of course, are an essential complement to analytical studies and models.

The ninth chapter deals with concrete applications of these techniques, resulting into a set of typical *models* for telecommunication systems.

Finally, all these tools, techniques, and models being acquired, we will now be able to deal with the methodological aspects. Methodology is the means to organize efficiently any industrial activity, and is an essential condition to ensure its success, particularly in very competitive environments. In this respect the last chapter proposes a methodology for performance activity, that may apply to projects in industry as well as in the research laboratory.

The reader desirous of completing his/her knowledge in any domain will find, at the end of the book, a selection of references.

We would like to thank particularly Bernard Depouilly and Jean-Claude Pennanec'h, and all the careful readers: Daniel Lardy, Daniel Le Corguillé, Claude Le Palud, Christophe Perraudeau, Thierry Sabourin.

This book has received strong support from Alcatel, INT (*Institut national des télécommunications*), and the "*Collection technique et scientifique des télécommunications*". We thank them warmly.

Georges Fiche and Gérard Hébuterne

This page intentionally left blank

1

Telecommunication Networks and Traffic

The subject of this book, performance evaluation, may be considered as a field of the vast discipline known as *teletraffic*, whose purpose is to determine rules which optimise the use of the *network equipment* that handles the traffic requested by users. The book is not therefore strictly concerned with traffic management techniques or equipment optimisation techniques. But, by presenting the techniques used to evaluate performances, we aim to establish the bases for the evaluation of the capacities of traffic handling equipments and for network dimensioning.

The basic function of a telecommunications network is to connect users to other users via their terminals, and to connect users to servers, or terminals to other terminals. The network sets up a connection between two or more terminals by making use of their source and destination addresses.

Behind this very general connection concept are a number of very different realities, bearing in mind the great variety of telecommunications networks, as we will see later. In a "circuit" type network, the term used is *connection*, a direct relationship established at a physical level. In an "IP" network, the term used is usually *session*, as there is normally no physical connection (even though the TCP operates in connected mode at the session level). Finally, the terminals connected will not only be fixed subscribers, but also mobile subscribers or servers.

But the fundamental functions remain the same. And throughout this book we will consider only the generic aspects of the world of telecommunications. Whatever the networks, we will be dealing with very general call and communication concepts. In doing so we will consider the concepts of arrival rate, duration, and resources engaged, whether in terms of circuits, memory or bit rate.

In this chapter we will introduce the main concepts relating to *networks* and *traffic,* which we will subsequently consider in more detail and manipulate.

Initially we will present the general characteristics of telecommunications networks, their structures and their functions. Despite ongoing technological changes, we will identify certain constant features. The performance evaluation techniques presented in this book are applied to these constant features, such as access, the transport network, and the control network.

Secondly, we will introduce the basic concepts of traffic, i.e. relating to the volume of data processed and transported in a telecommunications network. Evaluating the performance of an equipment or a set of equipments is only meaningful if the evaluation is related to a given type and level of demand. These essential concepts are the foundation of *teletraffic* theory.

1.1. Network concepts

A network can be described and analysed at several levels: architecture, technologies and services. We will see that the convergence of networks and the integration of services mean that these distinctions are becoming less and less clear cut. To describe this, we will look in turn at the main structures of networks from the topology viewpoint, and then set out the two main types of switching technology (circuit and packet), drawing distinctions between a certain number of networks from a functional viewpoint (telephone, data), before finally presenting their convergence in the "next" generation networks, the NGN.

1.1.1. *Network architectures*

A network, whatever its type, consists essentially of *nodes*, which route and switch data and control the network, and *links* that interconnect the nodes and transport the data. The arrangement of the nodes, and their physical and logical organisation form the network architecture. A distinction must be drawn between *physical* architecture (the locations of nodes and links, and how the cable connections are made) and *logical* architecture, which describes the routing of the data and the routing rules. The same physical infrastructure can be shared between different networks (the telephone network, a data network, etc.), with different logical architectures. Although architectures may vary considerably, certain basic structures and components can be identified.

1.1.1.1. *Network components*

As networks evolve, there is an increasing tendency to distinguish between the three main parts of a network: the core network, the access network, and the customer premises equipment.

The core network: Also known as the "backbone", the core network is the infrastructure that enables the interconnection of all users. Shared between all the subscribers, it offers the possibility of high-speed data transfer over long distances.

The access network (AN): This part of the network enables the connection of the user to the core network. Access networks are shared by a limited number of subscribers.

The customer premises equipment (CPE): This is the equipment that the user has on his/her premises (telephone, computer, local network, etc.).

From the user's viewpoint there are two main components of a network: *network services* and *network management.*

These are the two main functions that generate *traffic* in the physical elements of networks.

Services

The primary function of the network is of course to support services such as the telephone service, videoconferencing, and the transfer of images, e-mails and data files... The variety of services is very considerable, and their requirements at transport level (bandwidth, real-time constraints) are very different. Several switching technologies (notably circuit switching and packet switching) have been developed to help meet these requirements.

Management

The second function associated with the network is the management function. A network is constantly changing. New subscribers always need to be connected, new equipment needs to be installed, and new services need to be introduced. It is also of course essential to ensure that the network is functioning correctly by means of maintenance operations and traffic and quality of service observation operations. Network extensions, upgrades and observations are usually carried out by an operator. They too are organised around equipment and functions that constitute a management and operation network.

1.1.1.2. *Network structures*

The star network

The network elements are interconnected through a central element or *central node*. This type of architecture is suitable for networks that connect a limited number of users. The star network topology is found in private networks (PABX, see later) and in the case of terminals connected to a central computer (business network). The reliability of this type of network is wholly dependent on that of the central node, and the failure of this node means that all the terminals are disabled. On the other hand, routing is simple, as are extensions of the number of terminals. Communication requires appropriate protocols such as CSMA (see later), to manage access conflicts between terminals. An example of a performance study will be given later in the book.

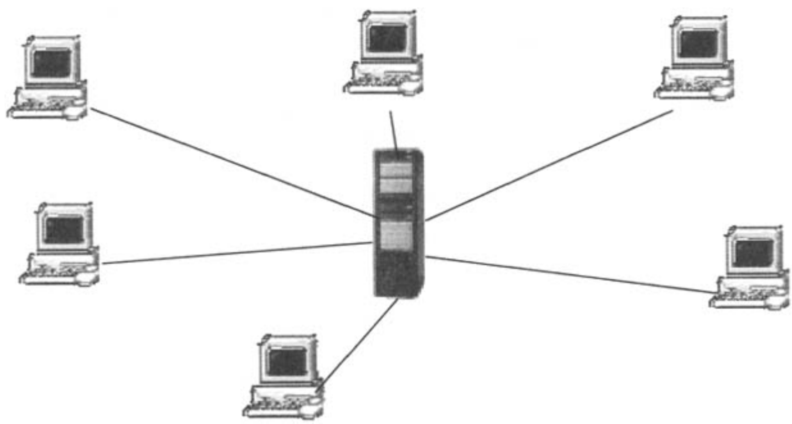


Figure 1.1. *Star network structure*

Meshed network

In a totally meshed network, each *node* is connected to the others by the same number of point-to-point links as there are relations to be established between nodes. A large number of links is needed for this type of topology – the number of relations grows proportionately to the square of the total number of nodes n (there are exactly $n(n-1)/2$ links). The great advantage of this type of network is its reliability: there is always the possibility of reaching another node, even in the event of a link breakdown, by going through an intermediate node. For sizing however, the traffic flow between nodes needs to be determined with considerable precision. This often leads to substantial oversizing.

In practice, networks are partially meshed, and the opening of a direct link depends both on optimisation and reliability criteria. Here again, precise knowledge of flows will enable optimum sizing.

Figure 1.2 shows an ATM or IP network of fully meshed *provider edge* nodes. Each edge node collects user *access gateway* (AGW) traffic, or concentrates and receives network VoIP (*voice over IP*) traffic, via an *IP gateway* (VoIP GW).

We could equally well have represented a network of computers or of IP routers.

This type of architecture is well suited for networks of limited size that must provide a very high quality of service (ease of routing, "*tunnelling*" possibility).

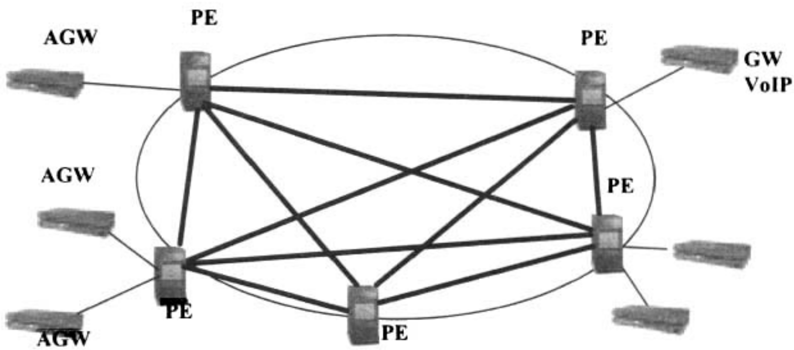


Figure 1.2. *Fully meshed network*

Ring network

In this network each of the n nodes is interconnected only to its two neighbours. The number of links is thus reduced to n. The network therefore forms a ring. To ensure the reliability of this type of network, the links may be duplicated individually, or the ring itself may be duplicated. Furthermore, the routing protocols can isolate a faulty node, and the network then becomes "*self-healing*".

This type of architecture is the basic structure for university data networks, metropolitan local networks and of course computer networks, using the well-known *token ring* technology (whose performances we will study). It is also used as the backbone of "regional hubs" of operator networks.

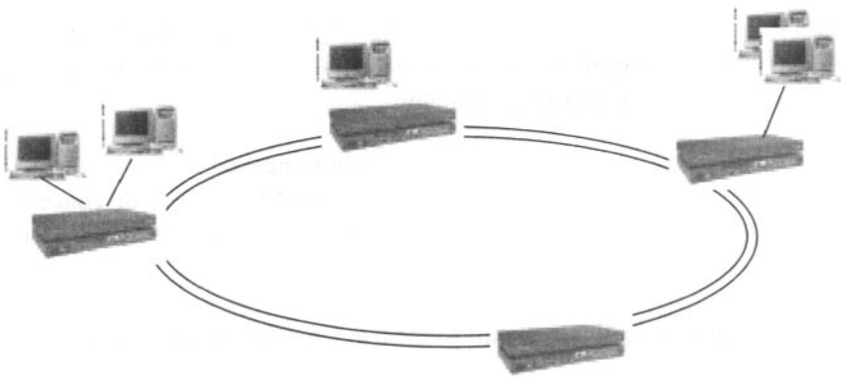


Figure 1.3. *Ring network*

Tree network

In a certain number of cases, node interdependences lead to the organisation of nodes into tree networks, in which each node has a responsibility (for routing, management, etc.) to the nodes linked to it. This architecture is used in this form (tree structure providing a multicast type service) or more often as a sub-network organisation structure.

Hierarchical structure

A network of this type is in fact composed of several subnetworks of different hierarchical levels. At the top level is the backbone used for the interconnection of the various lower-level networks (e.g. in telephony; the international network, consisting of international transit exchanges). At the next level are the national networks (e.g. with their main and secondary transit exchanges) and then come the access networks (connection centres or local exchanges, and subscriber connecting units).

Fixed and mobile telephone networks are typical examples of this structure. The subnetworks interconnected in this way (regional networks, urban networks) have a partial meshing structure. The tendency is to use full meshing at the upper levels of the hierarchy. The number of hierarchical levels depends on the size of the network (typically 3 in Europe and 4 in the USA).

Gateway structure

In a network of this type, which also consists of subnetworks, the concept of hierarchy is less important and that of cohabitation assumes greater importance. The subnetworks communicate with neighbouring subnetworks through *gateways*. The information circulates step by step. In both of these networks, transport reliability is ensured by the multiplicity of routes (access to several toll exchanges, access to several routers).

The internet network is a typical example of the gateway structure.

The two structures are shown in Figure 1.4.

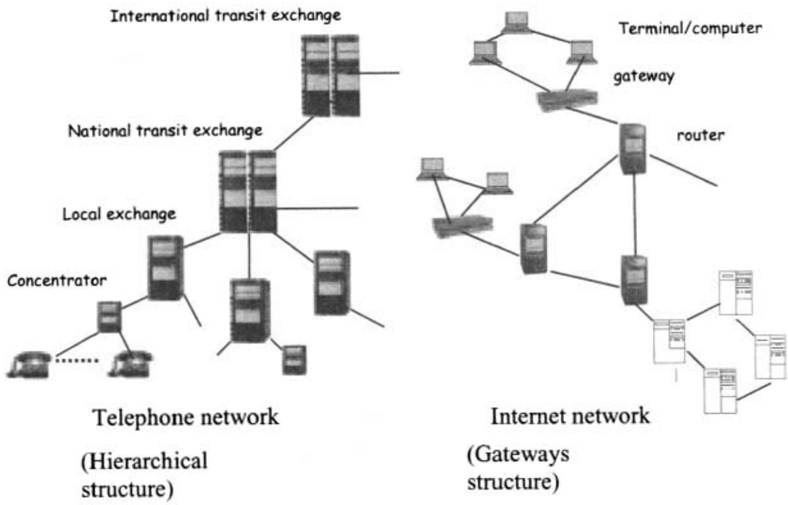


Figure 1.4. *Tree networks*

1.1.2. *Communication technologies: circuit and packet*

Obviously a network handles a great diversity of types of traffic sent by various sources. A wide range of methods has thus also been proposed to ensure an optimum communication service as appropriate for the various profiles. Amongst these methods two main switching techniques predominate: *circuit switching* and *packet switching*, of which a number of variants have been developed.

1.1.2.1. *Circuit switching*

The simplest solution to establish communication between point A and point B is to connect them by an electrical link. This is how the first telephony systems operated. The switches are essentially automatic devices with n inputs and n outputs, capable of linking one input to one output on request. A connection will be presented as a series of links that are reserved and of relations that are established in the switches between the links. The link must be established before any exchange can take place.

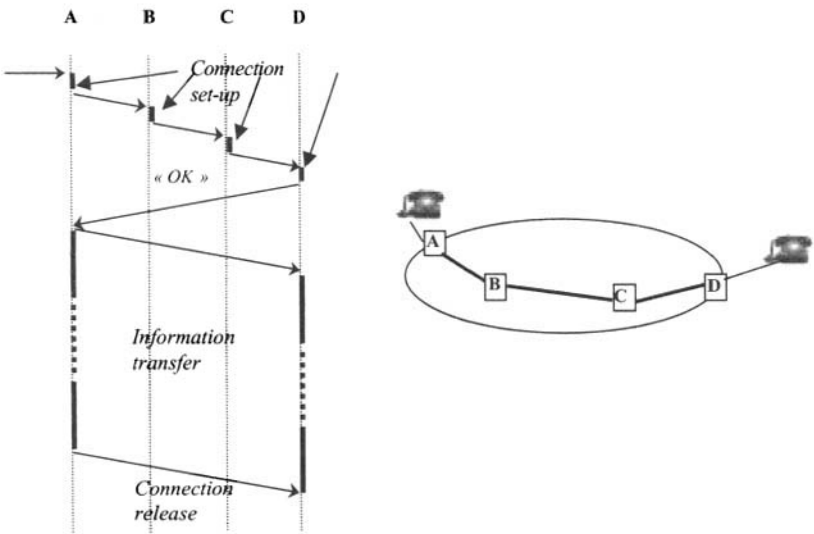


Figure 1.5. *Communication phases for circuit switching*

The essential characteristic of circuit mode is the reservation of a path and of resources, from end-to-end, for the whole duration of the call.

In point of fact, the connection of a real electrical circuit is only the simplest way of making the link. In most cases, a link with a high capacity will be shared between more modest flows. So for example PCM technology shares a 2 Mbit/s link between 32 circuits at 64 kbit/s (1.5 Mbit/s and 24 channels at 64 or 56 kbit/s in the USA).

Figure 1.5 shows the connection procedure. The phases of establishment through the network (shown schematically above by intermediate nodes B and C) are clearly indicated.

1.1.2.2. *Packet switching*

The information sent out by a computer equipment is intrinsically *bursty*, which is clearly visible for example in the case of an internet session. Both the terminal and the workstation produce, at their own rate (expressed in bit/s) blocks of information that are separated by silences that may last for several seconds. Accordingly, it would be uneconomical to establish a communication that would immobilise resources (transmission lines, switch paths). The solution used to enable efficient exploitation of the network's resources is the setting up of *packet switching*. In this way several communications can share the same resources.

In circuit switching, the communication is offered a bit rate d which is constant throughout the duration of the connection. Let us assume that the exchange takes place with a *burstiness* a (defined as the ratio of the instantaneous bit rate to the average rate). The allocated constant rate d is such that it enables the communication to transmit the bursts at peak rate without loss. It follows therefore that the effective rate of utilisation of the link is $1/a$. Bearing in mind that if a reaches 10 or more, which is frequently the case in data transmission, the cost-effectiveness of the service becomes problematic.

In packet switching, the information is split up by the originating equipment into blocks of moderate size (see Figure 1.6), termed *packets*, which are autonomous (i.e. capable of moving on the network thanks to a *header* that contains the destination address). The source sends its packets at its own rate, and the network multiplexes the packets from various origins in the same resources, to optimise their use (this is known as *statistical multiplexing*).

This splitting up enables a better use of the transmission resource than circuit switching, in which the transmission resources are allocated without sharing. On the other hand, the multiplexing of different connections on the same resources causes delays (and perhaps losses) which do not happen with circuit switching. This operation calls for careful sizing, and the setting up of congestion control mechanisms.

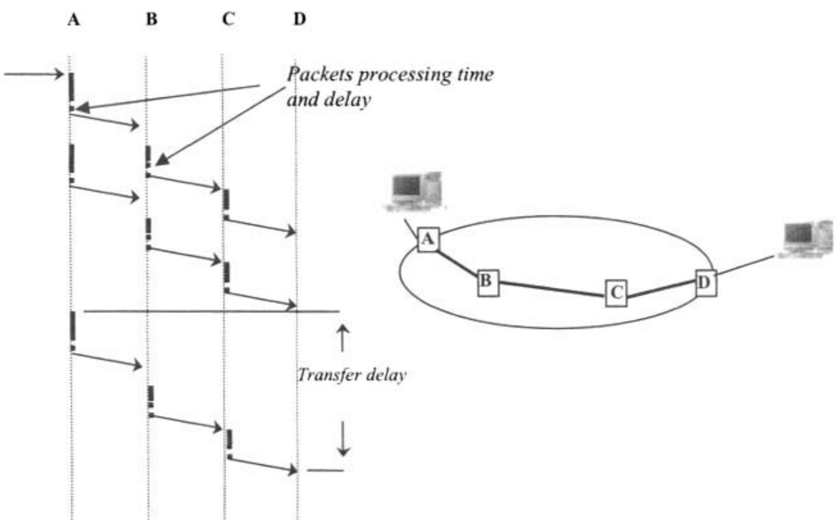


Figure 1.6. *A session in a packet-oriented network*

Finally it must be noted that in packet switching a distinction is drawn between two modes of operation (which are not to be confused with switching mode): *connection-oriented* mode and *connectionless* mode. In connection-oriented mode, a path is established (the virtual circuit). There is a prior exchange of initial signalling packets to reserve resources and establish the path. In connectionless mode, the routing decisions are taken on the basis of the header of each packet, at each node (for *routing* and *forwarding*, see below).

Connection-oriented mode has given rise to three main operating modes: *X25* (which corrects errors, guarantees packet sequencing and includes flow control), *frame relay* (no control at packet level), and ATM. *Asynchronous transfer mode* (ATM) switching is a packet technique in which the information unit is a packet of fixed length (53 bytes), the cell, with important control functions. The connection-oriented mode of ATM, associated with the fixed unit that is the cell, is used at maximum to offer guarantees on the available bandwidth, the maximum error rate, delays, etc., its disadvantage being the degree of complexity to which it gives rise.

The connectionless mode has been popularised mainly by IP (*internet protocol*). IP packets are of variable size: for example 20 (or 40) *header* bytes and a *payload* (useful information) of between 0 and 65,535 bytes. Basically an IP network does not guarantee quality of service, but instead works in "*best effort*" mode. IP tries in fact to transmit all the packets but without preventing desequencing, duplication, delays or loss. The necessity of reliable transport thus leads to the use of two connection modes for the same communication, at different levels. So with TCP (*transmission control protocol*) on IP, we have a connection mode at application level, on a mode that is connectionless at packet level. Conversely, with IP on ATM, we have a connectionless mode on a connection-oriented mode. We will look in detail a little later at the main communication protocols on IP, used by a major packet network such as the internet.

1.1.3. *Main networks*

After presenting architectures and then network technologies, we present here in more detail some main networks corresponding to services which are initially distinct (such as fixed telephony, mobile telephony, data, etc.), and then converge into a single network. We will describe their equipments and their functions – functions at the source of the traffic loading the previously identified equipments. For this purpose we will consider some examples of networks such as: the fixed telephone network, GSM and UMTS mobile networks, the IP network, the NGN network, and private networks.

1.1.3.1. *Conventional fixed telephone network*

The role of the telephone network is to connect two fixed subscriber stations.

The subscriber equipment is the telephone set(s). The subscribers have a fixed connection to access (connecting) units (concentrators, subscriber centres). The communication is established from the numbering (number requested) and maintained throughout the conversation. The possible bit rate is 64 kbit/s (56 kbit/s) or *n*x64 kbit/s for ISDN (*integrated services digital network*), but evolving on access up to Mbit/s with ADSL (*asymmetrical digital subscriber line*) access. This latest evolution is indisputably, as we will see later, a fundamental opening up of the conventional network towards new generation networks. ADSL enables, simply by adapting existing subscriber lines, downlink bit rates (network to user) in the order of 2 Mbit/s, and uplink bit rates in the order of 600 kbit/s.

The 3 basic functions of the network are:
– the interconnection of subscribers: supplying the signals from the terminals (voice, data) with a permanent dedicated transport medium throughout the duration of the call;
– signalling: the exchange of information (messages or frequency signals) enabling the set-up and then the release of the call and its medium (based on the dialled number);
– operation: i.e. the exchange of information and commands (messages) enabling the management of the network (traffic measurements, commissioning...).

These three functions give rise to three networks which are essentially physically separate: the transport network with its transmission links (PCM, STM, etc.) and its switching matrices, the control and signalling network with its exchanges and its signalling network (e.g. no. 7 signalling), and finally the operation network with its operation centres.

Figure 1.7 shows the main components of a public fixed network, the PSTN (*public switched telephone network*).

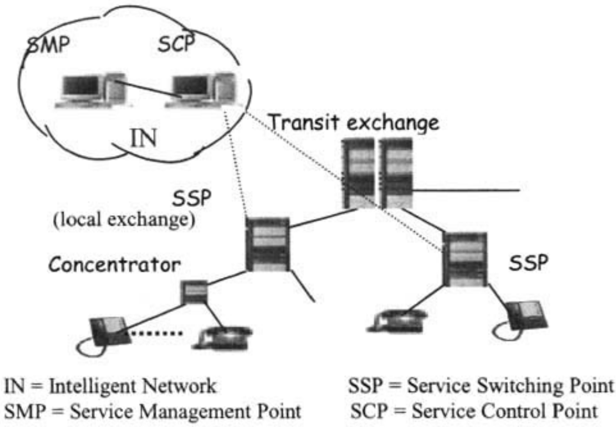


Figure 1.7. *The fixed telephone network*

We find in succession the *subscriber connecting units* (concentrators), the subscriber centres (*local exchanges*), and then the *toll exchanges*, to which are added the *operation centres*. To this are added the equipment of the *intelligent network* (IN), such as the SCP (*service control point*) and the SMP (*service management point*). The PSTN has now become a very powerful network (except for its bandwidth limitation), particularly thanks to the IN. The intelligent network constitutes a powerful environment capable of supplying tailored services, such as credit card calls, portability, free phone service, etc. It is the SCP which controls the supply of the service in real time, in cooperation with the switching exchanges. In this context, the switches become equipment with advanced functions, and are now known as SSPs (*service switching points*).

1.1.3.2. *N° 7 signalling network*

In discussing the PSTN network, and (as we will see later) the evolution of networks, reference must be made to signalling network no. 7. Signalling network no. 7 (SS7) is used to carry control information between the different elements of the network, such as the telephone switches, the databases and the servers. It is therefore an essential element in introducing new services on to the network.

The signalling network functions in accordance with a principle similar to that of packet switching: there is an exchange of "signalling frames" with flow control, between signalling terminations (*signalling points*, or SP) located in switches, servers, etc., and via relay points (*signalling transfer points*, or

STP). The transmission medium is the signalling channel, which explains why this is known as *common channel signalling* (CCS). A signalling channel is usually a 64 kbit/s channel reserved solely for transporting signalling between two signalling points (SP or STP). One or more signalling channels are reserved depending on the volume of messages to be exchanged, a volume which of course depends on the size of the switches. They then form a signalling link set.

All the signalling channels and signalling points (SP and STP) form a dedicated network which is completely separate from the voice transmission network (in logical terms at link level, for it makes use of the same physical resources). It is finally important to note that with this network the exchange of signalling is independent of the actual set-up of a switching circuit. This independence makes the network well suited for evolution to the new generation of networks such as the NGN (*next generation network*), which we will present later.

In general, an SS7 network has the schematic structure as in Figure 1.8.

For reasons of reliability, each signalling point is connected to two signalling transfer points, and the signalling transfer points are intermeshed.

There may be direct links between SP (or STP in the same region). There may be links between several levels of STP (regional level, national level, etc.). A distinction is drawn between three modes of functioning linking the signalling channel with the voice circuits that it serves:

– *Associated mode*: the signalling channel follows the same path as the circuits whose signalling it carries. (In this case it goes from SP to SP);

– *Non-associated mode*: the signalling channel follows a different path. (It then uses several STP); the latter is itself divided into *dissociated* mode and *quasi-associated* mode. In dissociated mode, the messages can use a large number of STP and can follow different paths. In quasi-associated mode however, the routing is predetermined with a maximum of two STP.

1.1.3.3. *Mobile networks*

The GSM network (global system for mobile communication)

This is the basic mobile telephone network. The functions are the same as before, but this time the subscribers to be connected are “mobile”. In addition to mobility, it offers services such as short messages, prepaid cards, information services and voice mailbox, etc.

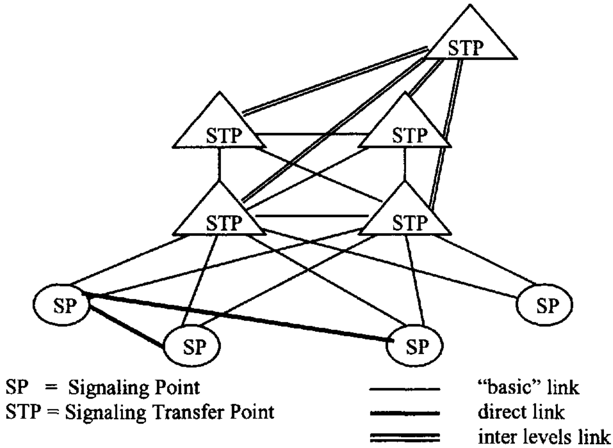


Figure 1.8. *SS7 signalling network*

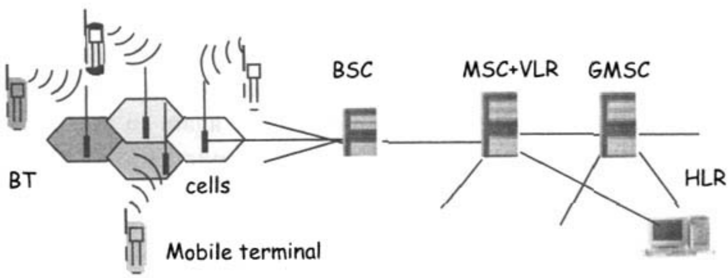
Access network

At any given moment, a subscriber belongs to a "cell", a zone covered by an aerial capable of offering a certain number of radio channels to the users of the cell (the equivalent of the subscriber concentrator). The subscriber equipment is the *mobile station* (the "mobile" or MS). The mobile subscribers are connected to the network via a radio link with the pilot station of the cell in which they are located, the BTS (*base transceiver station*). A control station, the BSC (*base station controller*) supervises several BTS.

The core network

The BTS are connected to a *mobile switching centre* (MSC) which is the equivalent of the subscriber centres and toll exchanges. The main difference between an MSC and a fixed network switch is that the MSC takes into account the impact of subscriber mobility (location, change of radio coverage zone, etc.). The MSC carries out the functions of control and connection of the subscribers located in its geographical zone. It also acts as a gateway between the fixed network and the mobile network, or between mobile networks, for incoming calls for which the location of the called party is not known. An MSC which receives a call from another network, and which routes this call towards the MSC where the called subscriber is in fact located, is called the *gateway MSC* (GMSC). To do this it consults the location database, the HLR that we describe below. In this way a fixed subscriber of the PSTN, or a mobile subscriber from another distant mobile network, can communicate with another mobile subscriber of the PLMN (*public land mobile network*), whatever his/her instantaneous location may be.

Subscribers may move from one cell to another, even during a call (this is then referred to as a *hand over*), for the radio system continuously tracks their location. Subscribers have a home location: the HLR (*home location register*) is the system that holds information relating to its home subscribers (identity, number, subscription options, services, etc.). In addition to this static data in the HLR are added dynamic data, such as the last known location of the subscriber, which enable routing to the MSC where the subscriber is in fact located. Finally, it is the VLR (*visitor location register*) which updates the data relating to subscribers "visiting" its zone, and which notifies the HLR. The VLR is usually included in the MSC.



BTS = Base Transreceiver Station
BSC = Base Station Controller
VLR = Visitor Location Register
MSC = Mobile switching Center
GMSC = Gateway MSC
HLR = Home Location Register

Figure 1.9. *Mobile network*

The GPRS and UMTS networks (universal mobile telecommunication system)

In the basic mobile network, GSM, the bit rate is 13 kbit/s for the radio part between the station and the BSC, and 64 kbit/s (56 kbit/s) after transcoding between the BSC and the rest of the network. Mobile networks such as GPRS (*general packet radio system*) and UMTS are evolutions which offer higher bit rates (144 kbit/s, 384 kbit/s and 2 Mbit/s), and can be used for communication in packet mode.

The GPRS network should be seen essentially as an evolution of the existing GSM network. It basically adds to GSM the possibility of sending data in packet mode. The bit rate can reach 144 kbit/s. The voice service uses the GSM network, while the data services are forwarded to the packet network (e.g. the Internet) *via* the GPRS network. Two new functional entities are involved: the SGSN (*serving GPRS support node*) which transmits data between the mobile terminals and the mobile network, and the GGSN (*gateway GSN*) which interfaces the mobile network with the data network.

In the UMTS network, the whole network evolves. A distinction is drawn between the radio part, RNS (*radio network subsystem*) and the core network part, CN (*core network*).

The RNSs form the access network. They include as before radio resources, but with higher bit rates in this case, and the RNC. The access network (UTRAN, i.e. UMTS *terrestrial radio access network*, for the general public network) consists of radio resources as before (but with a higher bit rate, the BTS becomes Node B), and of the RNC (*radio network controller*) which controls all the B Nodes of the UTRAN.

The core network is quite similar to that of the GPRS, but a distinction is now drawn between two domains: the circuit switched domain (CS) and the packet switched domain (PS). the MSC and GMSC are again in the CS. In the PS their equivalents are the serving GPRS support node (SGSN) and the GGSN (gateway GSN). Finally the common elements are also present, such as the HLR and the authentication centre.

A UMTS mobile is capable of communicating simultaneously via both domains.

Figure 1.10 shows the network and its functional components.

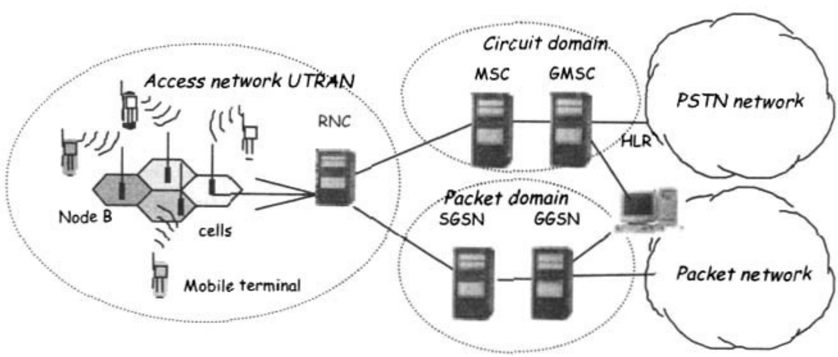


Figure 1.10. *UMTS network*

1.1.3.4. *The internet network*

This is a packet switched network. As explained earlier, there is no setting up of a channel strictly reserved for a call. This was initially justified by the fact that, when computers exchange information such as files, on the one hand the actual information exchange is brief (and lasts only a short time compared with the whole session), and on the other hand the type of data exchanged is conducive to division into packets, with few real time constraints. This enables a high degree of *multiplexing*, i.e. the mixing of packets of several calls on the same medium, with a bit rate at least equal to the sum of the call bit rates.

The IP network (*internet protocol*) is the interconnection of several *subnetworks* or *autonomous systems* (AS).

A subnetwork interconnects terminals, servers and computers. In most cases this is a LAN (see later). A subnetwork is interconnected with the other subnetworks via a port router. Each equipment has an address, which identifies it in the subnetwork (the subnetwork is defined by a specific prefix). Unlike a telephone-type network, the addresses have no hierarchical structure (so that an address does not necessarily enable the identification of the terminal). The routers have a partial view of the overall network and exchange information with neighbouring routers whose addresses they know. (Clearly the tables would become gigantic if each router had to know the whole of the network.) The updating of the address tables is moreover one of the problems that these networks must handle.

A router has two main functions: the *routing* of the packets and *forwarding*.

The routing of packets

The router has a routing table which contains a more or less detailed knowledge of the network topology. It is from this table that it calculates the route to be followed by a packet for a particular destination. IP routing itself is an enormous subject. As already indicated in the section on network architecture, the IP network ensures reliability of communication thanks to its many interconnections. However, this involves on the one hand dynamic routing capacities, to bypass failures and congestion, and on the other hand the need to control and optimise the available bandwidth, and route lengths. Today the internet transports all sorts of information: files, messages, pictures, voice, video, etc. Its evolution in terms of size and in terms of diversity of services, thus tends to strengthen the need for *hierarchy* in routing, and the concept of *differentiation* of the service between packet flows (*Intserv* and *Diffserv* type priorities for instance).

In this work we will consider an example of the problem of multiplexing traffic of different types using packet switch technology.

Packet forwarding

On the basis of the packet header, the router transfers the packet from one of its input ports to the appropriate output port (in accordance with the result of route calculation). Inside the routers problems of congestion and waiting, etc., may also arise. We will also look at these issues.

Communication protocols

A major IP packet network such as the Internet is based on three main protocols: TCP (*transmission control protocol*), UDP (*user datagram protocol*) and *RTP (real time transport protocol,* associated with RTCP: *RTP control protocol*).

The role of TCP is to supply a reliable data supply service to the application programmes, by drawing on the connectionless, unreliable packet supply service provided by IP. This is an end-to-end protocol. TCP is a connection-oriented service which begins by establishing a virtual (bi-directional) connection between two applications. The data are then transmitted on this connection. The sequencing is guaranteed and there is no loss. TCP resends the packets if an error occurs and adjusts the bit rate if congestion occurs, but this does not guarantee delays and makes it incompatible with real time data transport.

UDP is the transport protocol used by RTP to process real time flows (voice, video). UDP is a protocol without error correction, without bit rate reduction, but also with no guarantee that the packets are delivered into sequence. RTP enables the adding to the packets concerned of time markers and sequence numbers. RTCP, *via* periodic transmissions, offers possibilities of controlling RTP flows. RTP and RTCP are application level protocols.

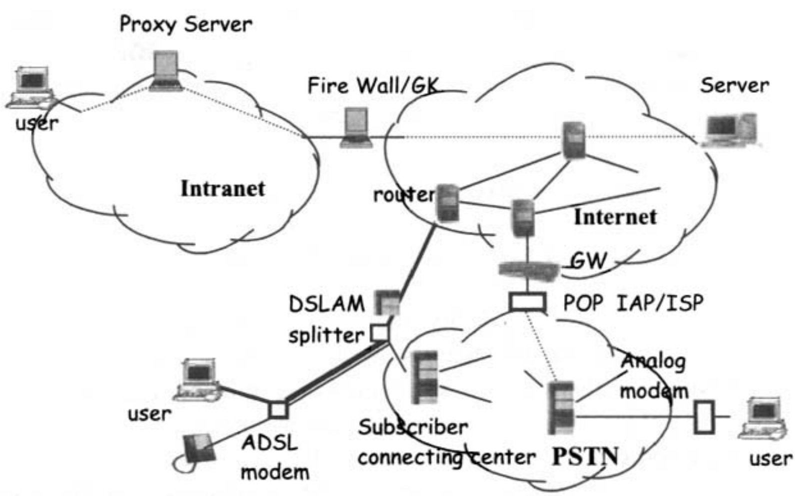
Finally, as IP has evolved towards the simultaneous transport of all services, both real time and non-real time, and with a quality of service guarantee, new functions have been added: the introduction of the concept of priority amongst packets (*Intserv*, *Diffserv*), and the possibility of creating through the networks *tunnels* (predefined paths, *tunnelling* function) by making use of evolved protocols such as MPLS (*multiprotocol label switching*).

Connections to internet

We have just explained that the internet network is in fact the interconnection of several "subnetworks". We will look in detail now at the various modes of connection and interconnection.

The terminal is connected via an access network, which for a residential subscriber will for example be the telephone network. In this case the user terminal will be equipped with a modem to access via the telephone network the *point of presence* (POP) of an internet access provider or internet service provider, which will then use another modem to connect to the internet core network. As the bit rate of the conventional telephone network is very limited (56 kbit/s, 64 kbit/s, 128 kbit/s), the user may also access the internet network via cable or an ADSL *(asymmetrical digital subscriber line)* link. In this case the bit rate is several Mbit/s and connection to the Internet network is direct and permanent, through wired access networks or DSLAM (*digital subscriber line access multiplexers*).

At subnetwork level, as some of the subnetworks are interconnected and thus open to the external world, the problem arises of protecting the internal information of *intranets* (private subnetworks) against leaks or external attacks.



POP = Point of Presence
DSLAM = Digital Subscriber line Access Mux
PSTN = Public Switched Telephone Network
ISP = Internet Service Provider
IAP = Internet Access Provider
GK = Gate Keeper

Figure 1.11. *The internet network*

This is the role of the fire wall and gate keeper functions at the gateway between an intranet and the internet. The fire wall will restrict the authorised types of connections depending on various criteria, for example by authorising a file transfer protocol transaction (FTP) only in the Intranet to Internet direction. In the same way, a proxy server can test the nature of a request before authorising connection. Figure 1.11 illustrates the various configurations.

1.1.3.5. *The next generation network (NGN)*

The *next generation network* represents the evolution of telephone networks and data networks towards a single network with a packet switch technology core network, carrying the data corresponding to the range of services offered to users (voice, video, files, messages, etc.). The separation of the control and transport planes is a key element of this architecture.

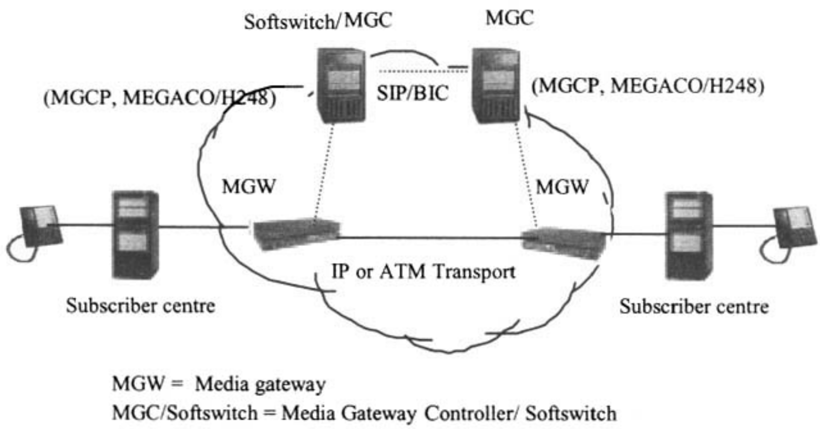


Figure 1.12. *The basic structure of the NGN*

The NGN introduces the flexibility that enables operators to adapt their activity and their network to changes in technologies and changes in the market. Operators with two types of network, telephone and data, can ultimately merge them and operate a single multi-service network. The NGN has a layered structure (terminal, access, transport, adaptation, control, application) with open interfaces enabling the combination of the various elements. The fact that the network medium part and the control part are separate means that they can evolve separately. The transport layer can be modified without impacting on the control and application layers. The packet transport may be either IP or ATM.

The NGN architecture is based on two main entities: the *media gateway* (MGW) and the *media gateway controller* (MGC), also known as the *softswitch* (of which the MGC then becomes a function).

Figure 1.12 provides an example of an architecture containing the two components.

The telephone subscriber terminal is connected to the access switch (access layer). The media gateway (adaptation layer) carries out the conversion at transport level between the coded information at 64 kbit/s and IP or ATM packets. The signalling, like the data, are exchanged via the packet network. The media gateway controller acts as the call server, and controls the MGW to set up the calls.

The setting up of the call between two subscriber terminal equipments is performed via the IP (or ATM) network under the control of the origin and destination media gateway controllers. Schematically, IP (or ATM) addresses are exchanged between the gateways, and there is then a request to set up a circuit call at the extremities on the telephone networks, followed by the exchange of the information via the telephone and the IP network. (We will look in detail at an example of NGN call set-up in the chapter on Methodology.) This is where use is made of signalling no. 7, conveyed after conversion on the IP network by a transport layer of the same level as TCP and UDP: SCTP (*stream control transmission protocol*) has been defined by the SIGTRAN group at IETF (see Chapter 2 on Quality of Service and standardisation groups).

The control protocol used by the MGC to drive the media gateways is either MGCP or MEGACO/H248. MGCP is a U.S. protocol, and MEGACO is the IETF H248 protocol of ITU-T (see Chapter 2).

If two MGCs are to dialogue with each other, for example to retrieve the IP (or ATM) addresses of a media gateway controlled by another MGC, they exchange signalling by means of the following protocols: SIP (IETF), i.e. *session initiation protocol*, or BICC (ITU-T), i.e. *bearer independent call control*.

The evaluation of the call set-up time in this type of network will be considered in detail as an example later in this book (see Chapter 10).

More generally, the NGN is intended to serve all types of telephone accesses, such as analogue telephones, ISDN telephones, IP telephones, PCs, private networks and ADSL lines.

These equipments are then either directly linked to the transport layer, or interfaced by means of a media gateway that carries out several functions.

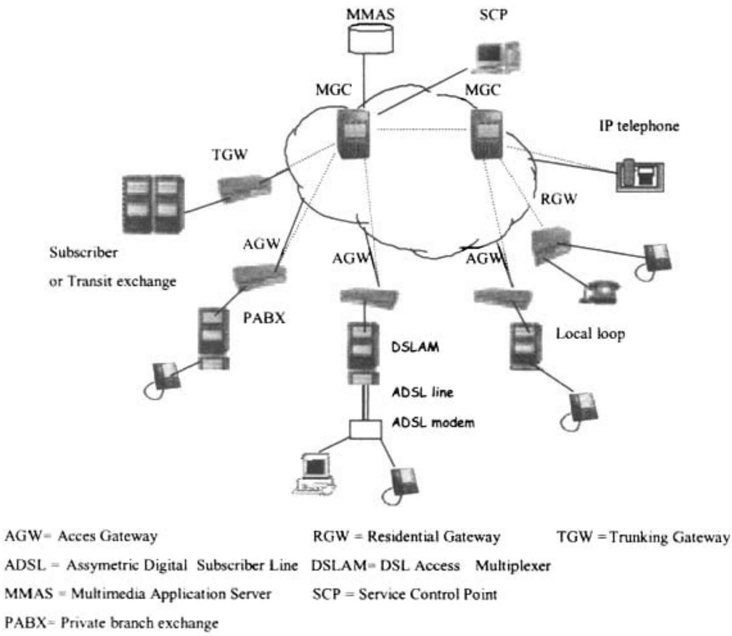


Figure 1.13. *The NGN and its access interfaces*

Conventional subscriber switches and transit switches meanwhile are interfaced by a *trunk gateway* (TGW). The connection function may also evolve to become the *access gateway* function (AGW) which directly interfaces the local loop, or in the same way a private switch. DSLAM (*DSL access multiplexers*) which group together ADSL lines carrying data, are also interfaced by an AGW. This demonstrates the considerable advantage of the ADSL line, which provides access to high bit rates for data networks. An analogic subscriber can also be directly connected to a *residential gateway* (RGW). An IP telephone on the other hand is connected directly to the IP transport layer (in which case the MGC supports H323 or SIP signalling), and it can access multimedia services offered by an application server such as MMAS (*multimedia application server*), etc.

Finally, the NGN network is also intended to interface both with fixed access and mobile access. As in the case of fixed telephones, the GSM circuit switches are then replaced by NGN access solutions.

Figure 1.13 shows some of the possibilities mentioned above.

To conclude this introduction to NGN, we should stress the fact that in this case, the core network (IP, ATM or IP/ATM) will support an extremely wide range of services. This raises the problem of managing packet traffic of very different types, and in particular that of the simultaneous transport of traffic types with strong real-time constraints (such as voice) and traffic types with very loose constraints (such as data files). This type of problem will also be considered in the rest of this book.

1.1.3.6. *Private networks*

The large networks we have just presented tend to be public networks. But private organisations, such as large companies and universities, organise the communication between their employees and their computer equipment by means of *private networks.*

PABX (Private branch exchange)

Private telephone networks such as PABX are quite similar to public networks. They are simply reduced to a minimum level of equipment for a small number of subscribers (ranging from a few dozen to a few thousand), but also offer facilities specific to companies such as an internal directory, conferencing, transfers, automatic callback, voice mail, call presentation and call filtering. Increasingly, these services are now also available on the public network.

LAN (Local area network)

Private computer networks, such as LAN, are intended to interconnect computer equipment, and now offer many facilities thanks to IP technology (intranet), such as file transfer and e-mail, but also voice and video image communication, real-time sharing of files and presentations, so making possible the holding of net meetings, etc.

LAN structures are of various types, the most common being the bus, star or ring structure. Interconnection of computers is carried out using standard mechanisms such as CSMA or token ring.

CSMA-CD (carrier sense multiple access with collision detection)

This is the mechanism used in *ethernet* networks. In this case each terminal wishing to transmit "listens" until no other terminal is active. Because of propagation times and minimum recognition times, collisions may occur, thus limiting effective bandwidth relative to the physical bit rate of the bus.

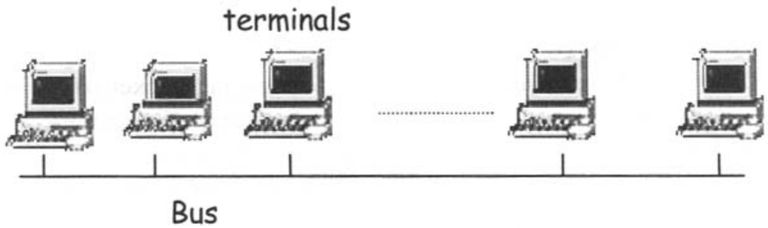


Figure 1.14. *LAN ethernet*

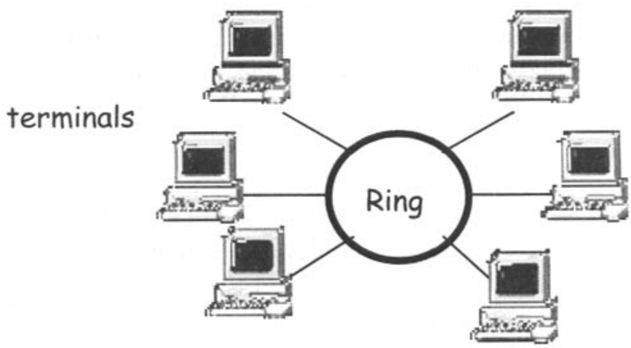


Figure 1.15. *LAN token ring*

The token ring

The token ring mechanism consists of circulating on the ring one (or more) tokens giving the right to transmit. The terminal that wishes to transmit takes hold of a "free" token as it passes, and associates it to its message after marking it as "busy". The terminal releases the token when it returns, and puts it back into circulation on the ring. It should be noted that it is the circulation of the token that results in the ring configuration of the network: the physical structure may be either a star or a bus.

The hub or switch

This is an evolution of the bus and ring structure which makes it easier to modify the number of terminals connected. The *hub* is a simple system of centralised connection, and the *switch* acts as a switching matrix. In physical terms the structure becomes a star structure, but in logical terms the communication continues to be of the bus or ring type.

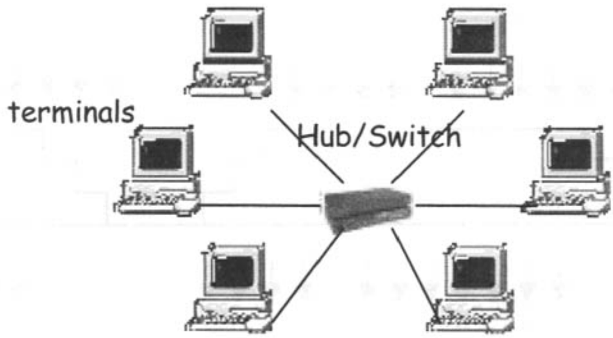


Figure 1.16. *Ethernet/token LAN with a hub (or a switch)*

Having presented the essential characteristics of telecommunication networks, we can now consider what will be the fundamental object of our study: traffic flow, and the associated performance, on the networks. With this aim in mind, we will begin by introducing the basic concepts of traffic and quality of service.

1.2. Traffic concepts

1.2.1. *Erlang concept*

The traffic of a telecommunications network is the volume of information transported or processed by the network. The information may consist of data relating to information exchanges between users (voice, images, e-mail, files, etc.), but also of data relating to exchanges of information between network "control machines" (signalling data on a circuit network, routing information on an IP network, and operating data, etc).

Clearly, the more frequent and the longer the exchanges between users and machines, the more resources are needed for traffic flow. For example, if a network receives over a given period permanent demand of 1 call per second, with each call lasting for 3 seconds, the network will continuously have N=3 calls coexisting. After a transient state, corresponding to the time needed to reach the expected load level, each end of call (corresponding to the outgoing process) will be replaced by a new start of call (corresponding to the incoming process), thus maintaining the level of load of the network during the period considered. The phenomenon is described in Figure 1.17.

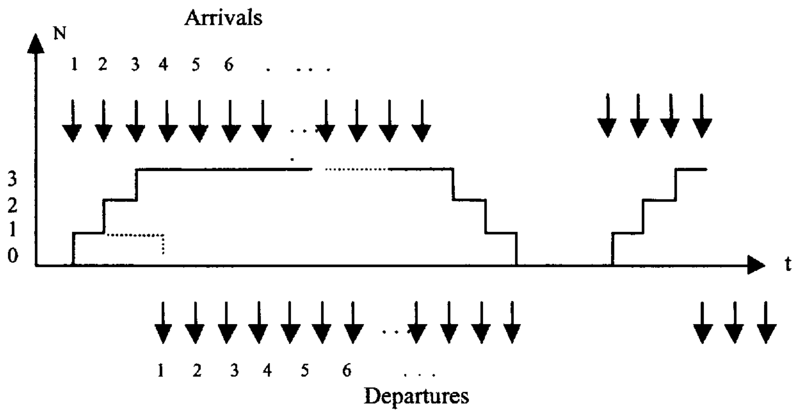


Figure 1.17. *Simultaneous calls and the notion of traffic*

To simplify, we have represented regular incoming flows and constant call times, but the phenomenon of course remains the same with arrivals and service times which vary around average values.

The average number of calls simultaneously in progress N is termed *traffic intensity*. The measurement unit is the *erlang*, denoted *E,* from the name of the distinguished Danish engineer A.K. Erlang (1878–1929) who established the first fundamental laws of traffic theory.

This concept is of course fundamental, as it defines the basis for sizing the network. So if a resource (a radio circuit, digital circuit, virtual circuit, bit rate, etc.) is associated with each of the N calls, a network with a capacity of at least N resources will be required to carry this traffic. The precise number of resources to be provided will depend on the law of arrivals and the service law. And this is exactly what enables the calculation of the famous Erlang formula in the case of Poissonian arrivals, i.e. arrivals conforming to a Poisson law. More generally, in order to be able to extend this concept to all types of telecommunications services, and all types of resources used, the following definition has been adopted: *a set of identical resources is said to carry at a given instant a traffic of N erlangs when N of its units are busy*. This definition covers both the concept of circuits and that of bit/s.

Formally, traffic in erlangs is denoted A, and if the number of busy resources is designated as $n(t)$, this gives the following for a period of observation T:

$$A = \frac{1}{T}\int_0^T n(t)dt \tag{1-1}$$

In more concrete terms, if one assumes a sufficient number of resources to carry all the requests presented, and if we call λ the mean number, constant, of requests per time unit and t_m the average occupation time of the resource by each request. This gives:

$$A = \lambda t_m \tag{1-2}$$

A.K. Erlang demonstrated the following fundamental result, called the *Erlang loss formula*, which gives the probability of rejection (B) of another request, because of a lack of resources, for a traffic A offered to N resources:

$$E(N,A) = B = \frac{\dfrac{A^N}{N!}}{\sum_{j=0}^{N} \dfrac{A^j}{j!}} \tag{1-3}$$

The traffic carried is:

$$A_h = A(1-B) \tag{1-4}$$

This formula also expresses the capacity of the system considered to handle the traffic offered to it. As the reality of a network is far more complex than this basic model, we will also have to deal with phenomena such as waiting, jitter, etc. But it will always be a matter of evaluating the resources necessary to handle traffic offered in acceptable conditions (loss, delay, etc.).

1.2.2. *Traffic offered, traffic handled*

This leads us to consider a fundamental distinction which is at the base of our performance study activity: the concept of *traffic offered* and the concept of *traffic handled (carried)*. The purpose of a network and of any telecommunications system is to handle if possible all the offered traffic as efficiently as possible (very low response time and transmission time, for example). In point of fact, it will not always be possible to accept all requests. In some conditions of abnormally high loads (in the event of a disaster for example) the systems must reject all requests, if only for self-protection. We will deal with this situation of *overload*. But also, leaving aside these extreme situations, it is clear that by the *random nature of traffic offered* (the level of demand varies randomly) and in order to *optimise resources*, there will also be a non-zero probability of a lack of resources and thus of demand rejection. The traffic handled will therefore generally be different from the traffic offered. The rules governing the relationship between these two values form the subject of quality of service standards.

1.2.3. *Quality of service*

The criteria used to determine rejection rates, or allowable response times, which are generally specified in international *standards*, are the bases of the concept of *quality of service* (*QoS*). Fundamentally, quality of service is related to the *perception* the *user* has of the network's response to his/her request. And it is important never to lose sight of this fundamental fact, for it is this which guides economic efficiency, and not performance for performance sake. However, achieving this objective is inevitably a complex matter, because of the diversity of requests, the diversity of the equipments involved, and the complexity of the networks used. Quality of service as seen by the user will in fact be the result of a coherent set of "*performances*" of all the network elements, performances which are also defined in international standards.

We have devoted a chapter to this issue (see Chapter 2), which is of course the motivation on which our efforts are focussed: evaluating the performances of equipments and determining the necessary and sufficient resources to ensure that the specified quality of service is finally achieved.

1.2.4. *Load profiles, load A and load B*

During a particular day demand may disappear at certain times and then reappear, with different load levels. This occurs in telephony for example with morning traffic, the low intensity hours at midday and then afternoon and evening traffic. This is also the case on data networks (internet traffic of professional users during the daytime, and of residential users in the evening). The concepts here are *peak hours* and *low intensity hours*, which periods may not necessarily be the same for different networks and even for different parts of the same network (different time zones, different types of service supported). We will then speak for a network or part of a network of the *load (or traffic) profile*. Figure 1.18 sets out a typical profile example based on observations on telephone exchanges.

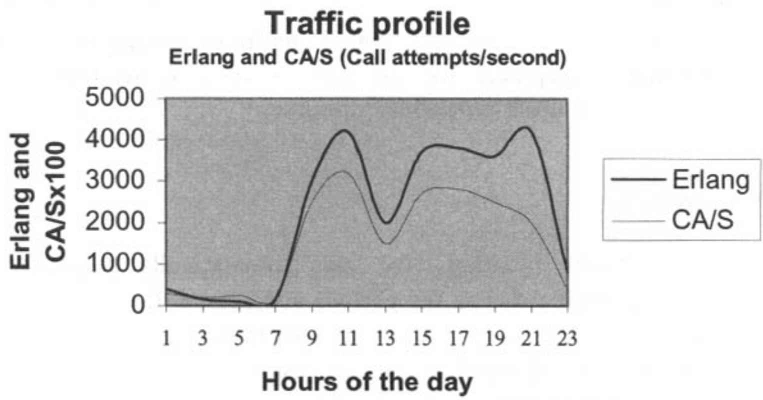


Figure 1.18. *Telephone traffic profile during a day*

The peak hours can be seen to be 0930–1130 hours, 1430–1800 hours and 1830–2100 hours, and of course for this type of traffic and the network considered, the off-peak hours at night and from 1230–1330. During these periods, variations in call time are also observed. Call times tend to increase in the evening: there are just as many erlangs even though the number of calls per time unit, CA/s (call attempt per second), is the lower. This is linked partly to the type of call made at this time of day (residential calls) and partly to the *charging* policy on the network observed, which favours this type of call in the evening. The latter parameter is a key element which the service supplier and network operator will use as the basis for making the most efficient use of their resources.

In addition to these variations depending on the time of day, substantial variations are also observed during the year, particularly at the time of important events such as national holidays, religious holidays, etc.

Clearly the network must be capable of responding adequately to the demands placed on it. It is for this reason that a distinction is drawn between "load A" and "load B". This distinction will be made more explicit in the chapter dedicated to quality of service. We will merely note at this point that load A corresponds to the most frequent situations, and that the quality of service perceived by the user must be as high as possible. For example if we refer back to our figure, and if this profile corresponds to a "normal" day for our network, load A will correspond to the traffic at peak hours. A detailed definition of a peak hour is given in ITU recommendations. Even if the nature of the traffic changes with the new packet type services, the concept remains valid (though periods may be different). Load B on the other hand corresponds to rare but foreseeable situations in which quality of service may be less good while remaining acceptable for the user. Clearly these recommendations reflect the inevitable need to optimise resources.

1.2.5. *Stationarity*

In the two previous situations, we have automatically assumed a certain "stability" in the traffic during the periods considered. It is assumed, in order to size and evaluate the quality of service, that there is a certain stability in the characteristics of the demand arrival processes over given periods of time. This property is known in mathematical terms as "stationarity".

In mathematical terms, a process is said to be stationary in the "strict sense" if its time distribution function is independent of any translation in time. It will in many cases be sufficient to consider stationarity in the "wide sense": in this case it is sufficient that the first two moments of the variable are independent of all translations in time. In this case one can state an important property: the expectation of a variable $X(t)$ is a value m, independent of the time, and is written:

$$E\{X(t)\}=m\,,\ \forall t$$

Thus the measurements over time of the mean traffic intensity during a stationary period, as represented in the traffic profile Figure 1.18 above, do indeed indicate the traffic flow during the day. During the day traffic levels change relatively slowly on the time scale of the length of a call (a few minutes for example in telephony), and present the necessary stationarity characteristics over periods of around one hour. The periods and time scales to be considered depend heavily on the services used, e.g. for internet type traffic in which the call length tends to be several dozen minutes.

1.2.6. *The concept of busy hour call attempts (BHCA)*

As already indicated, the traffic to be handled by a telecommunications network cannot be specified solely in terms of traffic intensity in erlangs. This does indeed well describe the phenomenon of occupation of the elements in charge of transporting user information. But allowance must also be made for the level of loading that this demand generates on the network control devices: call set-up requests or session set-up requests, signalling network loading, etc.

This load is expressed in terms of number of requests per time unit. In telephony the unit is the number of *busy hour call attempts* (BHCA). A *call attempt* (CA) is an attempt to set up a call which may be successful or not (conversation or incomplete dialling, called party busy, no response, etc.), its mean duration is calculated accordingly. This concept can easily be extended to include any other type of service request, in which case it becomes *busy hour events*.

In accordance with the definitions already given concerning the Erlang concept, there is a basic relationship between these two parameters, i.e. traffic intensity in erlangs and BHCA load. The formula is:

$$N_{BHCA} = \frac{A_{Erlang}}{\tau_{seconds}} 3600 \tag{1-5}$$

where τ is the mean duration of the request in seconds.

For example it is therefore said that a system must be able to handle a traffic of 10,000 erlangs and 360,000 BHCA (i.e. 100 CA/S, call attempt per second), for call attempts with an average duration of 100 seconds.

This terminology of busy hour calls attempts can of course be replaced by equivalent terms for different types of service, but the concept remains valid. It is important to bear in mind that the concepts are related to the "*user*" who sends or receives "calls". Similarly at user level it is probably the case that stationarity will continue to be observable over time periods that are easily humanly perceptible (which will not necessarily be true at the level of messages, packets, etc.). In this respect, it is impossible to overemphasise the importance in all traffic and performance studies of never forgetting the "user", who is the original source of the traffic and should also be the main focus of the studies.

In practice knowledge of three parameters is necessary: erlangs, BHCA, and call duration. On the one hand, there is not always a simple relationship between the specified load levels. For example, for B-loading the specification is usually a greater increase in BHCA than in erlangs (there are more attempts, but they are less successful), resulting in a shorter time per attempt. Furthermore, the sizing of the various network resources is mainly based on one or other of the parameters, depending on the resource in question. It is therefore clearer to express the capacity of the resource in the unit which best characterises it.

1.2.7. *The user plane and the control plane*

We have shown above that the resources of a network are sensitive to different traffic parameters, particularly if the resources are in fact control elements, or information transport elements that are in the process of being communicated. It is in this context that the concepts of *user plane* and *control plane* need to be introduced. The user plane consists of resources carrying "useful" information at user level (voice, image, data, files, etc.). The control plane will consist of resources in charge of setting up calls, signalling exchanges, observations, network management and network operation.

Clearly, the user plane will be primarily concerned with load in erlangs (erlangs of calls, of kbit/s, etc.) and the associated resources will be mainly at the *transport* level. In the same way, the control plane will be mainly concerned with BHCA demand (calls, transactions, etc.) and the associated resources will mainly be *handling processors* and *signalling links*. But as indicated earlier in the presentation of the networks, it is just as clear that there are many interactions between these different levels. For example, the packet network will be concerned at the level of its links by the flow of user packet traffic and by the transport of signalling packets (e.g. NGN network). In the same way, the control processors of a voice network are concerned not only with the setting up of calls but also by the number of erlangs of conversation to be charged. Any effort to size and assess performance will therefore require first the characterisation of the traffics in the different planes, in terms of erlangs and request rates, and then precise characterisation of the resources associated with handling the different types of traffic.

1.2.8. *Characterisation of traffic*

The resources necessary for satisfactory traffic flow can only be determined if the characteristics of the traffic are defined, on the one hand in terms of *services*, (call types, bit rate characteristics) and, on the other hand, in terms of *use* (*penetration* of subscribers) and of *flow distribution* over the different branches of the network. This leads into the concepts of *service characterisation*, and then into those of traffic *mix* and traffic *matrix*. These three concepts, which are very general but essential, apply just as well to data traffic as to voice traffic, video traffic, etc., and also at the level of a network node or that of any larger subsystem forming part of the network.

1.2.8.1. *Characterisation of services*

The great variety of services now offered by networks necessarily calls for a great variety of resource needs. This is particularly true in that, as we saw earlier in the presentation of switching technologies, it is necessary to constantly strive to make optimum use of resources in view of the real needs of each service. In traditional circuit networks, the characterisation of a service such as telephony consists of a call arrival law (usually Poisson), a call duration (exponential law with a mean value in the order of a few minutes) and a constant bit rate (e.g. 64 kbit/s). This means that it is easy to evaluate the resources used: in this case, as many 64 kbit/s channels as are indicated by applying the Erlang formula. The same voice service in a packet network will make it necessary to distinguish between coding with or without silence suppression, and thus between packet flows (of fixed length) at *variable bit rate* (and thus with an inter-arrival time that is to be defined) or at *constant bit rate*. If furthermore we consider e-mail type services, or more generally web-associated services, we will not only find flows of variable bit rates but also *packets* and *flows* of packets with highly variable lengths, with laws (packet length, flow arrival times) to be defined. Finally, in a multimedia environment, the subscriber during the same call initialises several sessions with different characteristics, depending on the nature of the information desired (images, voice, files, etc.), one very simple example being that of the *netmeeting* (conversation with images of the participants, whiteboard and sharing of *powerpoint* type presentations between several sites).

Thus, in addition to the call concept, we need to introduce the concepts of sessions, flows and packets, with the characteristics of the associated traffic laws. This characterisation is classically represented by the diagram in Figure 1.19, in which appear various *levels*: *call*, *session*, *flow* and *packet*.

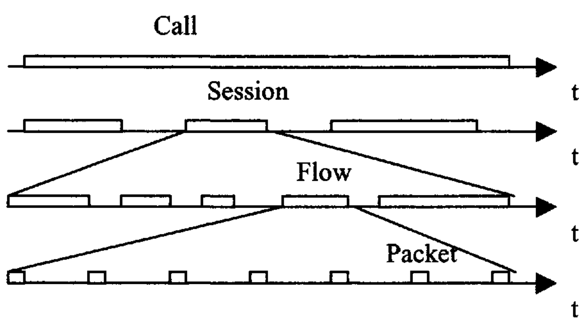


Figure 1.19. *The flows of a given service observed at different levels*

1.2.8.2. *Traffic mix*

The telecommunications networks we are studying, as shown earlier, support not only very varied *services* but also very different *user categories* with different *user profiles*. We therefore draw a distinction between professional users, residential users, and small and large business users. Depending on whether the advanced technology equipment level of the local network is high or low, in the geographical zone considered, and on the degree of interest shown by users in these technologies, users will make more or less use of the various services. The concepts are then those of *penetration rate* and the service *utilisation rate.*

It is on the basis of these characteristics that a *traffic mix* will be defined. For each user category will be determined the penetration rate and the service utilisation rate of each type of service (telephone calls on fixed or mobile network, VoIP session, video, web, on the packet network, etc.). Then, based on the penetration rate and utilisation rate of each of these services, for each user category, and the proportion of users in each category, it is possible to determine the traffic mix of an average user. For example, an average user may have a traffic of 0.1 E in mobile telephony, 0.1 E in telephony on IP, and 0.2 E of web, etc., with an associated number of call attempts and sessions. Another essential task will be to characterise each service, or group of services, in terms of network resource utilisation: circuits, constant or variable bandwidth, for bothway relations or not. This will lead to the characterisation of the "variability" of flows of certain services in terms of peak flow, average flow, variance, etc., and by means of more subtle approaches involving distinctions between levels (sessions, packets, etc.), to deal with problems of very high variability such as "self-similarity". We will discuss in this book how to deal with these typical aspects, particularly for IP packet networks.

These mixes are representative of the situation at a given moment and at a given place. These are references which always need to be updated in view of changes to the networks and based on observation. Clearly the mixes may differ markedly between countries with more or less developed telecommunication networks or between countries which are more or less industrialised for example. We will give precise examples in subsequent chapters.

1.2.8.3. *Traffic matrix*

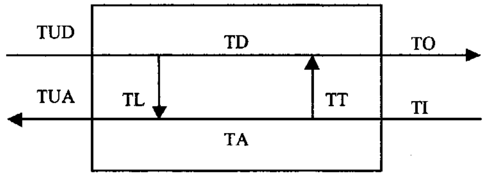
As we explained above, a telecommunications network is a structured set of nodes and links enabling the transport of information between users. Its efficient management from the viewpoint of traffic flow therefore requires, in addition to a knowledge of traffic mixes, a knowledge of the *distribution of traffic flows* between the various directions, and a knowledge of *traffic affinities* between the various origins and destinations. The role of the *traffic matrix* is thus to define flow distribution at the level of the nodes and at the global network level.

1.2.8.4. *"Node" level matrix*

A switching node on the network basically receives traffic and transmits traffic. Depending on its position in the network, at the frontier or at the heart of the network, its role will tend to be an access function or a transit function. An access node will for example provide access to the rest of the network for a set of users: calls *to* the network and calls *from* the network. It will also be able to connect subscribers locally. Because of these functions we attach to it the concepts of *originating traffic*, *terminating traffic*, and *local traffic*. On the other hand, in the core network, the transit node will switch traffic between inputs and outputs without any real concept of originating, terminating or local traffic. In this case the concepts considered are thus those of *incoming traffic* and *outgoing traffic*, or more generally the concept of *transit traffic*.

In this respect it is important to note that these concepts of incoming and outgoing traffic are clearly attached to the concept of call set-up request, depending on whether the request *goes towards*, or *comes from* the network, and not on the direction of information transmission during set-up or during the call. As already indicated, it is the characterisation of the service at bit rate level that will determine the direction of transfer of the information being communicated (e.g. for a unidirectional or bidirectional relation). This point is all the more important in that IP type services often involve bit rates that are very different between the "uplink" direction (from subscriber to network) and the "downlink" direction (network to subscriber). This characteristic is taken into account by the ADSL line, as indicated earlier.

Clearly the resources utilised for the set-up of a call, and the traffic flow during the call, are not the same in the case of a local call, an incoming call or an outgoing call (the analyses, the routing and the interfaces for example are all different). Any evaluation of the resources necessary for the handling of these types of traffic will therefore require a detailed description in the form of a matrix. Furthermore, and particularly at the level of a user connection node, there are important relationships between the various flows and also between the flow in erlangs and the flow in calls. The vocabulary must be precise in order to avoid any confusion for example between *subscriber traffic* and *switched traffic*. To illustrate this point, the overall representation and vocabulary of Figure 1.20 are usually adopted.



TUD : originating user traffic (departure)
TUA : terminating user traffic (arrival)

TD : outgoing traffic (departure)
TA : terminating traffic (arrival)
TL : local traffic
TT : transit traffic

TO : outgoing traffic
TI : incoming traffic

Figure 1.20. *Traffic matrix of a node*

The corresponding traffic intensities are usually expressed in erlangs. We obtain the following expressions:

User (subscriber) traffic:

$$Tu = TUD + TUA \tag{1-6}$$

which characterises the total activity of subscribers in erlangs.

Network traffic:

$$Tn = TO + TI \tag{1-7}$$

Traffic switched by the node:

$$Ts = TD + TA + TL + TT$$

which is the load to be handled by the node:

$$Ts = \frac{Tu + Tn}{2} \tag{1-8}$$

Subscriber traffic switched:

$TD + TA + TL = TUD + TUA - TL$, which characterises the number of subscriber calls to be handled. Internal traffic (i.e. local traffic) gives rise to only one call between two subscribers. Care must therefore be taken, in the event of a large proportion of local traffic, to evaluate correctly the number of calls generated by subscribers. In other words, for the same subscriber traffic value in erlangs, the number of calls to be processed at node level will be lower the greater the proportion of local traffic.

Network switched traffic:

$TD + TA + TT = TI + TO - TT$, which correspond to the same characteristics as at subscriber level. Note in particular that a pure toll exchange has to handle only half the total of the incoming traffic and outgoing traffic.

The following expressions are also generally used:

$TL = l \times Ts$, where l is the percentage of switched traffic giving rise to local (internal) traffic. (1-9)

$TT = t \times Ts$, t is the percentage of switched traffic giving rise to transit traffic. (1-10)

From the above expressions the following can also be obtained:

$$Tu = Tn\frac{1+(l-t)}{1-(l-t)}, \text{ or } Tn = Tu\frac{1-(l-t)}{1+(l-t)} \qquad (1\text{-}11)$$

A basic relationship which is expressed only on the basis of a constant connection capacity (total number of subscriber access connection links and of network connection links), a switching node will handle more subscribers the less network traffic and the more local traffic there is, and vice versa.

This is why node capacity is often expressed as switched traffic capacity rather than as subscriber and network traffic.

Separation of control plane and transport plane

The relations we have just set out must be seen as applying independently of the physical reality or otherwise of the transport connections in the node considered. Whereas traditionally in the telephone network the concepts of calls, processing load, and erlangs on network links are directly associated, this is not the case for *next generation networks* (NGN). An MGC can control several *media gateways* (e.g. several *trunk gateways*), and will thus handle the control traffic (in calls/second) corresponding to the load in erlangs on links which are not connected to it. This clearly demonstrates the impact of the separation of the control and transport planes, an essential characteristic of the NGN. In practice, this means that the construction of the control flow matrix provided earlier, and the establishment of the traffic mixes, must be carried out on the basis of the study of a broader configuration corresponding to a network subsystem. In this case we must consider a network matrix, as in the case of the global study of a network.

1.2.8.5. *Network level matrix*

In a network, the nodes are connected by links that must be sized according to the traffic they must handle. It is therefore necessary to define a *traffic matrix,* which indicates the volume of traffic exchanged between each pair of nodes. Generally speaking, the matrix is not necessarily symmetrical (readers will recall the distinction drawn earlier, establishing symmetry in terms of traffic offered and not in terms of occupation of circuits or bit rates).

The traffic matrix, once obtained, will enable the determination of the resources to be allocated to each link, and perhaps also the study of the principles of flow routing (if a flow from A to B is too low, it may be best not to open an A-B link, but instead to carry out the transit of the corresponding flow to another node C: this modifies the matrix which is used for sizing, by causing the disappearance of the A-B flow).

In point of fact, the traffic matrix corresponds to the flows to be handled between each pair of nodes, whether there is a link between them or not.

Furthermore, as we saw earlier, a matrix of this sort in a network such as the NGN may also form the basis for the construction of the control flow matrix at node level.

The construction of a network matrix consists of transposing the organisation of the flows of an actual node network (A, B, C etc.), as seen above, into a mathematically coherent matrix with the form shown in Table 1.1.

Each element x_{ij} of the matrix gives the value of the traffic flows (in calls/second or in erlangs, packets, etc.) circulating from node i to node j. The sums, by rows and by columns, represent the outgoing traffic (O) of node i, or the incoming traffic (I) of node j. Overall coherence (T) must of course be ensured at the level of the whole network.

Table 1.1. *Traffic flows matrix*

To / From	A	B	..	i	j	Total outgoing
A	x_{AA}	x_{AB}	..	..	..	$\mathbf{O_A}$
B	x_{BA}	x_{BB}	..	..	..	$\mathbf{O_B}$
..	..	..	..	..	..	..
i	..	..	..	x_{ii}	x_{ij}	$\mathbf{O_i}$
j	..	..	..	x_{ji}	x_{jj}	$\mathbf{O_j}$
Total incoming	$\mathbf{I_A}$	$\mathbf{I_B}$	..	$\mathbf{I_i}$	$\mathbf{I_j}$	**T**

The *valuation of the elements* in the traffic matrix is usually quite a difficult operation. In the case of an operational network, on which all measurements are possible, it could be constructed on an experimental basis, after elaborate measurement campaigns (it is necessary, at the output from each node, to discriminate between requests and classify them according to the final directions they are requesting). In the case of a network that does not yet exist, or is being transformed, the situation is even more complex. On the other hand, it is often possible to obtain an overall evaluation of the total traffic transmitted by a network node, for example, by estimating the global activity of the users connected to it.

It is then a matter of distributing these traffics over the various possible directions. Various methods have been used. The *gravity flow* models postulate an affinity between the nodes that depend on their respective distances. This is a model that may be justified in the case of a long-distance network (an international network, for example), but which would be unrealistic in the case of an urban network. In that case it is then possible to define on an a priori basis *affinity coefficients*, and distribute the flows on a pro rata basis.

The simplest approach will of course consist of implementing the Kruithof method, which distributes flows between nodes in such a way that the global traffic constraints are complied with on each of the incoming and outgoing traffics of each switch. This method will be presented in Chapter 9, "Models", in connection with forecasting the growth of a network, but the method may be applied for example by beginning from an arbitrary initial matrix.

This page intentionally left blank

2

Quality of Service and Performance

This chapter presents the fundamental notions of *quality of service* (QoS), and the attached performance parameters, for telecommunication networks and equipment. These notions sustain all the activity in performance evaluation of telecommunications systems. Actually, the network provider intends to offer, through a set of equipments, communication services with a quality level corresponding to the expectations of the customers. These objectives are expressed by means of *quality of service* parameters, and translated into *performance* parameters of the network elements, more generally defined by international standards, or specified in invitations to tender issued by network providers to manufacturers.

The ability, for equipment, systems and networks architectures to conform to these parameters and their evaluation, are the concern of this book and of the tools that are presented in the following chapters. The presentation of the concepts and of the related parameters is based upon the standards in use, briefly summarized in this chapter. However, the notions we intend to develop are of a *generic* nature, formulated in different ways in different standards but without any fundamental change because they are not strictly linked to the technology of the day, so that we need only make references to typical definitions from a significant set of standards.

2.1. Standardization organisms and norms

Across the whole sector of telecommunications, *standards* have a prominent position. The term "standard" stresses perfectly the need of these universal agreements, as their goal is to provide "standard" interfaces allowing heterogeneous components, provided by different manufacturers, to interconnect and communicate properly. The term 'norm' is sometimes used, referring to the commonly agreed level of quality that the network should provide.

In the sector of telecommunications, several international organisations aim at publishing such interworking rules:

– ITU (*International Telecommunication Union,* UIT in French, and referred to as CCITT – *Comité consultatif international des telecommunications* prior to 1993). This international organisation covers all domains of communications (networks, services, image, etc.). It publishes its work as *recommendations.*

– ETSI (*European Telecommunications Standards Institute*). This European organisation works in close relation with ITU, intending to specialise its recommendations to the European environment.

– IETF (*Internet Engineering Task Force*), a typical organisation of the IP world, accepting any contribution aimed at helping the development of the internet. Some of the IETF recommendations, termed as *Requests for Comments* (RFC), are de facto standards today (IP, TCP, SIP, etc.).

One could enumerate other bodies, for instance the *International Standards Organization (ISO)*, which groups national standards bodies and has a field of activity much wider than the "telecommunications" sector. Various forums are active too, their goal being mostly to act as "lobbies" favouring the development of a specific technology, and which may possibly come to issue standards completing the official ones, as it was the case of the *ATM-Forum* or the *Frame Relay Forum*, and others. National organisations may contribute also (AFNOR in France, T1 in USA, etc.). The bibliography gives the addresses of the major organisations.

2.2. Quality of Service (QoS)

The ITU Recommendation E.800 gives a definition of quality of service as "the collective effect of service performance which determines the degree of satisfaction of a user of the service". This is a rather general definition, which however conveys perfectly that, first the quality of service is significant for the "user" (whatever the user is, human being or service of higher level), and second that it depends upon numerous factors related to the performance of the network components, as we present hereafter. In what follows, the dedicated term QoS is in constant use referring to quality of service.

It is clear that a service can be used only if it is provided, and that, together with this provision, a description of the quality offered must be associated. From the provider's viewpoint, network performance is a concept by which network characteristics can be defined, measured and controlled in order to achieve a satisfactory level of service quality. It is up to the service provider to combine adequately different network performance parameters in such a way that the economic requirements of the provider as well as the satisfaction of the user are both fulfilled.

The user's degree of satisfaction with the service provided depends on the perception of the service performance in the different following domains.

Logistic support quality (service support performance). This refers to the ability of an organisation to provide the service in agreed delays, to ensure a correct management, e.g. in terms of billing, or assistance to the user. This is of obvious need for internet access for the management of service (subscribing, cancelling, installing a new version..), but also for mobile services, concerning the variety of pricing modes and of billing, and also for portability.

Service ability to be easily used (service operability performance). This concerns the provision of "easy-to-use" services, avoiding customer mistakes, assisting "navigation" through all the offered capabilities. Mobile terminals, with their numerous menus, are a typical example of the interest of convivial interfaces, and so are the internet sites, offering a large variety of services.

Security (service security performance). Here the point is the protection of the users and of the network against malevolent or fraudulent access to the services. The simplest illustration is the confidentiality of the data carried, but the protection provided by passwords for accessing computer accounts or mobile terminals are concerned too. Protecting the network against catastrophes, earthquakes or any other destruction is another fundamental aspect of security. It is actually measured essentially in the following domain.

Ability to obtain the service (serveability performance). This accounts for the ability of the service to be effectively available upon request. This means first the ability to reach the service, and once obtained to use it as long as requested. Air travel should give an explicit example, although far from our field, of the situation. First, a flight ticket must be purchased, and the flight must be on time. Then the journey must be without any incident, and terminate at the destination point. The parallel with telecommunication services is immediate; consider e.g. a mobile or fixed telephone call. First, the network must be available, then the called party must be accessible, and lastly, in case of answer, the communication must not be interrupted (think especially of handovers). Similarly, for web services, the internet service provider must be reached, then the access to the server must be provided, and the file transfer, or video communication must take place without loss of information or noticeable quality degradation. The terms of *accessibility* and *retainability* of the service are generally used to cover these aspects, and we now address them *via* the network performance parameters.

2.3. Network performance

Network performance is measured in terms of parameters, significant for network and service providers. They serve for system design, for system dimensioning and network provisioning, maintenance, etc., in order to satisfy both the customer and the provider.

The average customer is unaware of the way a specific service is provided, or of the design issues of telecommunication systems. However, we have stressed his/her interest in parameters giving account of his/her perception of the quality of service. These customer-oriented parameters cannot be used directly to settle network specifications. A network-oriented quality must thus be defined and expressed, both qualitatively and quantitatively, in order to communicate to the equipment and network provider information about:

- systems performance,
- network planning, equipment provisioning,
- operation and maintenance.

To this effect, several network performance parameters have been proposed, in relation to the user perception.

The approach has consisted, quite logically, in first defining *overall parameters*, also termed *end-to-end performance parameters*, then to define parameters at each network segment, called *intermediate parameters*.

2.3.1. *End-to-end and intermediate performance parameters*

The principle is as follows. The overall parameters take account of the perception the user has of the network performance, independently of its structure or technology, while the intermediate parameters ensure that the individual performance of each network element contributes to reach the global performance target. The following reference diagrams summarise this approach.

2.3.1.1. *Reference network configuration*

Figure 2.1 pictures a typical network configuration, providing communication between two users.

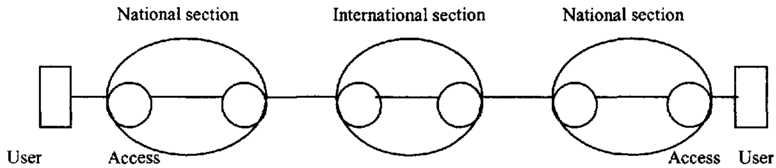


Figure 2.1. *Reference network configuration*

The number of nodes is arbitrary, depending on the relative position of the users and on the size of the network(s) to be crossed. The nature of the nodes is also arbitrary: *gateways* (GW), switching centre, routers, etc.

The overall performance parameters, such as connection set-up delays, information transfer delays, are defined from user to user. Moreover, similar parameters are specified for each element, bounding their contribution to the end-to-end performance: one speaks of the *budget* apportioned to each section of the network. The following pictures clarify this concept.

2.3.1.2. *Communication set-up reference diagram*

The first diagram presents the principle of the exchange of information (horizontally) from an initial request up to the confirmation that the action is successful. The vertical dimension represents the processing delays needed to set up the demand. The nature and the quantity of information to be exchanged depend evidently of the technologies and protocols in use (circuit, packet). The exchange may be complex, as for instance for communications between mobile users involving intelligent networks. However, the principle that leads to definition of the performance parameters remains the same.

2.3.1.3. *Communication, transfer phase reference diagram*

Once the communication has been set-up, the two parties may exchange "useful" information – referred to as information of user level: voice samples, data, images, files, etc.

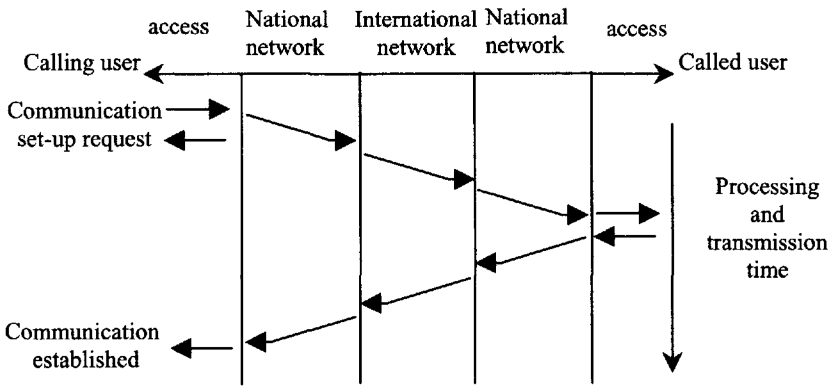


Figure 2.2. *Set-up phase*

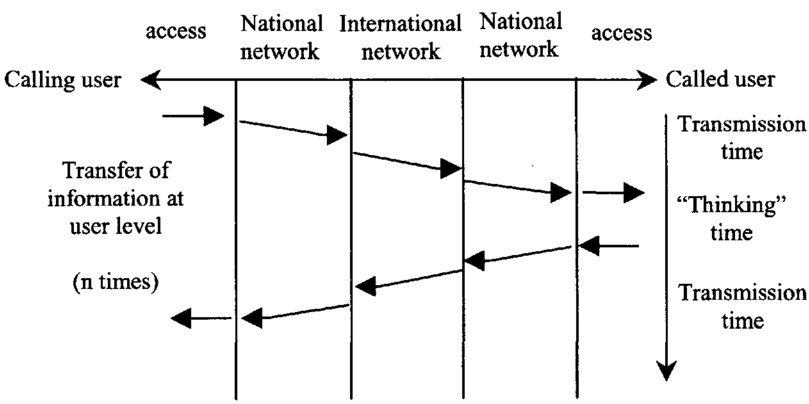


Figure 2.3. *Communication phase*

2.3.1.4. *Disengagement reference diagram*

The last phase is the communication *release*, also called *disengagement.* It proceeds similarly to the set-up phase (Figure 2.4).

These diagrams show clearly that the end-to-end performance requirements will depend partly on the network configuration, as a function of the number of nodes involved. This is why one defines performances of local, long-distance or international communications. These performances must however remain compatible with the perception the user has of the service. For instance, delays must not exceed certain bounds. On the other hand, it is possible to define performances for each network segment, independently of the overall configuration. Therefore, performance parameters are defined at the level of the nodes, the links, signalling points, etc.

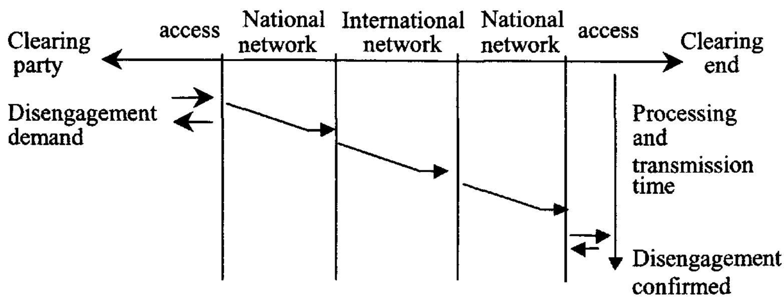


Figure 2.4. *Disengagement phase*

2.3.2. *Access, disengagement, and transfer phases*

From the above diagrams, two other notions emerge: the notion of communication set-up (and release), and the notion of communication established with its exchange of information. More generally, one refers to these as the *access phase (*with its complement the *disengagement)*, and as the *transfer phase*. Two major sets of performance parameters are defined, according to these two notions.

Two important reasons explain why these two families are distinguished, namely the *service* aspect and the *support* aspect.

The service aspects have been already discussed in the introduction. The point is to access the network, and the called party or the service provider within reasonable delays. This corresponds, for a connection-oriented service, to the connection set-up delay, but it could be, also, the delay to access a gateway or the edge provider, etc. Then, once the connection established, the quality of the information transferred between the two or more users must be at a level allowing its correct usage (semantic, temporal and spatial integrity).

Furthermore, the physical or logical support for each of these phases can be different. This is naturally the case, when the control network and the transport network are distinct. Actually, we have already presented in the first chapter the configuration of the telephone networks, with its clear separation between the signalling system SS7 for the set-up and release of the calls (using signalling points), and the voice transport network making use of PCM links and 64 kbit/s switches. The distinction is even emphasized with the concept of NGN (*next generation network*).

Other aspects, related with such notions as the quite different durations of the phases, and the different nature of the information processed, speak in favour of the distinction. For instance, the call set-up delay or the opening session duration is much lower than the call or the session itself. Similarly, the nature of the information transported during each phase is totally different (control messages and user information): this is clearly at the basis of the distinction already mentioned at traffic level, and it is in relation to that one talks of the traffic in the *control plane* and of traffic in the *user plane* (that is the "useful information" carried at user level)

Even if one has to be careful when interpreting these concepts, for a specific kind of network and of technology, they appear to be generic notions of essential use.

2.3.3. *Parameter values*

In order to characterise the targeted performance level, specific values are recommended for the parameters. Obviously, a single parameter cannot characterise a level by itself, e.g. a response time or a transfer delay. This would be far too simplistic, since the quantities manipulated in the domain of traffic are by essence random variables. Moreover, for services with strict delay constraints, it is important to guarantee both *maximum* and *normal* values. Now, giving a complete specification of the delay distribution would be complex and unrealistic. And lastly, as explained below, the traffic conditions vary, depending on the period of the year, of the day or even of the hour.

This leads one to specify, generally, four values: two load levels (said *normal load* and *high load*) and for each of the load conditions a *mean value* and a *quantile* of the performance parameter. Therefore, we ask of performance studies to provide not only averages, but also distributions, or at least their moments. Such a context explains the importance of the methods to estimate the moments of functions of random variables, of the sums of variables, and of the related approximate methods. Note simply at that point that the performance level is characterised by a mean value and a quantile, often for 95% (that means that, for instance, a delay must remain below a threshold 95 cases over 100). This will be specified for the different load levels we introduce now.

2.4. Traffic load conditions

Any telecommunication network is by nature submitted to variable solicitations – i.e. variable traffic conditions, as already evoked in the previous chapter. Not only does the traffic level fluctuate within the day, but also it varies during the year, or during specific events. However, one distinguishes three major conditions for the traffic environment:

– *normal day conditions* which correspond to the usual daily traffic conditions of the network,

– *high load conditions* which correspond to less frequent higher busy operating network conditions (special days such as year end or Mother's Day for instance may be considered),

– and finally *exceptional conditions* which correspond to load conditions far beyond the normal provisioning, and associated with totally unpredictable events (a catastrophe for instance).

In order to take account of these various configurations and to associate specific performance requirements, the notions of *normal load* (or load A), and *high load* (or load B) and *overload* (or exceptional) have been introduced. Note that these principles hold whatever the type of traffic, from signalling traffic to user traffic, and whatever the network technology. The load levels could even differ at a given instant for the various types of traffic. In addition, the conditions can differ from one country to another, or in different parts of a network. Only traffic observations allow one to define the load levels for the different network components.

Lastly, the traffic characteristics are in continuous evolution, as the number of users and their activity vary, and because of the evolution of the services and deployed technologies (for instance, mobile *versus* fixed customer, circuit-oriented *versus* IP technology, etc.). Therefore, operators have to re-evaluate constantly the volume and the nature of the traffic offered. As we have described in Chapter 1, these volumes are measured during periods where the traffic may be considered *stationary* (see Chapter 7).

That means that for the given period, it is possible to characterise the actual arrival process by a stationary model, with a given mean, variance, etc., and for instance as a Poisson process. Note that this model appears as being well suited to all kinds of traffics. Indeed, it renders clearly the natural behaviour of the users, who express their needs (call requests, internet sessions, etc.) randomly, and independently of each others; but also, as detailed in the section devoted to IP traffic characterisation, it induces important properties for traffic characterisation at flow and packet levels

2.4.1. *Normal load, load A*

This load level, called *normal load traffic intensity*, corresponds to the most frequent busy operating conditions of the network for which normal user service expectation should be met. In the recommendations it devotes to the network operation (see Rec.E.500), ITU-T recommends definition of the reference load traffic intensity over a monthly period, in order to get a statistically meaningful set of measurements (see Chapter 5) and to take account of seasonal variations. The daily traffic intensity is first determined by measures over consecutive time intervals called *read out periods* (the hour for instance). Then a set of days is chosen out of the month and ordered from the lowest to highest daily peak traffic intensity measurements. Particular days such as Christmas Day, etc. are excluded. The fourth highest value will be chosen as the normal load traffic intensity for the month being considered. Then, from these monthly values a *yearly representative value* (YRV) will be determined, defined as either the highest observed intensity or the second highest (if the traffic intensity tends to be fairly homogeneous from month to month). This will be the reference normal load for dimensioning. In fact, the point is not the exact value but rather reflects the spirit of the recommendation, which aims at differentiating normal and unusual situations. This is, basically, a question of optimising the installed equipments, as one does not intend to dimension for the worst-case situation. So, for instance, operators may commit to be able to carry "only" the normal load level in degraded situations (partial or temporary failure of equipment).

Such an approach is re-used in most ITU recommendations (see e.g. Rec. Q.543) and in the requirements issued by the operators concerning the equipments, when defining the notion of load A. The notion is defined as: "Reference load A is intended to represent the normal upper mean level of activity which network operators would wish to provide for on user lines and network equipment, while load B is intended to represent an increased level beyond normal planned activity level." This clearly describes relative situations, one being more severe than the other. We are thus now lead to define more precisely the notions of "high" load, and load B.

2.4.2. *High load, load B*

High load conditions identifies not very frequently encountered operating conditions for which user service expectation would not necessarily be met, but for which the level of performance achieved should be high enough to avoid significant user dissatisfaction (spread of congestion etc). ITU, in Rec.E.500, recommends selection among the observations over a monthly period as above, the day having the second highest daily peak traffic intensity measurement. This traffic intensity is then defined as the high load traffic intensity for the month being considered. Then, again as for normal load, from these monthly values, a *yearly representative value* (YRV) will be determined, defined as either the highest observed intensity or the second highest (if the traffic intensity tends to be fairly homogeneous from month to month).

As already said, Rec.Q.543, only specifies load B as a traffic increase over reference load A. The orders of magnitude are the most important to remember here: the reference load B is around 1.2 times load A, measured in volume of incoming calls, and 1.4 times for outgoing calls, that is around 1.3 on the average. Expressed in erlangs, reference load B is around 1.25 times reference load A.

These values have to be taken as orders of magnitude only, as the deployment of new services could alter them significantly. However, they clearly reveal the need to distinguish the different flows, originating and outgoing. In periods of heavy load, a significant proportion of the call attempts fails and does not result in effective user traffic (in terms of erlangs) in the network. More generally, the performance analysis must take account of the different types and conditions of traffic in the various domains (signalling, control, transport, etc.). Practically, one is lead to determine the most stringent conditions for each of the domain.

In this respect, the orders of magnitude given in Rec. Q.543 for load A and B are worth considering. Call rejection probability, around 10^{-3} for load A, goes to 10^{-2} for load B; set-up delay at the access around 600 ms for load A increases to around 800 ms for load B.

These figures must only be taken as orders of magnitude; however, they are likely to apply whatever the service or the technology in use, since they correspond to fundamental user perception. We come back to these points, at the end of the present chapter, by summarizing the main figures one can find in the recommendations in force. For the time being, notice that the values for load level B are not loose compared with load A. The reader is invited to verify (after reading the chapter about queues) that for the basic M/M/1 system the ratio on delays corresponds to a ratio on loads under the form $\frac{T_B}{T_A} = \frac{1-\rho_A}{1-\rho_B}$. Thus, for a ratio of 1.5 and $\rho_A = 0{,}7$, it becomes $\rho_B = 0{,}8$ that is only 1.15 times the load level A, although the recommendation mentions a ratio around 1.25.

This example makes it obvious that the performance targets for load B delays are, logically, the most constraining, for designing and dimensioning communications systems. This conclusion is rather general; however each situation demands a verification of all traffic conditions, as the systems scarcely conform with such simple models as the M/M/1.

At last, it is necessary to recall that "high load" is a notion to envisage carefully, at the network level, since the high load periods do not coincide in all parts of the network. This is why some of the standardisation organisms have specified overall performance objectives for normal load only. On the other hand, the two load levels are generally accounted at local level, see section 2.5.4.

2.4.3. *Overload*

This section addresses situations that can be said to be uncommon or exceptional. They are basically characterized by a level of the offered traffic greatly larger than the available network resources (also called engineered capacity). However, one can distinguish two kinds of such situations, namely the ones which can be forecast (such as feasts, religious or social events, TV games, etc.), and the very unpredictable events, related to accidents, natural catastrophes, etc.

The first kind of situation could be anticipated, to some extent, by relying on experience and past observations. But it would result in costly investments due to the unavoidable overdimensioning, and the network provider prefers simply using protective actions, accepting carrying the traffic with a lowered quality of service. One must emphasize here the effect of call repetitions (reattempts): the user having a demand rejected, retries several times, resulting in a dramatic growth of the call requests. A slight overdimensioning could have probably helped in avoiding this cumulative effect. In such circumstances, it is thus important to have the best possible estimation for the level of "fresh" demands (excluding reattempts). We will come back to this flooding effect typical of overload situations, and to models able of capturing the influence of retrials.

In the case of totally unpredictable events, such as catastrophes, no overdimensioning would be enough to get rid of the overload. The volume of traffic is unpredictable, and so is the portion of the network concerned by the phenomenon (as opposed for instance to congestion related to a TV game). Here again impatient behaviour and retrials amplify considerably the observed traffic on the network's directions and destinations.

These uncommon situations raise the question of performance, both for the network and for its components. Firstly, what is the problem if call requests in excess fail? The problem is that in case of congestion, *all* calls are subject to being rejected. Imagine the simple situation of calls rejected because of congestion in the destination node. For instance, some catastrophe provokes an abnormally high traffic in this node. All intermediate resources have been seized uselessly (circuits, processing capacity in switches or routers, signalling network), preventing other traffic flows to be processed normally. The situation can even be worse, in a IP network without any admission control, as all other sessions in progress are perturbed at the packet level.

Similarly, and this is a fundamental issue, imagine a processor (devoted to call establishment, or packet forwarding, etc.) which spends most of its time processing tasks for calls which fail further in the network, or which must be abandoned. At last, the processor becomes unavailable to process correctly any request (think of a physician, who would continue to auscultate while keeping answering calls to decline new patients). Here too, rules are needed to regulate the accepted traffic. This is what *overload control* or *regulation* mechanisms have to ensure.

The solution seems easy at a first glance: above a given threshold simply reject new demands. In fact, this is actually the basic principle, provided one can perform such denial, or *rejection*, at a minimum cost, and with the additional complexity related to the obligation of processing in priority certain types of calls, such as emergency numbers (police, fire brigades, etc.)

All these requirements have been specified, first in the international recommendations, then also, and more precisely in the operator-specific rules. We will present, later on, the content of the ITU recommendations. Let us survey here the set of common requirements that one can find.

In case of abnormal congestion, the goals are:

– Definitely prevent a total crash of the system or network. This seems obvious, but in these uncommon situations, defence mechanisms are intensely solicited, so that residual faults, normally harmless, can have dramatic effects. We develop in Chapter 10 some considerations about the test of such mechanisms.

– Maintain a system or network throughput not too far from the engineered capacity. Target values and principles are explained in the recommendations, as will be developed later on. These targets are attained mainly by rejecting calls in excess, provided the overload is rapidly detected. However, it must be balanced with the following goal:

– Give preference to the processing of certain categories of calls such as terminating calls (they have already used network resources and so must preferably succeed), urgent calls (fire brigade, police, etc.), priority lines, etc. The problem is that recognizing that a call pertains (or not) to one of these categories needs some preliminary processing, so to in most cases reject the demand. Obviously, an indication of priority (attached to the calling line, to the packet, etc.) is quickly analysed. But analysing the called number needs more resource. Even a very low "cost" of the analysis may lead to a total inefficiency, especially due to the avalanche phenomenon, which currently provokes 1000% overloads! These questions will be further developed when dealing with models, but notice at that point that even if priority calls can be detected, there may occur a persistent overload level such that all demands must be rejected blindly – as this is a vital question for the equipment or the network.

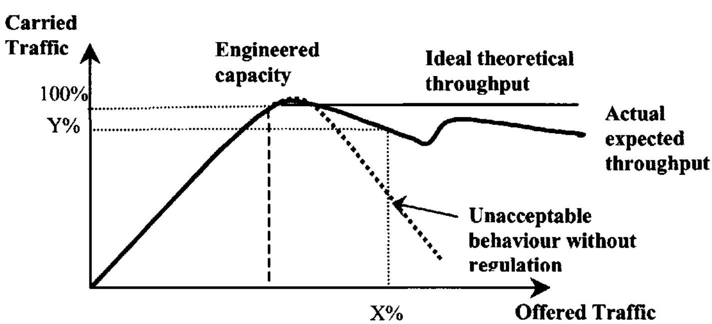


Figure 2.5. *Traffic carried in overload situation*

Figure 2.5 summarizes the possible behaviour of the system or the network, and complements the kind of picture one may find in ITU-T recommendations. It displays the variation of the carried traffic versus the offered load, taking account of the various rejection levels of the regulation mechanisms. The model is further addressed in Chapter 9.

2.5. Parameters and standards

The set of performance parameters evoked above is presented in several recommendations issued by the standardisation bodies already cited, such as ITU (International Telecommunication Union), ETSI (European Telecommunication Institute), or T1 (Telecommunication standardization organism for the USA), and IETF (Internet Engineering Task Force).

We give hereafter a survey of the main parameters and standards related to network performance. We do not pretend to give an exhaustive treatment, restricting to the most important parameters and standards those with which every system or network designer has to work. The world of standardisation is complex and ever changing. However, despite the variety of the organisations and the complexity of their structures, a certain convergence on the principles can be observed, even as technologies are changing. Therefore, the definitions and the references to the recommendations we quote remain inputs from which the reader will find easily the more recent contribution on the specific subject of interest.

In this spirit, most of the reference quoted hereafter concern ITU-T, as this organisation remains for the telecommunications sector the major "harmonization" forum, where the main concepts have been satisfactorily addressed, alleviating the difficult task of the reader, at least for a first contact with the world of standardization.

The general presentation of the performance parameters is in particular the scope of the ITU-T recommendations I.350 and E.430. Concerning IP, Rec. I.380 (or Y.1540) also defines a certain number of parameters, particularly for transfer.

In these recommendations one can find a general matrix presentation, expressing the correspondence between on the one hand those aspects related with the accessibility and retainability of the service, from the user standpoint, and on the other hand the network behaviour, in terms of traffic performance and dependability.

Table 2.1. *Performance parameters*

	Trafficability		Dependability
	Speed/delay	Accuracy/ integrity	Availability/ reliability
Access			
Transfer			
Disengagement			

Two main classes of performance parameters can be distinguished, the ones related to the "trafficability", i.e. the traffic performance, and those related to the dependability of the network and of its equipment.

2.5.1. *Trafficability*

2.5.1.1. *Access and disengagement*

During the access phase, speed and delay are the significant figures. Indeed, this phase corresponds, for the classical telephone service, to sending dial tones, receiving the number called, setting up the connection, ringing the called party, etc. For the new generations, it corresponds to session opening, and all those activities by which terminals are put in relation. Similarly, for the *release* (or disengagement) phase, the speed with which the resources are released is specified.

These operations have to be performed within reasonable delays so that the subscriber keeps a satisfactory perception of the service. The speed with which these operations are performed depends mainly on the capacity for exchanging messages between network elements, and on the duration of the processing these messages require in the control elements of the network. For either circuit or packet-switched modes, this concerns the performance of the *network control plane*, as described in ITU-T Rec. E.711, 712 and 713, as opposed to the *user plane*, devoted to transporting information between users.

The whole specification requires a rather large number of parameters, specified in many recommendations, standards, etc., of which only the most important part is enumerated here, as we are mainly concerned with "generic" aspects of the problem.

2.5.1.1.1. Overall (end-to-end) parameters

First we defined overall parameters and objectives related to the upper level of the network, from user to user. This is the purpose of standards such as for instance ITU recommendations I.352, or E.431, E.721 (fixed networks) and E.771 (mobile network) for circuit, and I.354, X.135, X.136 for packets, but also new recommendations for IP such as Y.1540 (or I.380), and Y.1541. These performance parameters are termed *end-to-end performance parameters*, as they characterize delays to be respected between user interfaces or network ends. Particularly, they are these parameters that are the most perceived by users. The following parameters are concerned:

Set-up delay. Call set-up delay is defined as the length of time that starts with the call or session set up attempt indication and ends with the return of the communication established indication. The reference events used in measuring set-up delay are occurrences of significant messages (*initial address message*, IAM, and *answer message*, ANM, for instance) or packets (*call request packet*, CR, and *call connected packet*, CC). Values are given in ITU recommendations I.352, I.354, E.721, E.771.

Clearing (disconnect) delay. Call clearing delay is defined as the time interval that starts with the clearing terminal demand and ends with the reception of the release indication by the other terminal. The reference events to be used for the measure are significant messages or packets as for the set-up phase (*disconnect and release messages*, *clear request packet*, *clear indication packet*). Values are given in ITU recommendations such as I. 352, I.354, E.771).

Blocking probability. End-to-end blocking probability is the probability that any call attempt will be "blocked", i.e. refused by the call acceptance mechanisms because of a lack of resource, or "abandoned" because of excessive set-up delays (leading for instance to abandonment by the user or rejection by the system) or loss of messages etc. Values are given in ITU recommendations E.721, E.771, Q543.

2.5.1.1.2. Intermediate parameters

Then are defined performance parameters at a lower level in the network, closer to the equipment: network portions, nodes, links, etc. The compliance of each element with the objectives specified for these parameters makes possible the compliance with the overall objectives specified for the preceding parameters.

In order to understand the logic of these parameters, the best approach is to follow how a call progresses throughout the network. This is illustrated in the previous diagrams in Figures 2.2 to 2.4.

Beginning from the left (calling user), the successful progression of the call is guaranteed by the following parameters:

Access call set-up delay. Access call set-up delay is defined as the interval from the instant that the information required for outgoing direction selection is available for processing in the access node, until the instant the set up demand has been passed to the subsequent node of the network. For instance for telephony, the significant information will be the subscriber dialling, a set-up message, or a SS7 signalling message such as the IAM (*initial address message*). Values are recommended in standards such as recommendations Q.543, E.721 from ITU.

Access call release delay. Call release delay at the access node is the time interval from the instant the release demand is received from the user terminal that terminated the call until the release message is returned to the same terminal, indicating that the terminal can initiate or receive a new call. The reference events to be used for the measure are messages or packets such as the *disconnect* or *release* message, *clear request packet, clear confirmation*... Values are recommended in standards such as recommendations Q.543, E.721, and I 354 from ITU.

Set-up message transfer time. Set-up message transfer time in an intermediate network element is the period starting when the set up message is received by the node, and ending when the message enters the outgoing signalling link for the first time. This is the case of a transit switch receiving the *initial address message* (IAM). The processing time for this message is much greater than when transferring any simple subsequent message, and one talks about processing intensive message. Values for this parameter are specified in recommendations devoted to signalling, such as ITU Rec. Q.766.

Message transfer time. It is the same delay as above but for simple messages without significant processing (such as subsequent n°7 messages), a simple routing function is needed. Values are recommended in Recommendation Q.766, and Rec Q.706 for n° 7 signalling transfer points.

Similar considerations apply to packet networks. The parameter is then the following:

Packet transfer delay. It is the same concept as above but for any packet, such as for instance a *call request* packet in the case of a virtual connection, or for subsequent packets. Packet transfer delay is the period needed to transfer successfully a packet across a network portion. Values are given in standards such as Recommendation X.135 (packet switched networks), and Recommendation Y.1541 (IP networks).

Incoming call indication sending delay. Incoming call indication sending delay is defined as the interval from the instant at which the called party identification is received at the terminating node, until the instant at which the call indication is passed to the called terminal. For instance, for telephony this corresponds to the time from the instant the called number is received, until the ringing signal is sent to the user terminal (fixed or mobile), by the terminating exchange. Values are given in standards such as recommendation Q.543 from ITU.

Answer transfer delay. Answer transfer delay at a node is defined as the interval from the instant the answer indication is received to the instant that the answer indication is passed toward the next network element. The answer signal corresponds to the indication that a communication path has been successfully established and that the user information transfer phase can start. A typical example is the ANS (*answer*) message for a transit exchange. In some way it is symmetrical to the IAM message, however with a shorter processing time, it is then classified as a simple message (as opposed to a processing intensive one). Values are given in recommendations Q.543 and Q.766 from ITU.

Signalling transfer delay. Signalling transfer delay is the time taken for a node to transfer a message from one signalling system to another (e.g. between user signalling and network signalling, or between two different networks). The interval is measured from the instant that the message is received from a signalling system, until the moment the corresponding message is passed to another signalling system. Values are given in Recommendation Q.543 and Q726 from ITU.

Transmission time. Transmission time includes delay due to equipment processing as well as propagation delay on the physical support (terrestrial or submarine by means of coaxial cable or optical fiber, or by air by means of radio relay or satellite). The transmission time is clearly a function of the distance and the speed of the signal in the transmission facility (especially for the propagation part). Values per unit of distance (generally the kilometre) and per type of support are given in recommendation G.114 from ITU.

Signalling message loss ratio. Signalling message loss ratio is the proportion of signalling messages that are lost at a node due to internal reasons, such as excessive delays, errors, failures etc. Values are given for n°7 signalling system in recommendation Q.706 from ITU.

Packet loss ratio. It is the same concept as just above, for packet. Values are given for IP packets in ITU recommendation Y.1541.

Set-up demand blocking probability at a node. It is the same parameter as for the overall performance, but for a node. Set up demand blocking probability is the proportion of demands that are blocked due to the acceptance control mechanisms because of the lack of resources, or abandoned because of excessive delays, loss of messages etc. Values are given in standards such as recommendation Q.543 from ITU.

Authentication delay, routing information acquisition delay. These parameters are specific to mobile telephony. Authentication delay is the time needed to achieve the authentication operation (processing time plus occasionally, depending on the system design, the access to a database). Routing information acquisition delay is the time needed for interrogating the HLR (*database lookup*), plus in case of roaming (see Chapter 1), a second data base access for interrogating the VLR. Values are given in recommendation E.771 from ITU.

2.5.1.2. *Communication, user information transfer*

Performance during this phase concerns the highest level of information exchanged between the users (voice samples, files, video images, etc.). Recommendations provide requirements in order to guarantee transmission quality mainly with respect to semantic and temporal integrity, i.e. concerning transmission delay, jitter, loss of information etc.

2.5.1.2.1. End-to-end parameters

As for the access and release phases these parameters aim to define a satisfactory overall level of perception by the user..

During the user's information transfer phase two main performance aspects are concerned, from the traffic handling viewpoint. These are, first, the end-to-end transfer *delay*, especially in relation to the user interactivity capacity and second, the quality of the transfer in terms of *interruption* probability, in relation with the *integrity* of the communication, and its *retainability* objectives, as we will see in the next section about dependability.

End-to-end transmission delay. End-to-end transmission delay is defined as the total transmission delay for information and includes delay due to equipment processing (switching or routing, etc.), as well as coding and packeting delay, and propagation delay. This delay should be kept under a given value in order to maintain a normal use of the information, especially from an *interactivity* point of view. For instance, for voice services an increase of the transmission delay beyond a certain value would cause unacceptable conditions for a conversation between subscribers (just think to the difficulty that one may sometimes experience in the case of a long distance international communication). It is with regard to this issue, and more particularly in the context of VoIP (*voice over IP*), that specific models have been developed in order to quantify the user perception as a function of the delay, such as the *E model* from recommendation G.107 with its associated measure, the MOS (*mean opinion score*). Values for the delay parameters are given in recommendations G.114 and G.131 from ITU.

Interruption transfer probability. Interruption transfer probability is defined as the probability of occurrence of short periods of time (less than a few second for instance), during which the transmission quality (power, noise level, bit error ratio, packet loss ratio etc.) decreases below certain thresholds (see below intermediate parameters). As long as these periods are short enough and rare, the user will perceive them as an acceptable service degradation (this is the reason why they are called interruptions). To the opposite when they are beyond a certain duration or a certain frequency, they will lead to what is called a premature release of the communication, as we will see below in the dependability section. Values for integrity are given for instance in recommendation E.855 from ITU, and in recommendation E.850 for premature release.

Handover failure probability. It is clearly a specific parameter for a mobile. One of the most important requirements of cellular systems is the ability to hand over communications *in progress* as the mobile user moves between cells. The handover failure probability is the probability that a handover fails because of a lack of resources in the target cell or in the network, or for any other reason such as excessive delays, insufficient signal strength etc. Values to be respected are given in ITU recommendation E.771.

2.5.1.2.2. Intermediate parameters

As for the access and disengagement phases, we now specify intermediate performance parameters, whose role is to help towards the construction of a network able to comply with the overall requirements, as just described above. Transfer phase and access or disengagement phases may have common or similar parameters. However, the value may be different depending on the phase considered (e.g. loss of information for voice during transfer phase is less important than for signalling during set-up phase).

Packet transfer delay. As already defined for the access phase, packet transfer delay is the period of time needed to successfully transfer a packet across a network portion. Values are given in standards such as recommendation X.135 (packet switched networks), and recommendation Y.1541 (IP networks). Values are also given for *cells* in the case of ATM technology, in recommendation I.356 from ITU.

Voice sample transfer delay (*round trip delay*). This parameter is defined in recommendation Q.551 for the important case of circuit switched telephony. Voice transfer delay, said also group transfer delay or round trip delay, is the delay to transfer a voice sample (or *group* of bits) through an exchange and for different network configurations (with concentrators or not).

Packet transfer delay variation. Packet transfer delay variation is defined as an upper bound, in practice a quantile, of the transfer delay distribution of the unit of information through the network or a network portion. This parameter is of first importance with the new technologies carrying real time services. The so-called sensitive packets of a same communication suffer from different delays, resulting in a *jitter*, which must be maintained within reasonable limits in order to fulfil the synchronisation constraints attached to these real-time services. For instance, voice transmitted using the G.711 standard requires one sample speech every 125 µs, imposing quite stringent constraints on the transfer delay through a packet network. Even if output buffers are able at compensating delay variations, they still must be limited, as this contributes to increase the end-to-end delay. Values are given in recommendations Y.1541 and I.356 from ITU.

Information loss probability. The probability of loss of information during the transfer phase is defined as the ratio of the number of transfer units, that are lost or above a given degraded threshold, to the total number of units to be transferred during the communication. The purpose is to guarantee the semantic integrity of the information during the transfer phase. Indeed transmission errors may occur (electrical perturbations or degradations...) as well a packets loss due to queue overflow, failures, misrouting etc. Concerning transmission errors, values are given in ITU recommendation G.821 and G.826 for parameters such as for instance severely errored seconds (SES), rate and duration (number of consecutive SES). Concerning packet or cell loss, one will refer again to recommendations Y.1541 and I.356.

2.5.2. *Dependability parameters*

For the access phase, the dependability performance parameters concern the probability that the user is able to access the network, i.e. that the equipments are available. In the communication phase, these parameters take account of the ability to make use of the service as long as needed, all equipments remaining continuously available during this duration.

The notions of reliability and availability will be presented, in a mathematical approach, in Chapter 6. At that point, the reader must understand that *reliability* represents the probability that an equipment or a set of equipments perform the required function during a given time interval T, and availability represents, for the same equipments, the probability to be in a state to perform the required function at any instant of time during a given time interval.

2.5.2.1. *Access*

2.5.2.1.1. Overall parameters

Network access probability (network accessibility). Network accessibility is the probability that a user of a service after a request receives a positive answer from the network, within specified delay, blocking tolerances, operating conditions such as defined above. This notion covers both traffic, reliability and availability aspects. The probability of success of a set-up demand is the global result of all these factors. Threshold are thus defined for delays (generally very long, greater than 10 seconds), blocking etc., beyond which the network is considered as unavailable. Accessibility probability measures and values are specified, either in terms of mean accessibility (see Rec. E.845) or in terms of distribution, called short term accessibility (see Rec. E.846). This latter concept as presented in Rec. E.846, is a little bit more complex than the average value, and allows one to specify a probability depending on the degree of unavailability. Indeed, a network may present more or less important perturbations in terms of number of isolated users, or duration. The more severe the perturbation the more its probability should be low (for instance very short unavailability periods are more easily accepted).

In most of the cases, in practice, one will distinguish only two situations: either the unavailability period is short (a few to ten seconds) and this period will not be counted in the measure, or it is longer and then it will be counted for the evaluation of the mean value (unavailability and accessibility).

End-to-end path availability. A path is said available if the whole set of network elements that are necessary for the success of a communication between two users is available. The path availability is defined as the proportion of time that the path is in an available state, i.e. all its elements are in an up state, over a given period (one year for instance). Values for this parameter are given in recommendations such as ITU Rec.G.827, G.827.1. Availability requirements for IP services may also be found in recommendation I.380 (or Y.1540) from ITU, which also specifies the following parameter:

Mean time between path outages. The mean time between path outages is the average duration of any continuous interval during which the path is available. Consequently, this parameter indirectly specifies the maximum admissible interruption frequency (i.e. the minimum reliability), but also a maximum level of maintenance activity. These latter performance aspects in complement to the availability allow one to characterize what is called the dependability performance.

2.5.2.1.2. Intermediate parameters

Just as for the traffic performance, these are parameters related to a segment of the network, to nodes, etc., having the purpose of building by steps a network fulfilling the overall performance requirements.

Access node availability. This is the same concept as for the path. Availability of the access node is expressed as the ratio of the accumulated time during which the node (or part of it) is capable of proper operation to a given time period of statistically significant duration. Taking account of the orders of magnitude (availability close to 100%), one rather specifies its complement, called unavailability. Moreover, different targets are specified for the unavailability of a single customer and for that of a group of customers. ITU-T Rec. Q.541 proposes typical values. This type of parameter, both with those concerning set-up delays, is among the most important ones, for the system design. It conditions the reliability of these equipments in direct relation to the customers, in very large numbers, and possibly imposes a certain amount of redundancy, which by comparison would not be needed in the core network, where multiple routes provide the necessary protection.

Path segment availability. The corresponding objectives are *apportionments*, or *budget*, of the overall objectives as already defined, to portions of the network (national, international...). Values to be considered are given in recommendations G.827, G.827.1 and I.380 from ITU.

Mean time between failures for a segment or a node. As for the whole network, the number of failures must be limited, as well as the maintenance effort. As these constraints are mostly operator-dependent, they are not specified in the international standards. However, they are expressed in tenders issued to equipment providers. Examples will be discussed in Chapter 10.

Logistic delays. Logistic delays correspond to travel times and repair times in case of failures. Here too, such constraints are operator specific, and the international standards do not provide any instruction with regard to them. However, we show, in Chapters 6 and 10, the fundamental importance of these parameters in fulfilling availability targets. Clearly, these constraints have quite a different impact, according to the repair delays the operator tolerates, from a few hours to a full weekend.

2.5.2.2. *Transfer and disengagement*

The two phases are here associated, as once the communication is established the only dependability-related constraint is that the communication is successful up to the instant chosen by the user to close it.

The parameters that are defined here are of the domain of reliability, as described previously, and deal with what is called the *retainability* (or *service continuity*).

2.5.2.2.1. Overall performance

Premature release probability. Premature release probability is defined as the probability that an established communication will be prematurely released consequently to a network or network element malfunction, that is to say for a reason other than intentionally by any of the users involved in the communication. Values are given in recommendations such as ITU Rec.E.850.

2.5.2.2.2. Intermediate parameters

Premature release probability. This is the same concept as above but applied to network segments or equipment. Values to be allocated to national and international segments are specified in recommendation E.580. Concerning the access node one may refer to ITU Rec.Q.543.

2.5.3. *Performance during overload*

These parameters are neither explicitly mentioned in the 3x3 matrix of I.350, nor in the E series recommendations. Indeed, they concern neither equipment unavailability nor normal load or high load performances, but rather the traffic processing capacity when the network or some of its equipment are submitted to severe overloads, e.g. in case of catastrophes or TV games, etc.

The performance level required in these situations calls for special attention, and they are described hereafter, based upon the chapter devoted to this point in Rec. Q.543. The recommendation deals actually with equipment, but the principles remain valid at the network global level; examples of applications are discussed in Chapter 9.

Processing capacity during overload. The processing capacity of a network (or network element) during overload is defined as the traffic that can successfully process a network, or a network element (switching exchange, softswitch-call server, router, server, gateway...) when the traffic offered exceeds its engineered capacity, i.e. its available (installed) processing capacity. The fundamental requirement is first to react sufficiently quickly to overload and second to continue to process an amount of traffic not too much below the engineered capacity, even during very high levels and very long periods of overload. The region of correct behaviour during overload is specified by means of two values as follows: the throughput at an overload of Y% above the engineered capacity load should be at least X% of the throughput at engineered capacity. Typical values for X and Y are respectively 50% and 90%. (for instance, a network element with installed capacity is 100 call/s, when offered call attempts rate is 150 per second should handle at least 90 call/s). Beyond these values it is simply required that the throughput should be 'acceptable'. The expected behaviour is depicted by a curve as in Figure 2.6.

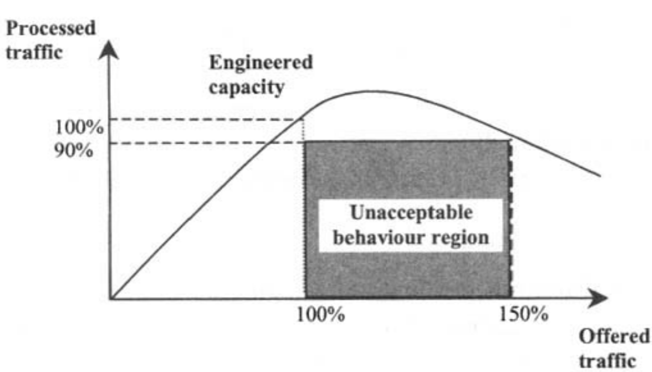


Figure 2.6. *Throughput performance during overload*

In order to achieve this objective, the following principles are recommended:

– implement efficient overload detection mechanisms;

– implement priority mechanisms for the call acceptance and the call rejection during overload. For instance preference will be given to terminating calls (or, more generally speaking, incoming demands) compared to originating calls (in order to ensure the success of demands which have already been accepted in the network and have consumed resources). Priority will also be given to the acceptance of priority class lines (hospitals, ministries), and calls to priority destinations (fire brigade, police etc);

– implement at the processing level priorities between tasks in order to give preference to calls or sessions already accepted, before accepting new ones;

– guarantee an operation (supervisory, measurements) as normal as possible, and particularly maintain the charging functions.

Obviously, all these requirements have a strong impact on the performance analysis of equipment and networks, as all these mechanisms must be integrated in the performance study.

Performance during overload. Clearly, in most of the cases the overall grade of service perceived by the user will seriously deteriorate during severe overload conditions, particularly because a large number of demands in excess will be rejected by the protection mechanisms. However, accepted demands should receive an acceptable grade of service, but not necessarily the same as under normal condition. For instance, delays should be not too far from those specified under high load or load B conditions in order to guarantee a minimal throughput.

2.5.4. *Synthesis tables*

The following tables synthesise the previous considerations, by giving the main performance parameters, and the typical associated values. These values are only indications, although they provide typical orders of magnitude, which ought to be observed, whatever the network and technology, as they derive directly from the perception users may have of delays and information loss.

2.5.4.1. *Trafficability*

Parameter	Normal load		High load	
	Mean	95%	Mean	95%
Set-up delay for a national call (E.721)	5000ms	8000 ms	7500 ms	12 000 ms
Set up delay at the access for an outgoing call Q.543)	600 ms	800 ms	800 ms	1200 ms
Incoming call indication sending delay (Q.543)	650 ms	900 ms	1000 ms	1600 ms
Clearing delay for a national call (I.352)	1250ms	1750 ms	Not specified	Not specified
Release delay at the access Q.543)	250 ms	300 ms	400 ms	700 ms
Blocking probability: – national call – international call (E.721)	 3% 5%	Not applicable	 4,5% 7.5%	Not applicable
Blocking probability at the access: – originating call – terminating call (Q.543)	 0.5% 0.5%	Not applicable	 3% 3%	Not applicable
Transfer delay at a node for an new call demand signalling message (Q.766)	180 ms	360 ms	450 ms	900 ms
Transfer delay at a node for a voice sample (Q.551)	0.9 ms	1.5 ms	Not specified	Not specified
End-to-end transfer delay for an IP packet: – real time – file (Y.1541)	 100 ms 1000ms	Not specified	Not specified	Not specified
Transfer delay variation for an IP packet (10^{-3} quantile): – real time (Y.1541)	50 ms	Not specified	Not specified	Not specified
End-to-end IP packet loss probability (Y.1541)	10^{-3}	Not applicable	Not specified	Not applicable
Handover failure probability (E.771)	0.5%	Not specified	Not specified	Not specified

2.5.4.2. *Dependability parameters*

Mean network unavailability (E.846)	$6\ 10^{-2}$
Unavailability at the access nod for: – a user – a large group of users (Q541 and network providers)	 30 mn/year 3 mn/year
Premature release probability for an international communication (E850): – typical configuration – worst case	 $4\ 10^{-4}$ $1.6\ 10^{-3}$
Premature release probability at a node (Q543)	$2\ 10^{-5}$
Maintenance load at an access node (network providers)	< 15 failure per year and per 10 000 users
Logistic delays (network providers): – immediate – postponed – without urgency	 3.5 hours 12 hours 72 hours

This page intentionally left blank

3

Probability

In this chapter we introduce the fundamental concepts of probability theory. Indeed, performance analysis is mainly concerned with phenomena which are basically unpredictable: examples are the occurrence of equipment breakdowns, the arrivals of connection requests and their durations. These *random phenomena* may be well described however, on a statistical basis, by using the tools and methods of *probability theory.*

We present the notions and properties of random variables, and the major theorems founding the theory. We develop also the properties of the *transforms* of distribution functions, which appear as essential tools in solving complex probabilistic problems.

3.1. Definition and properties of events

3.1.1. *The concept of event*

Probability theory is mainly built on the concept of *random event.* The particular outcome of *this* experiment cannot be predicted, although the set of all possible outcomes is known: any particular outcome will be called a "random event".

The concept of random event is quite natural. Consider the classical example consisting in tossing a die: the outcome is one of the six faces coming up, numbered 1, 2, 6. "I toss the die and 3 comes up" is an event. Each such occurrence is an *elementary event.* Six different outcomes are possible corresponding to any one of the six faces of the die coming up. Nothing is known about the next outcome, except that it belongs to a certain set, which one is usually able to describe. The set of all the possible outcomes (here, the six possible faces coming up) is the *event space*, or the *sample space*, and is usually denoted as Ω.

The formalism of set theory is useful in describing events and their properties. Indeed, the properties of events may be stated in terms of the classical operations of the set theory. This last provides methods to analyze combinations of events in order to predict properties of more complex ones. The terminology and correspondence are summarized below in Table 3.1.

Table 3.1. *Events and sets*

Event	Any subset of the sample space
Sure event	Ω the sure event is the occurrence of anyone of the elements of Ω
Impossible event	$\varnothing$ the impossible event is the occurrence of an event which is not an element of Ω, thus belonging to the empty set $\varnothing$. $\varnothing$ does not contain any elementary event ω
Elementary event	$\omega \in \Omega$, ω belongs to Ω
Compound event	A, is a subset of Ω. One denotes that the elementary event ω belongs to set A by $\omega \in A$
Complementary event	$\overline{A}$, is the complement of A (the event A does not occur)
A or B	$A \cup B$, union of A and B. The union of sets A and B contains all the elements which belong *at least* to one of the two sets.
A and B	$A \cap B$, intersection of A and B. The intersection of sets A and B contains all the elements which belong *both* to A and B
A and B mutually exclusive	$A \cap B = \varnothing$. There is not any event common to A and B
A equivalent to B	$A = B$, that can be read A equals B

All these definitions are in fact quite natural. Consider again the experiment about tossing a die, "3 comes up" is an elementary event, while "I draw an even number" is a compound event. Similarly, "I draw an odd number" is the opposite event to the previous one, and "7 comes up" is an impossible event, etc. The experiment could be extended, for instance when tossing two dice: the outcome "the sum of the two faces coming up is 6" is a compound event, etc.

From these basic definitions, we are now in a position to introduce the following definitions and properties, themselves quite intuitive.

3.1.2. *Complementary events*

Let A stand for an event (i.e. a subset of the sample space Ω). $\overline{A}$ denotes the complementary event (it contains all the elementary events which are not in A). The following relations hold and have an immediate and intuitive understanding.

Basic properties

- $A \cup \overline{A} = \Omega$.
- $\overline{A} = \Omega - A$ (the symbol "–" denotes the *difference*, sometimes called the *reduction of* Ω *by* A).
- $A \cup \Omega = \Omega$; $A \cap \Omega = A$.
- $A \cap \varnothing = \varnothing$, A being a subset of Ω, there is no element of A in the empty set.
- $A \cap \overline{A} = \varnothing$, A and its complement have no common element.
- $A \cup A = A$, $A \cap A = A$.

And also:

$\overline{\Omega} = \varnothing$, $\overline{\varnothing} = \Omega$,

$\overline{\overline{A}} = A$, the complement of the complement of A is A itself.

3.1.3. *Properties of operations on events*

3.1.3.1. *Commutativity*

Union and intersection are commutative. The proof is straightforward starting from the definitions:

$A \cup B = B \cup A$, $A \cap B = B \cap A$.

3.1.3.2. *Associativity*

Union and intersection are associative. Let A, B and C be subsets of Ω:

$A \cup (B \cup C) = (A \cup B) \cup C$, $A \cap (B \cap C) = (A \cap B) \cap C$.

3.1.3.3. *Distributivity*

Union is distributive over intersection, and intersection is distributive over union. The proof is straightforward starting from commutativity and associativity properties.

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C), \ A \cap (B \cup C) = (A \cap B) \cup (A \cap C).$$

3.1.3.4. *Difference*

The difference between A and B is composed of all the elements in A that do not belong to B:

$$A - B = A \cap \overline{B}. \tag{3-1}$$

If B is included into A, then $A - B$ is said to be the *complement* of B *with respect* to A.

3.1.3.5. *De Morgan rules*

$$\overline{A \cap B} = \overline{A} \cup \overline{B} \tag{3-2}$$

$$\overline{A \cup B} = \overline{A} \cap \overline{B} \tag{3-3}$$

Consider the first rule: the proof is straightforward using the distributivity and commutativity properties:

$$(A \cap B) \cup (\overline{A} \cup \overline{B}) = (A \cup \overline{A} \cup \overline{B}) \cap (B \cup \overline{A} \cup \overline{B}) =$$

$$((A \cup \overline{A}) \cup \overline{B}) \cap ((B \cup \overline{B}) \cup \overline{A}) = (\Omega \cup \overline{B}) \cap (\Omega \cup \overline{A}) = \Omega,$$

thus $(\overline{A} \cup \overline{B}) = \overline{(A \cap B)}$, as $(A \cap B) \cap (\overline{A} \cap \overline{B}) = \varnothing$.

The second rule is directly derived from the first one:
$\overline{(\overline{A} \cap \overline{B})} = \overline{(\overline{A})} \cup \overline{(\overline{B})} = A \cup B$, thus $\overline{A} \cap \overline{B} = \overline{A \cup B}$.

Note: These rules are easily *verified* by building the corresponding so called Venn diagram. We will use below this type of diagram to illustrate some properties of probabilities (see e.g. Figure 3.1).

These rules offer a systematic way to deal with complex logic relations. They happen to constitute the basic design techniques for logic circuits such as the ones encountered in automatism and computer systems. In the following, the event algebra will be a preferred tool for the analysis of link systems (see Chapters 6 and 9). This is naturally the foundation of probability theory, as explained hereafter.

3.2. Probability

3.2.1. *Definition*

We are given a phenomenon that is observed through the occurrence of events denoted *A, B*, etc. The concept of probability will be defined on the set Ω of all the possible events. The probability of an event *A*, denoted by *P*(*A*), is a positive number verifying the following conditions:

1. $0 \leq P(A) \leq 1$;

2. $P(\Omega) = 1$, the probability of the "sure event" is unity;

3. $P(\varnothing) = 0$, the probability of the "impossible event" is zero;

4. If *A* and *B* are two exclusive events (i.e., $A \cap B = 0$), then $P(A \cup B) = P(A) + P(B)$.

The mathematical concept of probability of an event corresponds to the relative *frequency of the event* during a sequence of experimental trials. One considers that this relative frequency goes to a limit, which is the probability of the event, as the number of trials increases to infinity.

Let *n* be the number of times the event *A* occurs during an experiment of *N* trials, then:

$$P(A) = \lim_{N \to \infty} \frac{n}{N}. \tag{3-4}$$

3.2.2. *Basic theorems and results*

3.2.2.1. *Addition theorem*

Given two events *A* and *B* from the set Ω,

$$P(A \cup B) = P(A) + P(B) - P(A \cap B). \tag{3-5}$$

The above theorem results directly from properties of operations on event as we have just seen. Indeed:

$$A \cup B = A \cup (B - (A \cap B)).$$

The sets A and $(B - (A \cap B))$ are mutually exclusive, thus:

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

This result can be intuitively understood if one thinks on the way to count the events in the different subsets: in summing the number of elements in A and in B, the elements of $A \cap B$ are counted twice.

In applied probability the following symbolism is generally adopted:

$$P(A \cup B) = P(A + B), \; P(A \cap B) = P(AB),$$

so that the basic expression is often written as:

$$P(A + B) = P(A) + P(B) - P(AB). \tag{3-6}$$

It is easy to extend the result to the general case of n events:

$$P(A_1 + A_2 + \cdots A_n) =$$

$$\begin{aligned} &P(A_1) + P(A_2) + \cdots P(A_n) - P(A_1 A_2) - \cdots P(A_{n-1} A_n) + P(A_1 A_2 A_3) + \cdots \\ &+ (-1)^{n-1} P(A_1 A_2 \cdots A_n) \end{aligned} \tag{3-7}$$

This is the so-called Poincaré's theorem.

And, if the events are mutually exclusive:

$$P(A_1 + A_2 + \cdots A_n) = P(A_1) + P(A_2) + \cdots P(A_n) \tag{3-8}$$

3.2.2.2. *Conditional probability*

The notion of conditional probability is of central importance. It formalizes the idea according to which the knowledge about an event is modified by any information one can acquire about the system. To give an elementary example:

consider throwing two dice. “I bet the faces coming up sum to 12”. What is the probability I am right? If I know nothing about the trial, the event is given probability 1/36 (36 possible outcomes and only one favourable: both dice coming up with 6). Now, assume I am told the first trial gives 3. For sure I lose. If on the other hand the first throw gives 6, the probability of a win is 1/6 (I only need that the second throw comes up with 6, in order to win).

This illustrates the notion. It is commonly denoted as $P(B/A)$, which is read “probability of B given (or knowing) A”. Its expression is defined as:

$$P(B/A)=\frac{P(A.B)}{P(A)}. \qquad (3\text{-}9)$$

The following Venn diagram serves as verifying this property, in an example where the knowledge about the realization of the event B is dependent on the realization of A. In this example, A contains n_a = 5 elementary events, B contains n_b = 4 elementary events, of which 3 are common with A (n_{ab} = 3). The total number of outcomes is N = 10:

$P(AB)$ = 3/10, $P(A)$ = 5/10, $P(B/A)$ = 3/5 = 6/10, $P(B)$ =4/10, and $P(A)P(B)$ = (5/10)(4/10) = 2/10.

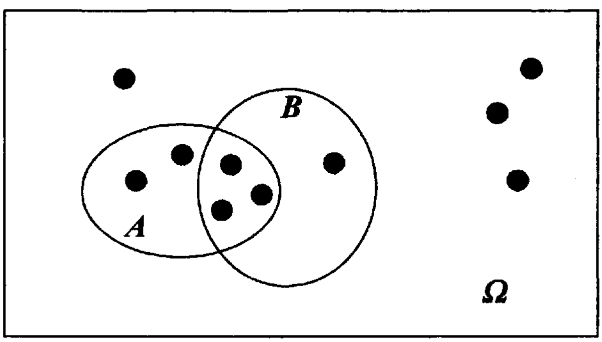


Figure 3.1. *Venn diagram illustrating the notion of conditional probability*

Such a situation corresponds, e.g. to the probability of finding, among 10 persons taken at random, boys (event A, probability around 0.5) taller than some value (event B of probability 0.4 in the example). This should illustrate the property according to which boys are likely to be taller than girls.

3.2.2.3. *Multiplication theorem*

Given two events A and B from the sample space Ω:

$$P(AB) = P(A)P(B/A). \tag{3-10}$$

This is merely the preceding result under its product form.

Similarly we have:

$$P(AB) = P(B)P(A/B) \tag{3-11}$$

Generalization to n events

$$P(ABC) = P(A)P(B/A)P(C/AB).$$

Indeed:

$$P(ABC) = P(AB)P(C/AB) = P(A)P(B/A)P(C/AB),$$

and thus, also:

$P(C/AB) = \dfrac{P(ABC)}{P(AB)}$, and so on for n events.

Independent events

Two events are said to be *independent* if:

$$P(A/B) = P(A).$$

Knowing that B has happened brings no special knowledge about A: the occurrence of B (resp. A) does not affect the probability of the occurrence of A (resp. B). Then:

$$P(AB) = P(A)P(B). \tag{3-12}$$

Coming back to our previous example, assume we are interested in the colour of the eyes. One can admit that this colour is independent of gender. Assume a proportion of 40% of blue eyes among girls as well as boys. When 10 persons are taken at random, one should observe (on average) $n_a = 5$ (5 boys), $n_b = 4$ (4 blue eyed persons), and $n_{ab} = 2$.

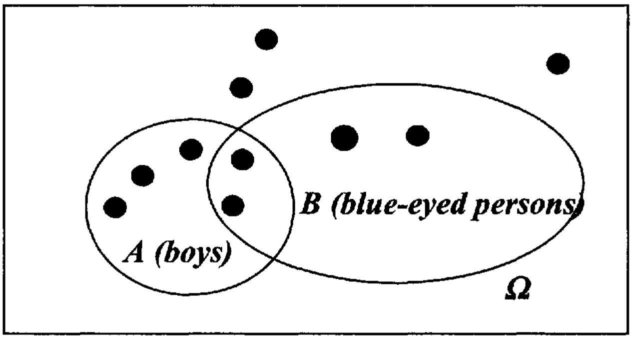


Figure 3.2. *Independent events*

Then $P(A) = 5/10$, $P(B/A) = 2/5$, $P(AB) = 2/10$, equal to $P(A)P(B) = (5/10)(4/10) = 2/10$ (compare with the previous case, Figure 3.1).

More generally, events A_i are said to be mutually independent if, for any combination A_i, A_j:

$$P(A_1A_2..A_i..A_n) = P(A_1)P(A_2)\cdots P(A_i)\cdots P(A_n).$$

3.2.2.4. *The a posteriori probability theorem*

Let an event A be associated with one of the B_i, mutually exclusive. B_i is said to be a cause for A. Assume one knows the set of conditional probabilities, $P(A/B_i)$. The conditional probability of B_i knowing A, expresses the probability that B_i is the cause of A. This probability is said "*probability a posteriori*" *of the event* B_i.

First, the relation:

$$P(A) = \sum_{i=1}^{n} P(B_iA),$$

simply states that if the event occurs it is necessarily associated to a cause which is one of the B_i (which are mutually exclusive: only one of them may occur). Moreover, introducing conditional probabilities:

$$P(B_iA) = P(B_i)P(A/B_i),$$

the relation may be written under the form known as the *theorem of total probability*:

$$P(A) = \sum_{i=1}^{n} P(B_i) P(A / B_i). \tag{3-13}$$

Using once again conditional probabilities, $P(B_i / A) = \dfrac{P(B_i A)}{P(A)}$,

one may state the following result, known as *Bayes'formula* (or theorem of *probability a posteriori*):

$$P(B_i / A) = \frac{P(B_i) P(A / B_i)}{\sum_{j=1}^{n} P(B_j) P(A / B_j)}. \tag{3-14}$$

The following example will illustrate this result. Three urns U_1, U_2, U_3, contain red and black balls. There are 2 black balls and 1 red in U_1, 5 black and 1 red in U_2, 1 black and 5 red in U_3. A black ball has just been drawn, what is the probability it comes from U_3? If B stands for the event "a black ball is selected", then:

$$P(U_3 / B) = \frac{P(U_3) P(B / U_3)}{\sum_{j=1}^{3} P(U_j) P(B / U_j)}$$

One assumes all urns are equally likely to be chosen:

$P(U_1) = P(U_2) = P(U_3) = 1/3$.

Also, $P(B/U_1) = 2/3$, $P(B/U_2) = 5/6$, $P(B/U_3) = 1/6$,

and thus: $P(U_3/B) = (1/3)(1/6)/((1/3)(2/3+5/6+1/6)) = 0.1$

3.3. Random variable

3.3.1. *Definition*

A random variable is a function which associates a real number x with the realization of a random event. More formally, a random variable X is an application X of Ω into R, the set of the real numbers. For instance, the height of the first person to enter in my office is a random variable, and so is the duration of the next telephone call that I will receive.

3.3.2. *Probability functions of a random variable*

3.3.2.1. *Notations*

As before, let denote as ω an event from Ω. We write for short $X = x$ for the event $\{\omega : \omega \in \Omega \text{ and } X(\omega) = x\}$.

Similarly, we write $X \le x$ for the event $\{\omega : \omega \in \Omega \text{ and } X(\omega) \le x\}$.

3.3.2.2. *Distribution function or cumulated probability*

The distribution function of the random variable X is the function:

$$F(x) = P[X \le x]. \tag{3-15}$$

This function is non-decreasing, and such that $F(\infty) = 1$.

Now, $F(y) = P[X \le y]$, then, if $x < y$, $F(x) \le F(y)$.

And also:

$$P[x < X \le y] = F(y) - F(x). \tag{3-16}$$

3.3.2.3. *Probability density function*

Given a random variable X, with distribution function F, one may wonder about the probability that X takes some value. This leads to the following distinction: two types of random variable must be distinguished, the *discrete* random variable and the *continuous* random variable.

The random variable X is said to be discrete if:
$\sum_{x \in T} p(x) = 1$, with $p(x) = P[X = x]$, its *probability mass function*, and T the set of all the possible values x for X. T is either finite or denumerable (countably infinite), i.e. T consists of a finite or countable set of real numbers $x_1, x_2, \ldots$

Now, for a continuous random variable, the probability of observing a given value is null, as the variable X can take all the possible values x between two given values a and b. The set T of the possible values x for X is then said non denumerable. So, the only statement making sense is "the value is between x and y". This leads to introducing the notion of density.

The *probability density function* $f(x)$ of a random variable X is such that

$$\int_{-\infty}^{x} f(x)dx = F(x). \tag{3-17}$$

Particularly we have:

$$\int_{-\infty}^{\infty} f(x)dx = 1, \tag{3-18}$$

and for two given values a and b, if $a < b$:

$$P[a \le X \le b] = \int_{a}^{b} f(x)dx.$$

Notice in particular that for a continuous variable $P[X = x] = 0$.

Thus:

$$P[a \le X \le b] = P[a < X \le b] = P[a < X < b] = F(b) - F(a), \tag{3-19}$$

and also:

$$P[x < X \le x + dx] = F(x + dx) - F(x).$$

$$f(x) = \frac{dF(x)}{dx}. \tag{3-20}$$

3.3.3. *Moments of a random variable*

The moments of a random variable give a simple and synthetic characterization of its behaviour. As a consequence they also make easier studying combinations of several independent variables, as it will be illustrated below.

3.3.3.1. *Moments about the origin*

The n^{th} order moment of a random variable about the origin is, for a discrete variable:

$$m_n = \sum p_k x_k^{\ n} , \tag{3-21}$$

and for a continuous variable:

$$m_n = \int x^n f(x) dx . \tag{3-22}$$

3.3.3.2. *Central moments*

The n^{th} central moment of a random variable is, for a discrete variable:

$$\mu_n = \sum p_k (x_k - m_i)^n , \tag{3-23}$$

with m_i the i^{th} moment about the origin.

And for a continuous variable:

$$\mu_n = \int x^n f(x - m_i)^n dx . \tag{3-24}$$

In practice central moments will be generally computed about the *mean*, such as defined below.

3.3.3.3. *Mean and variance*

In the application of probability theory the two first moments of a random variable, namely its mean and variance, are of particular importance.

3.3.3.3.1. Mean (expectation) of a random variable

The *mean* (or *expected value*) of a random variable is the first order moment about the origin, and is conventionally denoted m.
For a discrete variable:

$$m = E[X] = \sum_{k=1}^{n} p_k x_k . \tag{3-25}$$

For a continuous variable:

$$m = E[X] = \int_{-\infty}^{\infty} x f(x) dx . \tag{3-26}$$

The mean is also denoted as:

$$E[X] = \overline{X} .$$

Intuitively, one can easily understand the physical meaning of the expectation: it is the weighted value of the set of all possible outcomes. In a way this value summarizes the result of a great number of experiments.

3.3.3.3.2. Variance of a random variable

The *variance* of a random variable is the second order moment about the mean, and is conventionally denoted V or σ^2. For a discrete variable:

$$\mu_2 = \sigma^2 = \sum_{x_k} p_k (x_k - m)^2 . \tag{3-27}$$

For a continuous variable:

$$\mu_2 = \sigma^2 = \int_{-\infty}^{\infty} (x - m)^2 f(x) dx . \tag{3-28}$$

σ is called the *standard deviation.*

This quantity is also of evident interest, as it takes account of the variation of the realizations around the mean – i.e. the *dispersion*. As its value increases, larger deviations from the mean are likely to occur.

3.3.3.3.3. Properties of the variance

Applying the definition of the variance, we have:

$$\sigma^2 = E(X - E(X))^2 = \overline{(X - \overline{X})^2} .$$

Developing the squared term:

$$\sigma^2 = \overline{X^2 - 2X\overline{X} + \overline{X}^2} ,$$

and thus, (the expectation of a sum is the sum of expectations):

$$\sigma^2 = \overline{X^2} - \overline{2X\overline{X}} + \overline{\overline{X}^2},$$

$$\sigma^2 = \overline{X^2} - 2\overline{X}\,\overline{X} + \overline{X}^2,$$

and finally:

$$\sigma^2 = \overline{(X - \overline{X})^2} = \overline{X^2} - \overline{X}^2. \tag{3-29}$$

In other words one has the important relationship: the variance is equal to the mean of the squares minus the square of the mean. This result will be of frequent use throughout this book. Particularly note here that from (3-29) we derive $\sigma^2(X/N) = \sigma^2(X)/N^2$.

3.3.3.4. *Examples of application*

First, let us compute the mean of a variable obeying a Poisson distribution, which will be further detailed in Chapter 4.

Let X be a random variable whose possible values are 0, 1, 2...k, with probability p_k:

$$X = \{0,1,2...k,...\}, \text{ and } P(X = k) = p_k = \frac{\lambda^k}{k!}e^{-\lambda}.$$

This is the Poisson distribution.

For the mean, we have:

$$\overline{X} = \sum k p_k = 0e^{-\lambda} + \lambda e^{-\lambda} + 2\frac{\lambda^2}{2!}e^{-\lambda} + \cdots k\frac{\lambda^k}{k!}e^{-\lambda} + \cdots,$$

$$\overline{X} = \lambda e^{-\lambda}(1 + \lambda + \frac{\lambda^2}{2!} + \cdots \frac{\lambda^{k-1}}{(k-1)!} + \cdots) = \lambda e^{-\lambda}(e^{\lambda}),$$

that is: $\overline{X} = \lambda$. (3-30)

For the variance:

$$\sigma^2 = \overline{X^2} - \overline{X}^2.$$

Now, we have:

$$\overline{X}^2 = \lambda^2,$$

and also:

$$\overline{X^2} = \sum k^2 p_k = 0e^{-\lambda} + \lambda e^{-\lambda} + 2^2 \frac{\lambda^2}{2!} e^{-\lambda} + \cdots k^2 \frac{\lambda^k}{k!} e^{-\lambda} + \cdots.$$

Observing that:

$$\frac{k^2}{k!} = \frac{k(k-1)+k}{k!} = \frac{k(k-1)}{k!} + \frac{k}{k!} = \frac{1}{(k-2)!} + \frac{1}{(k-1)!},$$

one gets:

$$\overline{X^2} = \lambda^2 e^{-\lambda}\left(1 + \lambda + \frac{\lambda^2}{2!} + \cdots \frac{\lambda^{k-2}}{(k-2)!} + \cdots\right) + \lambda e^{-\lambda}\left(1 + \lambda + \frac{\lambda^2}{2!} + \cdots \frac{\lambda^{k-1}}{(k-1)!} + \cdots\right),$$

i.e. $\overline{X^2} = \lambda^2 + \lambda$.

Thus, finally:

$$\sigma^2 = \lambda. \tag{3-31}$$

This result is worth remembering: in the case of a random variable obeying a Poisson distribution with parameter λ, its mean and its variance are equal, and equal to λ.

3.3.4. *Pairs of random variables*

We now consider the case of two random variables such as defined before. Actually we now introduce the properties of a function of several random variables: the results obtained for two variables will be easily extended to the general case of many variables. In particular, of prime importance for the following are the results about the sum of random variables.

3.3.4.1. *Definition*

A pair of random variables is the set of two random variables, each having its own domain of variation and probability distribution.

As an example, consider tossing two coins (numbered 1 and 2). The result is a couple of variables, each having values "Heads" or "Tails". Denoting as H_1, T_1 and H_2, T_2 the possible realizations of the individual trials, the experiment results in four different outcomes H_1H_2, H_1T_2, T_1H_2, T_1T_2, each one being equally likely so that the probability of each outcome is ¼.

3.3.4.2. *Joint probability*

3.3.4.2.1. Joint distribution

Let X and Y two random variables defined on Ω. The joint distribution function of the pair X, Y is the function:

$$F(x,y) = P\{X \le x; Y \le y\}, \tag{3-32}$$

also denoted as $F_{x,y}$.

As in the case of the single random variable one can develop:

$$P\{x_1 < X \le x_2; y_1 \le Y \le y_2\}.$$

We have:

$$P\{X \le x_2; Y \le y\} = P\{X \le x_1; Y \le y\} + P\{x_1 \le X \le x_2; Y \le y\}.$$

We write first:

a) $P\{x_1 \le X \le x_2; Y \le y\} = P\{X \le x_2; Y \le y\} - P\{X \le x_1; Y \le y\}$.

Then in a same way:

$$P\{x_1 \le X \le x_2; Y \le y_2\} = P\{x_1 \le X \le x_2; Y \le y_1\}$$
$$+ P\{x_1 \le X \le x_2; y_1 \le Y \le y_2\},$$

and then:

b) $P\{x_1 \le X \le x_2; y_1 \le Y \le y_2\} = P\{x_1 \le X \le x_2; Y \le y_2\}$

$$-P\{x_1 \le X \le x_2; Y \le y_1\}.$$

Hence:

$$P\{x_1 \le X \le x_2; y_1 \le Y \le y_2\} = (F_{x2,y2} - F_{x1,y2}) - (F_{x2,y1} - F_{x1,y1}),$$

and finally:

$$P\{x_1 \le X \le x_2; y_1 \le Y \le y_2\} = F_{x2,y2} - F_{x1,y2} - F_{x2,y1} + F_{x1,y1}. \quad (3\text{-}33)$$

3.3.4.2.2. Joint density function

The joint density function of two variables X and Y is the function:

$$f(x,y) = \frac{\partial^2 F(x,y)}{\partial x \partial y}. \quad (3\text{-}34)$$

3.3.4.2.3. Fundamental property of the joint density

$$P\{x \le X \le x+dx; y \le Y \le y+dy\} = d^2F(x,y) = f(x,y)dxdy \quad (3\text{-}35)$$

Indeed, from the above result for the joint distribution:

$$P\{x \le X \le x+dx; y \le Y \le y+dy\} = F(x+dx, y+dy)$$
$$-F(x+dx,y) - F(x,y+dy) + F(x,y),$$

$$P\{x \le X \le x+dx; y \le Y \le y+dy\} = dF(x+dx,y) - dF(x,y)$$

$$P\{x \le X \le x+dx; y \le Y \le y+dy\} = d^2F(x,y).$$

Hence:

$$P\{x \le X \le x+dx; y \le Y \le y+dy\} = d^2F(x,y) = f(x,y)dxdy.$$

Consequently:

$$P\{x_1 \le X \le x_2; y_1 \le Y \le y_2\} = \int_{x1}^{x2}\int_{y1}^{y2} f(x,y)dxdy.$$

3.3.4.3. *Marginal probability of a pair of random variables*

3.3.4.3.1. Marginal distribution function

The distribution of each variable of a pair of variables is called *marginal*. The marginal distributions of X and Y are respectively the functions:

$$F_x = P\{X \le x; -\infty \le Y \le \infty\}, \tag{3-36}$$

$$F_x = \int_{-\infty}^{\infty}\int_{-\infty}^{x} f(x,y)dxdy,$$

and:

$$F_y = P\{-\infty \le X \le \infty; Y \le y\}, \tag{3-37}$$

$$F_x = \int_{-\infty}^{y}\int_{-\infty}^{\infty} f(x,y)dxdy.$$

3.3.4.3.2. Marginal density function

From the above, it follows for X and Y respectively:

$$f_X(x) = \int_{-\infty}^{\infty} f(x,y)dy, \tag{3-38}$$

$$f_Y(y) = \int_{-\infty}^{\infty} f(x,y)dx. \tag{3-39}$$

3.3.4.4. *Conditional probability of a pair of random variables*

3.3.4.4.1. Conditional distribution function

The conditional distribution function of X with respect to Y is the function:

$$F(x/y) = P\{X \le x / Y \le y\}, \tag{3-40}$$

and according to the theorem of conditional probability:

$$F(x/y) = \frac{F(x,y)}{F(y)}. \tag{3-41}$$

Thus:

$$F(x,y) = F(y)F(x/y) = F(x)F(y/x).$$

3.3.4.4.2. Conditional density function

From the above, it follows directly for X and Y respectively:

$$f_X(x/y) = \frac{f_{XY}(x,y)}{f_Y(y)}, \tag{3-42}$$

$$f_Y(y/x) = \frac{f_{XY}(x,y)}{f_X(x)}. \tag{3-43}$$

3.3.4.5. *Functions of a pair of random variables*

One considers here not only the relation between two random variables but a function of these variables. This is a common situation when dealing with random phenomena.

3.3.4.5.1. Definition

Let X and Y be two random variables, and let U, V two functions of this pair:

$U = g(X,Y)$, and $V = h(X,Y)$.

Let us denote as G and H the inverse functions:

$X = G(U,V)$, and $Y = H(U,V)$.

We have:

$$P\{x \le X \le x + dx; y \le Y \le y + dy\} = P\{u \le U \le u + du; v \le V \le v + dv\},$$

or, in other words:

$$f_{XY}(x,y)dxdy = f_{UV}(u,v)dudv.$$

Thus:

$$f_{UV}(u,v) = \frac{f_{XY}(x,y)}{J_{\frac{uv}{xy}}}, \tag{3-44}$$

with J, the "Jacobian" of U and V:

$$J = \begin{vmatrix} \frac{dU}{dx} & \frac{dU}{dy} \\ \frac{dV}{dx} & \frac{dV}{dy} \end{vmatrix}, \tag{3-45}$$

i.e. the determinant of the partial derivatives of U and V with respect to x and y.

3.3.4.5.2. Example of application

In order to make this notion more concrete, consider the simple configuration of two equipments "in series" (see Chapter 6). Their characteristics (processing time, or failure rate for instance) are two random variables X and Y. We are interested in the probability density function of the variable "sum" (the total processing time for instance) $U = X + Y$.

Taking here simply $V = X$, we have:

$$J = \begin{vmatrix} \frac{dU}{dx} & \frac{dU}{dy} \\ \frac{dV}{dx} & \frac{dV}{dy} \end{vmatrix} = \begin{vmatrix} 1 & 1 \\ 1 & 0 \end{vmatrix} = 1.$$

Hence:

$$f_{UV}(u,v) = f_{XY}(x,y),$$

a result which is intuitively obvious in such a simple case.

Now consider U, its density is the marginal density of the pair (u, v):

$$f_U(u) = \int f_{UV}(u,v)dv,$$

and then:

$$f_U(u)=\int f_{UV}(u,v)dv=\int f_{XY}(x,y)dx=\int f_X(x)f_Y(y)dx\,.$$

But we have also that $y=u-x$, and thus:

$$f(u)=\int f(x)f(u-x)dx\,.$$

The density of the sum is the *convolution product* of the densities of x and y, and is usually symbolized as

$$f(u)=f(x)\otimes f(y)\,.$$

In the following, we will see that many problems involving several random variables make use of convolution products. In many cases it will be possible to derive directly the solution from the definition. This is further developed in the next section. Then, we will present other resolution methods, based on powerful tools such as Laplace transforms, generating functions or characteristic functions.

Of prime importance is the case of the sum of independent random variables.

3.3.4.6. *Sum of independent random variables*

Imagine a complex system, for which a global performance parameter is studied (e.g., some processing delay). The analysis proceeds often by decomposing the system in subsystems which are sequentially invoked during the processing. Thus, if the process is a sequence of n elementary tasks in the various subsystems, the total delay is the sum of the n elementary delays. This exemplifies the importance of studying the properties of the sum of random variables. We set up here the main results concerning these sums.

3.3.4.6.1. Probability density of the sum of independent random variables

Let X and Y be two independent random variables, and $Z = X + Y$ their sum. We have:

$$F(z)=P\{Z\le z\}=P\{\{X+Y\}\le z\}, \tag{3-46}$$

and thus:

$$F(z) = \int_{-\infty}^{\infty} dy \int_{-\infty}^{z-y} f(x,y)dx.$$

Since X and Y are independent variables, one can write:

$$F(z) = \int_{-\infty}^{\infty} \left[\int_{-\infty}^{z-y} f_X(x)dx \right] f_Y(y)dy,$$

$$F(z) = \int_{-\infty}^{\infty} F_X(z-y)f_Y(y)dy, \tag{3-47}$$

and thus:

$$f(z) = \int_{-\infty}^{\infty} f_X(z-y)f_Y(y)dy. \tag{3-48}$$

Once again, the convolution product appears, as in the previous example.

In the case of discrete variables, we have immediately:

$$P(Z=k) = \sum_{i=0}^{k} p_X(i)p_Y(k-i). \tag{3-49}$$

3.3.4.6.2. An example of application

Let X and Y be two variables, each one obeying a Poisson distribution with parameters λ_1 and λ_2 respectively. Recall the expression for the Poisson law (see section 3.3.3.4 and Chapter 4):

$$p(k) = \frac{\lambda^k}{k!}e^{-\lambda}.$$

Then the density of their sum $Z = X+Y$ is given by:

$$P(Z=k) = \sum_{i=0}^{k} p_X(i)p_Y(k-i) = \sum_{i=0}^{k} \frac{\lambda_1^i}{i!}e^{-\lambda_1} \frac{\lambda_2^{k-i}}{(k-i)!}e^{-\lambda_2},$$

$$P(Z=k)=\frac{e^{-(\lambda_1+\lambda_2)}}{k!}\sum_{i=0}^{k}\frac{k!}{i!(k-i)!}\lambda_1^i\lambda_2^{k-i}=\frac{e^{-(\lambda_1+\lambda_2)}}{k!}\sum_{i=0}^{k}\binom{k}{i}\lambda_1^i\lambda_2^{k-i}.$$

And using the expansion, $(a+b)^k=\sum_{i=0}^{k}\binom{k}{i}a^ib^{k-i}$, we have the remarkable result:

$$P(Z=k)=\frac{e^{-(\lambda_1+\lambda_2)}}{k!}(\lambda_1+\lambda_2)^k. \tag{3-50}$$

The sum of two Poisson independent random variables is itself a Poisson variable. This is a basic property for all traffic and performance studies: for instance we conclude that a system which is offered several independent Poisson flows (calls, messages) will be globally offered a Poisson flow.

3.3.4.7. *Moments of the sum of independent random variables*

3.3.4.7.1. Mean

The mean of the sum of independent variables X, Y... is equal to the sum of their means:

$$E[X+Y+\cdots]=E[X]+E[Y]+\cdots. \tag{3-51}$$

Let us give the proof for two variables:

$$E[X+Y]=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(x+y)f_{XY}(x,y)dxdy.$$

$$E[X+Y]=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}xf_{XY}(x,y)dxdy+\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}yf_{XY}(x,y)dxdy,$$

$$E[X+Y]=\int_{-\infty}^{\infty}xf_X(x)dx+\int_{-\infty}^{\infty}yf_Y(y)dy=E[X]+E[Y].$$

The extension of this result to the sum of any number of variables is obvious. Furthermore notice that it follows from the proof that the result holds whether the variables are independent or not.

3.3.4.7.2. Variance

The variance of the sum of two *independent* random variables is equal to the sum of their variances:

$$V[X+Y]=V[X]+V[Y]. \tag{3-52}$$

Let us give the proof for two variables.

By the definition of the variance, we have:

$$V[X]=\sigma^2=E\left[\{X-E[X]\}^2\right].$$

Hence for the sum of two variables:

$$V[X+Y]=E\left[\{(X+Y)-E[X+Y]\}^2\right].$$

And, as just proven above:

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

Thus:

$$V[X+Y]=E\left[\{(X-E[X])+(Y-E[Y])\}^2\right]=$$
$$E\left[(X-E[X])^2+(Y-E[Y])^2+2(X-E[X])(Y-E[Y])\right],$$

$$V[X+Y]=$$
$$E\left[(X-E[X])^2\right]+E\left[(Y-E[Y])^2\right]+2E\left[(X-E[X])(Y-E[Y])\right].$$

The last term of this equation is called the *covariance* of X and Y, denoted as $Cov[X,Y]$:

$$Cov[X,Y]=E\left[(X-E[X])(Y-E[Y])\right].$$

Thus one has the important following result:

$$V[X+Y]=V[X]+V[Y]+2Cov[X,Y]. \tag{3-53}$$

Notice the generality of the result, as up to now we did not make use of any independence assumptions.

Consider now the case of independent variables. Developing the expression of the covariance and observing that:

$$E\left[XE[Y]\right]=E\left[YE[X]\right]=E[X]E[Y],$$

we have:

$$Cov[X,Y]=E[XY]-E[X]E[Y].$$

More generally:

$$E[XY]=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} xyf_{XY}(x,y)dxdy,$$

and, if the two variables are *independent*:

$$E[XY]=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} xyf_X(x)f_Y(y)dxdy=E[X]E[Y]. \qquad (3\text{-}54)$$

The mean of the product of two *independent* random variables is equal to the product of their means. As for the sum the extension of the result to any number of variables is obvious.

And thus, for independent variables:

$$Cov[X,Y]=E[XY]-E[X]E[Y]=0.$$

And now, for the variance one has the important result:

$$V[X+Y]=E\left[(X-E[X])^2\right]+E\left[(Y-E[Y])^2\right]=V[X]+V[Y].$$

The variance of the sum of two *independent* random variables is equal to the sum of their variances. Once again, the extension of the result to any number of variables is straightforward.

3.3.4.8. *Practical interest*

The previous results have obvious and important applications, and especially here for performance studies. When trying to derive the probability distribution of a phenomenon corresponding to the sum of individual variables, they allow derivation of exact characteristics of the resulting distributions, by summing up individual moments. The practical interest will appear also in statistics, and especially in sampling theory.

3.4. Convolution

We now study in detail the properties of this fundamental operation.

3.4.1. *Definition*

Let $f(t)$ and $g(t)$, be two functions: their convolution product (convolution for brevity), is the function:

$$f(\tau)=\int_0^{\tau} f(t)g(\tau-t)dt \tag{3-55}$$

The interpretation is obvious. Imagine that $f(t)$ and $g(t)$ represent the probability density functions of the processing delay of two elements in sequence, the probability of a total delay equal to τ is the product of the probability that the first delay is t and the second one $\tau-t$, whatever the value for t (theorem of the total probability).

The convolution corresponds to the probability density of the sum of random variables.

Evidently, the concept of convolution product can be used for discrete variables. An important application is the study of common resources, such as a buffer (in any queueing system). Here, the problem is to estimate the probability of having N records, in the case where two servers work to empty the queue (two outgoing links in a router, sharing a common memory). Knowing the probability distributions p_1 and p_2 to have $n_1 = n$ packets waiting for the first server and $n_2 = N-n$ waiting for the second one:

$$p(N)=\sum_{n=0}^{N} p_1(n)p_2(N-n).$$

More generally, the convolution product of several functions is denoted using the following symbol:

$$f(t)\otimes g(t)\otimes h(t)\cdots.$$

3.4.2. *Properties of the convolution operation*

3.4.2.1. *The convolution is commutative*

$$f(t)\otimes g(t)=g(t)\otimes f(t).$$

Indeed, changing the variable:

$\theta=\tau-t$, thus $t=\tau-\theta$, and $dt=-d\theta$ yields:

$$\int_0^\tau f(t)g(\tau-t)dt=-\int_\tau^0 f(\tau-\theta)g(\theta)d\theta=\int_0^\tau f(\tau-\theta)g(\theta)d\theta.$$

Let us now compute the convolution of two functions of constant use in performance studies: the exponential and the normal distributions.

3.4.2.2. *Convolution of exponential distributions*

Given:

$$f(t)=Ae^{-at},\ g(t)=Be^{-bt},$$

one has:

$$f(\tau)=\int_0^\tau Ae^{-a(\tau-t)}Be^{-bt}dt=ABe^{-a\tau}\int_0^\tau e^{(a-b)t}dt,$$

$$f(\tau)=\frac{ABe^{-a\tau}}{a-b}(e^{(a-b)\tau}-1),\ f(\tau)=\frac{AB}{a-b}(e^{-b\tau}-e^{-a\tau}).$$

3.4.2.3. *Convolution of normal distributions*

We will define precisely this important function in the next chapter. We now just take its simplest expression called the *standard normal distribution*:

$$f(x)=\frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}.$$

Then the convolution of two normal distributions is:

$$f(X)=\frac{1}{2\pi}\int_{-\infty}^{\infty}e^{-\frac{1}{2}\left[x^2+(X-x)^2\right]}dx.$$

Developing the expression:

$$u = x^2 + (X - x)^2,$$

yields:

$$u = 2x^2 + X^2 - 2xX = \frac{X^2}{2} + 2x^2 - 2xX + \frac{X^2}{2} = \frac{X^2}{2} + (x\sqrt{2} - \frac{X}{\sqrt{2}})^2.$$

Now, putting:

$$v = \frac{1}{\sqrt{2}}(x\sqrt{2} - \frac{X}{\sqrt{2}}) = x - \frac{X}{2},$$

we obtain:

$$f(X) = \frac{1}{2\pi} e^{-\frac{X^2}{4}} \int_{-\infty}^{\infty} e^{-v^2} dv.$$

But:

$$\int_{-\infty}^{\infty} e^{-v^2} dv = \sqrt{\pi}.$$

We have thus the remarkable result:

$$f(X) = \frac{1}{2\sqrt{\pi}} e^{-\frac{\left(\frac{X}{\sqrt{2}}\right)^2}{2}}. \tag{3-56}$$

The convolution of two normal distributions is itself normal. This is a basis for many approximations, since the normal approximation is often a reasonable assumption, provided the variables are not too far from their mean, and in this case the sum is simply the normal distribution with parameters the sum of individual parameters. This will be developed in the next sections.

Moreover, the *central limit theorem* is another fundamental property when summing a large number of random variables, which states that the limit goes to a normal distribution, independently of the individual distributions. This result is presented in Chapter 5.

3.5. Laplace transform

We now introduce the notion of transforms, which provide an efficient manipulation of random variables. They provide essential tools in various problems, such as the resolution of systems of differential equations (state equations) in reliability or in queueing theory, or the utilization of stochastic relations for the resolution of complex queueing systems (see Chapters 6 and 7). Also they enable one to calculate easily the successive moments of a distribution or the derivation of the distributions of sums on random variables.

Note also that each of these transforms conserves all the information of the original function. That means that there is a one-to-one correspondence between a function and its transform.

As for the other types of transforms presented hereafter (characteristic functions, generating functions), the main appeal of the Laplace transform is that it decomposes the original function into a sum (or integral) of elementary exponential functions easy to manipulate. It allows the derivation of some of the results previously obtained in a simpler way (particularly concerning the sum of variables).

3.5.1. *Definition*

The Laplace transform of a function *f(t)* is the function:

$$F^*(s) = \int_{-\infty}^{\infty} f(t)e^{-st}dt. \tag{3-57}$$

In most applications, the functions are such that:

$f(t) = 0$ if $t < 0$.

Then, keeping in mind that the lower limit actually corresponds to 0, the transform is written as:

$$F^*(s) = \int_{0}^{\infty} f(t)e^{-st}dt.$$

3.5.2. *Properties*

3.5.2.1. *Fundamental property*

A fundamental property of the Laplace transform concerns the convolution, just as introduced above. Its formulation is: *the Laplace transform of the convolution of functions is just the product of each transform.* So for two functions $f(t)$ and $g(t)$ we have:

$$F^*\{f(t)\otimes g(t)\}=F^*(s)G^*(s)\,. \tag{3-58}$$

Indeed:

$$F^*\{f(\tau)\}=\int_{\tau=0}^{\infty}\int_{t=0}^{\tau} f(\tau-t)g(t)dte^{-s\tau}d\tau\,.$$

$$F^*\{f(\tau)\}=\int_{t=0}^{\infty}\int_{\tau=t}^{\infty} f(\tau-t)e^{-s(\tau-t)}d\tau g(t)e^{-st}dt\,.$$

$$F^*\{f(\tau)\}=\int_{t=0}^{\infty} g(t)e^{-st}dt\int_{\tau-t=0}^{\infty} f(\tau-t)e^{-s(\tau-t)}d\tau\,.$$

and thus:

$$F^*\{f(t)\otimes g(t)\}=F^*(s)G^*(s)\,.$$

The generalization of this result is straightforward.

Furthermore, using the results of the previous section, it can be reformulated by stating that *the Laplace transform of the distribution of a sum of variable is the product of the individual transforms.* This is a basic property that the Laplace transform shares with the other transforms (characteristic and generating functions), as it will be seen below.

3.5.2.2. *Differentiation property*

From the definition of transform, we have immediately:

$$\frac{df(t)}{dt}\Leftrightarrow sF^*(s)\,,$$

and:

$$\frac{d^n f(t)}{dt^n} \Leftrightarrow s^n F^*(s). \tag{3-59}$$

(However keeping in mind that we have $f(t)=0$ if $t<0$.)

3.5.2.3. *Integration property*

$$\int_{-\infty}^{t} f(t)dt \Leftrightarrow \frac{F^*(s)}{s},$$

and for the n^{th} order integration:

$$\int_{-\infty}^{t} \cdots \int_{-\infty}^{t} f(t)dt^n \Leftrightarrow \frac{F^*(s)}{s^n} \tag{3-60}$$

(with $f(t)=0$ if $t<0$).

3.5.2.4. *Some common transforms*

Here are several results concerning the transforms of most commonly used distributions. This amounts to building pairs of functions {original function, transform}, so that finding the original from the transform can be most of time done by inspecting the table of pairs.

3.5.2.4.1. Unit step function

By definition:

$$\mu(t) = \begin{cases} 1 & t \ge 0 \\ 0 & t < 0 \end{cases}.$$

This function expresses the fact that a probability distribution exists only for positive epochs. For instance the function denoted as $e^{-at}\mu(t)$ is an exponential function taking values only from $t = 0$.

Using directly the definition (3.57), one gets

$$\mu(t) \Leftrightarrow \frac{1}{s}. \tag{3-61}$$

3.5.2.4.2. Dirac delta function (unit impulse)

By definition:

$$\int_{-\infty}^{\infty} \delta(t)dt = 1 \text{, with } \delta(t) = 0 \text{ if } t \neq 0 .$$

This function provides a way to deal with discontinuities of distributions. For instance, it allows representation of a discrete variable as a continuous function. Another important use is for representing discontinuities at the origin. For instance, some equipment, whose reliability is governed by the exponential distribution during its operational life, might have a probability of being up (or down) at the instant to be put into service. This possibility is represented by the Dirac delta function.

Applying the definition of the transform, we have:

$$\int_{0}^{\infty} \delta(t)e^{-st}dt = 1 .$$

Thus:

$$\delta(t) \Leftrightarrow 1, \tag{3-62}$$

and in a same way:

$$\delta(t-\theta) \Leftrightarrow e^{-s\theta} . \tag{3-63}$$

From the above, we deduce the important property:

$$f(t) \otimes \delta(t-\theta) \Leftrightarrow F^*(s)e^{-s\theta} ,$$

and thus:

$$f(t) \otimes \delta(t-\theta) = f(\tau-\theta) .$$

The convolution of a function with a Dirac delta function shifted on the time axis also shifts the function.

The Dirac function is also denoted:

$$u_0(t) = \delta(t) ,$$

and:

$$u_{-1}(t) = \int_{-\infty}^{t} u_0(x)dx.$$

And we have:

$$u_{-1}(t) = \mu(t).$$

The unit step function is the integral of the Dirac delta function.

More generally, for the n^{th} integrals and derivatives of two functions, we have the following relationships:

$$u_{-n}(t) = \frac{t^{n-1}}{(n-1)!} \Leftrightarrow \frac{1}{s^n}, \tag{3-64}$$

$$u_n(t) = \frac{d}{dt}u_{n-1}(t) \Leftrightarrow s^n. \tag{3-65}$$

3.5.2.4.3. Exponential function

$$Ae^{-at}\mu(t) \Leftrightarrow \frac{A}{s+a}. \tag{3-66}$$

Notice that when taking the limiting case $a = 0$, one finds again the result obtained for the unit step function

3.5.2.4.4. Application

Let us apply this result to one of our preceding examples, the convolution of two exponential functions:

$$f(t) = Ae^{-at}, \ g(t) = Be^{-bt}.$$

From the above we have:

$$F^*(s) = \frac{A}{s+a}, \ G^*(s) = \frac{B}{s+b}.$$

$$F^*(s)G^*(s) = \frac{AB}{(s+a)(s+b)} = \frac{AB}{a-b}\left(\frac{1}{s+b} - \frac{1}{s+a}\right).$$

Now, by inspection (using the results above, see exponential transform), we come back to the original (untransformed) function, and we obtain:

$$f(\tau) = \frac{AB}{a-b}\left(e^{-b\tau} - e^{-a\tau}\right).$$

The result is of course the same as in the previous section.

Let us now consider the remarkable case of the sum of n exponential variables identically distributed, which happens to be a basic configuration in performance studies, as it will be illustrated throughout this book.

$$F^*(s) = \left(\frac{A}{s+a}\right)^n,$$

whose original is:

$$f(t) = \frac{A^n t^{n-1}}{(n-1)!} e^{-at}. \tag{3-67}$$

Clearly, for $n = 1$ the solution is:

$$f(t) = Ae^{-at}$$

Let us verify that for $n = 2$ the solution is:

$$f(t) = A^2 t e^{-at}.$$

We have:

$$F^*(s) = \int_0^\infty A^2 t e^{-at} e^{-st} dt = A^2 \int_0^\infty t e^{-(s+a)t} dt.$$

And after a simple integration by parts ($\int_0^\infty u dv$, with $u = t$ and $dv = e^{-(s+a)t} dt$),

we obtain $F^*(s) = \dfrac{A^2}{(s+a)^2}$.

Proceeding step by step in a same way, one can verify that the general solution is $f(t) = \dfrac{A^n t^{n-1}}{(n-1)!} e^{-at}$.

We will see in Chapter 4 that this is a very useful function, called the Erlang-*n* distribution.

3.6. Characteristic function, generating function, z transform

These functions, just like Laplace transforms, provide basic tools for statistics and queueing theory. They have close mutual relationships, each one having however a preferred domain of usage, as we will see in this section.

3.6.1. *Characteristic function*

The characteristic function provides another expression for the Laplace transform. This is an essential tool of queueing theory for complex systems, as will be presented, and especially in Chapter 7.

3.6.1.1. *Definition*

The characteristic function of a random variable *X*, whose distribution function is *F*(*x*), is the function:

$$\phi(u) = \int_{-\infty}^{\infty} e^{iux} dF(x). \tag{3-68}$$

One can see that it is the Fourier transform of *F*(*x*).

Furthermore we have:

$$\phi(is) = \int_{-\infty}^{\infty} e^{-sx} dF(x) = F^*(s).$$

This gives an immediate way to obtain the characteristic function from the Laplace transform, and vice versa. This function has been introduced by Laplace, for absolutely continuous functions. Its use has been mainly developed through the work of Paul Levy. The function is written as:

$$\phi(z)=\int_{-\infty}^{\infty} e^{zx}dF(x),\ z \text{ being a complex variable.} \tag{3-69}$$

In the case of a discrete variable we have:

$$\phi(z)=\sum_k p_k e^{zk}$$

which is also written, for brevity:

$$\phi(z)=E\left[e^{zX}\right]. \tag{3-70}$$

Once again, one can remark that the transform defines entirely the original probability distribution. We present below its main properties and several major results related with the use of the residue theorem, leading to simple applications.

We will illustrate in Chapter 7 the help it can bring for solving complex problems in queueing theory, for instance by means of Pollaczek's method and for obtaining limit distributions.

3.6.1.2. *Properties*

The characteristic function always exists on the imaginary axis, and is such that $\phi(0)=1$, $|\phi(u)|\le 1$, and such that $\phi(u)$ is continuous, defined and positive.

3.6.1.3. *Inversion formula*

At points where *F(x)* is continuous one has:

$$F(x)=\frac{1}{2\pi i}\int_{-i\infty+\delta}^{i\infty+\delta} e^{zx}\phi(-z)\frac{dz}{z}, \tag{3-71}$$

provided that this integral converges. One can recognize a Cauchy integral in the complex plane, where the contour of integration is a vertical line parallel to the imaginary axis at distance $x=\delta(\delta>0)$, and traversed from below to above. This result is obtained by using the Heaviside function, as explained now.

3.6.1.3.1. The notion of event indicator and the Heaviside function

The Heaviside function is defined by:

$$H(x)=1,\text{ for } x>0\,;\; H(x)=\frac{1}{2},\text{ for } x=0\,;\; H(x)=0,\text{ for } x<0,$$

and for $x \neq 0$ it is represented by the Dirichlet integral:

$$H(x)=\frac{1}{2\pi i}\int_{-i\infty+\delta}^{i\infty+\delta} e^{zx}\frac{dz}{z} \tag{3-72}$$

which is as before a Cauchy integral, taken in the complex plane, and where the contour of integration (closed to ∞) is a vertical line parallel to the imaginary axis, at distance $x = \delta$, and traversed from $-\infty$ to $+\infty$. In the following, for convenience, we will denote this integral $\int_{Cs}$, its contour being located just on the right and close to the imaginary axis.

This function allows definition of the probability of an event:

Indeed, given the event $x > 0$, let us call the *indicator* of the event a function equal to 1 for $x > 0$, and equal to 0 otherwise, i.e. the function $H(x)$. Then we have the fundamental following relationship: the probability of the event is the expectation of its indicator, $F(x)=E\{H(x)\}$.

Indicator of an event and distribution function

Let X be a random variable such as already defined. One has:

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

According to the definition its characteristic function is:

$$\phi(z)=\int_{-\infty}^{\infty} e^{zx}dF(x)=E(e^{zX}).$$

Consider the event $(x-X)$. The indicator of the event $(x-X)>0$ is $H(x-X)$, and thus:

$$F(x) = E\{H(x-X)\}, \tag{3-73}$$

or:

$$F(x) = E\left\{\frac{1}{2\pi i}\int_{Cz} e^{z(x-X)}\frac{dz}{z}\right\} = \frac{1}{2\pi i}\int_{Cz} e^{zx}E(e^{-zX})\frac{dz}{z}.$$

And finally, provided $F(x)$ is continuous at point x:

$$F(x) = \frac{1}{2\pi i}\int_{Cz} e^{zx}\phi(-z)\frac{dz}{z}$$

This is just the inversion formula presented above.

3.6.1.3.2. Use of the inverse function

Now we present how to compute the integral in the inversion formula. First remember that if $f(z)$ is an holomorphic function of the complex variable z, (see Appendix 1) in the domain D bounded by a closed contour C, and if $f(z)$ is continuous in $D+C$, then for any point z_0 of D one has:

$$\frac{1}{2\pi i}\int_{C+} \frac{f(z)}{z-z_0})dz = f(z_0),$$

the path being traversed in the positive direction (counterclockwise).

Then one may verify that $f(z)$ can be expanded according to the powers of $(z-z_0)$ inside the circle centred on z_0, and that the series converges so long as z remains inside C.

Consequently one can apply the residue theorem to obtain the solution, which is: $\frac{1}{2\pi i}\int_{C+} f(z)dz = \sum R_i$ where R_i is the residue at the singularity z_i (see below).

The usefulness of the characteristic function is made clear as in most cases it is simpler to express functions of random variables through their characteristic functions (e.g., in the case of sums of variables), and then to come back to the distribution. However, integrating the inversion formula may happen to be uneasy, or even impossible, and one has sometimes to rely on approximations.

Most often, the original distribution function is directly obtained by a simple inspection, as it has been illustrated for the Laplace transform.

Nevertheless, the inversion formula is of great help in several cases, especially to derive asymptotic results (as illustrated just below), and also to derive fundamental results by applying the Pollaczek approach (see Chapter 7).

The use of the residue theorem is essential in these calculations, and its main aspects are presented here.

3.6.1.4. *Asymptotic law*

One looks for an asymptotic expression of the distribution $F(x)$ for large values of x. This kind of expression is useful as it provides approximate results for many queueing problems. We begin with the inversion formula on which we apply the residue theorem.

The residue theorem

Let us recall the residue theorem: let z_1 be a pole or an isolated singular point (see Appendix 1) for the function $f(z)$, (*$f(z)$ is holomorphic in a circle whose centre is z_1, except in z_1*), the residue of $f(z)$ at this point is the coefficient R_1 of $\dfrac{1}{z-z_1}$ in the Laurent expansion around z_1. Let C be a simple curve, closed, and traversed in the positive direction. If within C, $f(z)$ does not have any other singularity than z_1, then:

$$\frac{1}{2\pi i}\int_{C+} f(z)dz = R_1$$

Indeed, one may replace C by any other similar curve, for instance a circle of centre z_1, and one will then verify that this expression is the coefficient of $1/(z-z_1)$ in the Laurent expansion of $f(z)$ (see Appendix 1).

And, in a more general way (for several singularities), the residue theorem is:

$$\frac{1}{2\pi i}\int_{C+} f(z)dz = \sum R_i \tag{3-74}$$

Computing residues

Generally, in order to obtain the residue of $f(z)$ at the pole z_i, we expand $f(z)$ in its Laurent series around z_i., and the residue is the coefficient of the term $1/(z-z_i)$. The Taylor series may be used too (see Appendix 1): z_i being a pole of order n, the residue in that point is the coefficient of $(z-z_i)^{n-1}$ in the Taylor development of $\psi(z)=\left[(z-z_i)^n f(z)\right]$, i.e.:

$$R_n = \frac{1}{(n-1)!}\frac{d^{n-1}}{dz^{n-1}}\left[(z-z_i)^n f(z)\right]$$

Especially for a simple pole (pole of order 1), if $f(z)=\dfrac{P(z)}{Q(z)}$ with $P(z_1)\neq 0$ and $Q(z_1)=0$,

$$R_1 = \lim_{z\to z_1}\frac{(z-z_1)P(z)}{Q(z)} = \lim_{z\to z_1}\frac{P(z)}{\left(Q(z)-Q(z_1)\right)/(z-z_1)}, \text{ i.e. } R_1=\frac{P(z_1)}{Q'(z_1)}.$$

Example

Consider the function $f(z)=\dfrac{1}{(z+1)(z-1)^2}$. It has a simple pole, $z_1=-1$, and a multiple one of order 2, $z_2=1$. In order to expand around z_2, we take $z=1+h$, (h infinitively small). Then the Laurent expansion is:

$$f(z)=\frac{1}{h^2(2+h)}=\frac{1}{2h^2(1+\frac{h}{2})}=$$

$$\frac{1}{2h^2}(1-\frac{h}{2}+\frac{h^2}{4}-\frac{h^3}{8}\cdots)=\frac{1}{2h^2}-\frac{1}{4h}+\frac{1}{8}-\frac{1}{6}h\cdots$$

The residue, coefficient of $1/h$ (i.e. of $1/(z-1)$), is thus $R_2=-\dfrac{1}{4}$. Similarly, at the simple pole we take $z=-1+h$ and we get $R_1=\dfrac{1}{4}$ (the exercise is left to the reader).

We may also apply directly the general formula. For instance:

$$R_1 = \left[(z+1)f(z)\right]_{z=z_1=-1} = \left[\frac{1}{(z-1)^2}\right]_{z=z_1=-1} = \frac{1}{4},$$

or also: $R_1 = \left[\frac{1/(z-1)^2}{(z+1)'}\right]_{z=z_1=-1} = \frac{1}{4}$.

Similarly:

$$R_2 = \frac{d}{dz}\left[(z-1)^2 f(z)\right]_{z=z_2} \text{ thus } R_2 = \left[\frac{-1}{(z+1)^2}\right]_{z=z_2=1} = -\frac{1}{4}.$$

We obtain of course the same results.

Asymptotic formula

Now we are in a position to establish the asymptotic result we are looking for. Let us assume that the first singular point of our integral is a simple pole z_1 (necessarily real, see P. Levy theorem). Applying the residue theorem to the inversion formula at the poles $z = 0$ and z_1 yields:

$$F(x) = 1 + \frac{R_1}{z_1} e^{z_1 x} + \frac{1}{2\pi i} \int_{-i\infty+\delta_1}^{i\infty+\delta_1} e^{zx} \phi(-z) \frac{dz}{z},$$

with $\delta_1 < z_1$, and R_1 residue of $\Phi(-z)$ at z_1. The last integral decays to zero when x grows to infinity, and then for very large value of x we have:

$$F(x) \approx 1 - \frac{-R_1}{z_1} e^{z_1 x}, \text{ and so } P(>x) \approx \frac{-R_1}{z_1} e^{z_1 x}, \tag{3-75}$$

which may also be written, taking this time $\Phi(z)$ and its residue at pole z_1:

$$P(>x) \approx \frac{R_1}{(-z_1)} e^{-z_1 x}. \tag{3-76}$$

We will develop several simple applications of this fundamental result in Chapter 7. The solution is clearly exact when the singular point is a unique pole.

3.6.1.4.1. Moments

Expanding in power series e^{zx} in the definition, one has:

$$e^{zx} = 1 + x\frac{z}{1!} + x^2\frac{z^2}{2!} + \cdots + x^n\frac{z^n}{n!} + \cdots,$$

and:

$$\phi(z) = 1 + m_1\frac{z}{1!} + m_2\frac{z^2}{2!} + \cdots + m_n\frac{z^n}{n!} + \cdots.$$

Thus taking the n^{th} order derivative:

$$\phi^n(0) = m_n, \tag{3-77}$$

m_n is the n^{th} order moment about the origin, such as defined before. So, the moments are easily derived from the characteristic function.

3.6.1.4.2. Example of application

Let X be a random variable and $f(x) = \lambda e^{-\lambda x}$, with $f(x) = 0$ if $x < 0$. Its characteristic function is:

$$\phi(s) = \int_0^\infty e^{zx}\lambda e^{-\lambda x}dx = \lambda\int_0^\infty e^{(z-\lambda)x}dx = \frac{\lambda}{\lambda - z},$$

and:

$$\phi'(z) = \frac{\lambda}{(\lambda - z)^2},\ \phi''(z) = \frac{2\lambda}{(\lambda - z)^3}.$$

Thus:

$$E[X] = m_1 = \phi'(0) = \frac{1}{\lambda},$$

$$E\left[X^2\right] = m_2 = \phi''(0) = \frac{2}{\lambda^2},$$

and:

$$\sigma^2 = \overline{X^2} - \overline{X}^2 = \frac{1}{\lambda^2}.$$

3.6.1.4.3. Sum and difference of independent random variables

Consider the variable $Y = X_1 + X_2$, with X_1 and X_2 two independent random variables. We denote respectively as $\phi(z)$, $\phi_1(z)$, $\phi_2(z)$, their characteristic functions. According to the definition:

$$\phi(z) = E\left[e^{z(X_1+X_2)}\right],$$

thus:

$$\phi(z) = E\left[e^{zX_1}e^{zX_2}\right].$$

Using the properties of the means of independent variables:

$$\phi(z) = E\left[e^{zX_1}\right]E\left[e^{zX_2}\right].$$

We find again the basic relationship, already demonstrated above for Laplace transforms:

$$\phi(z) = \phi_1(z)\phi_2(z) \tag{3-78}$$

Similarly, for the difference $Y = X_1 - X_2$, one has:

$$\phi(z) = \phi_1(z)\phi_2(-z) \tag{3-79}$$

These results may be easily generalized to any number of independent variables.

3.6.1.5. *Some usual transforms*

Just as for Laplace transforms, several basic transforms are of constant help in applications. The correspondence with Laplace transforms is immediate, given the relation mentioned in the introduction: $\phi(is) = F^*(s)$. Chapter 4 contains a more comprehensive set of the characteristic functions of usual distributions (Poisson, binomial, etc.).

3.6.1.5.1. Sure function (or almost sure)

$$F(x) = \begin{cases} 0 & x < a \\ 1 & x \geq a \end{cases}.$$

A constant service time provides a typical example.

$$\phi(z) = e^{az}. \tag{3-80}$$

3.6.1.5.2. Exponential function

This function is of constant use in queueing problems to describe inter-arrival delays or service durations:

$$f(x) = Ae^{-ax}, \text{ for } x \geq 0,$$

$$F(x) = \begin{cases} 0 & x < 0 \\ 1 - Ae^{-ax} & x \geq 0 \end{cases}.$$

$$\phi(z) = \frac{A}{a - z}. \tag{3-81}$$

3.6.1.5.3. Geometric distribution

$$p_n = pq^n \ (n = 0,1,2..), \text{ with } q = 1 - p,$$

$$F(n) = 1 - q^{n+1}.$$

$$\phi(z) = \frac{p}{1 - qe^z}. \tag{3-82}$$

3.6.2. *Generating functions*

3.6.2.1. *Definition*

For discrete random variables, the characteristic function is written as:

$$\phi(z) = \sum_k p_k e^{zx_k}.$$

Assume now that X takes only positive integer values. It may then be convenient to introduce the generating function of the p_k:

$$F(z) = \sum_k p_k z^k, \tag{3-83}$$

where z is a complex variable. It is also referred to as the z *transform*. It is defined for $|z| < 1$, since $F(1) = \sum_k p_k = 1$.

We have also:

$$\phi(z) = F(e^z). \tag{3-84}$$

The characteristic function derives from the generating function through a simple change of variable.

3.6.2.2. *Moments*

From the definition, one may directly derive the important relationships concerning moments. Indeed we have:

$$F'(z) = \sum_k k p_k z^{k-1},$$

and so:

$$F'(1) = \sum_k k p_k .$$

Thus:

$$E[X] = \overline{X} = m_1 = F'(1). \tag{3-85}$$

Similarly:

$$F''(z) = \sum_k k(k-1) p_k z^{k-2},$$

then:

$$F''(1) = \sum_k k(k-1) p_k = \overline{X(X-1)} = \overline{X^2} - \overline{X},$$

and finally:

$$\sigma^2 = \overline{X^2} - \left(\overline{X}\right)^2 = F''(1) + F'(1) - \left(F'(1)\right)^2. \tag{3-86}$$

3.6.2.3. *Some usual transforms*

Here are the simplest and most common pairs of discrete distributions and their generating functions.

3.6.2.3.1. (Discrete) unit step function

$\mu_k = 1$, for k = 0,1,2...

And then:

$$F(z) = \sum_k 1 \times z^k = \frac{1}{1-z}.$$

Thus:

$$\mu_k \Leftrightarrow \frac{1}{1-z}. \tag{3-87}$$

3.6.2.3.2. Geometric function

$f_k = Aa^k$, for k = 0, 1, 2, ...

$$F(z) = \sum_k Aa^k z^k = \frac{A}{1-az}.$$

$$Aa^k \Leftrightarrow \frac{A}{1-az}. \tag{3-88}$$

(The reader will easily link this result with formula (3-82).)

3.6.2.4. *Convolution*

At last, just as for the Laplace transform, it can be shown that:

$$f_k \otimes g_k = F(z)G(z). \tag{3-89}$$

The transform of the convolution of functions is equal to the product of their transforms.

Example of application: a multirate network

We illustrate the use of generating functions by applying them to a fundamental issue in multimedia networks, namely mixing services of different bit rates.

Assume the number of simultaneous connections of a given service (of a given bit rate) obeys a Poisson distribution. The problem is to estimate the total bitrate generated by all services. The Poisson law is:

$$p(k)=\frac{\lambda^k}{k!}e^{-\lambda}.$$

And its generating function is:

$$F(z)=\sum_k \frac{\lambda^k}{k!}e^{-\lambda}z^k=e^{-\lambda}\sum_k \frac{(\lambda z)^k}{k!}, \text{ i.e. } F(z)=e^{\lambda(z-1)}$$

If each connection requires a bitrate equal to d, it amounts to say that it consumes d units of resources of the transmission link, e.g. d megabits/s of the total capacity D (assumed here infinitely large).

When k sessions are simultaneously in progress the volume of resources they require is kd. Thus,

$$p(kd)=\frac{\lambda^k}{k!}e^{-\lambda},$$

$$F(z)=\sum_k \frac{\lambda^k}{k!}e^{-\lambda}z^{kd}=e^{-\lambda}\sum_k \frac{(\lambda z^d)^k}{k!},$$

$$F(z)=e^{\lambda(z^d-1)}.$$

Now,

$$\lambda(z^d-1)=\lambda((1+z-1)^d-1)=\lambda(1+d(z-1)+\frac{d(d-1)}{2!}(z-1)^2+\cdots-1) \text{ and}$$

so:

$$F(z)=e^{\lambda(d(z-1)+\frac{d(d-1)}{2!}(z-1)^2+\cdots)}. \tag{3-90}$$

This corresponds to a single service. Assume now that several services with bitrates d_1, d_2, ... are offered, the sources being independent and Poissonnian. It becomes immediately:

$$F(z) = F_1(z)F_2(z) + \cdots = \prod e^{\lambda_i (z^{d_i} - 1)},$$

and $F(z) = e^A$,

with:

$$A = (\lambda_1 d_1 + \lambda_2 d_2 + \cdots)(z-1) + \frac{(\lambda_1 d_1 (d_1 - 1) + \lambda_2 d_2 (d_2 - 1) + \cdots)(z-1)^2}{2!} + \cdots.$$

This gives the characteristic of the distribution resulting from the superposition. This is no longer a Poisson distribution, as one recognizes by comparing it with the previous expressions. However, by introducing the parameter

$$\delta = \frac{\lambda_1 d_1^{\,2} + \lambda_2 d_2^{\,2}}{\lambda_1 d_1 + \lambda_2 d_2},$$

one gets:

$$A = \frac{(\lambda_1 d_1 + \lambda_2 d_2 + \cdots)}{\delta} \left[\delta(z-1) + \frac{\delta(\delta-1)}{2!}(z-1)^2 + \psi(z-1) \right],$$

and thus:

$$F(z) = e^{\frac{(\lambda_1 d_1 + \lambda_2 d_2 + \cdots)}{\delta} \left[\delta(z-1) + \frac{\delta(\delta-1)}{2!}(z-1)^2 + \psi(z-1) \right]}.$$

This expression can be compared with equation (3-90) for a single service, if the terms in $z-1$ of power higher than 2 (i.e. $\psi(z-1)$) are neglected. The system with 2 services with different bitrates is thus approximately equivalent to a single Poisson source with parameter $\lambda = \frac{(\lambda_1 d_1 + \lambda_2 d_2 + \cdots)}{\delta}$ and bitrate $\delta = \frac{\lambda_1 d_1^{\,2} + \lambda_2 d_2^{\,2}}{\lambda_1 d_1 + \lambda_2 d_2}$. This is easily generalized to an arbitrary number of services.

The reader should remark that, in the case where the bitrates are the same $d = d_1 = d_2 \ldots$, one gets again a Poisson law with parameter $\lambda = \lambda_1 + \lambda_2 \ldots$

An intuitive interpretation

In order to have a deeper understanding of this result, the same problem can be addressed by means of the moments. For a single service, the definition gives directly the value of the first moments:

$$\overline{X} = d\sum kp_k = \lambda d\ ,\ \overline{X^2} = d^2\sum k^2 p_k = d^2(\lambda^2 + \lambda),$$

and thus:

$$m = \lambda d \text{ and } \sigma^2 = \overline{X^2} - \overline{X}^2 = \lambda d^2$$

As the variables are independent, the distribution of the superposition of two services has its first 2 moments given by:

$$m = \lambda_1 d_1 + \lambda_2 d_2 \text{ and } \sigma^2 = \lambda d_1^{\ 2} + \lambda d_2^{\ 2}.$$

In order to approximate this distribution by a unique Poisson law with parameters λ and δ, one must have:

$$\lambda\delta = \lambda_1 d_1 + \lambda_2 d_2 \text{ and } \lambda\delta^2 = \lambda d_1^{\ 2} + \lambda d_2^{\ 2},$$

and thus:

$$\delta = \frac{\lambda_1 d_1^{\ 2} + \lambda_2 d_2^{\ 2}}{\lambda_1 d_1 + \lambda_2 d_2} \text{ and } \lambda = \frac{(\lambda_1 d_1 + \lambda_2 d_2)}{\delta}.$$

This is again the same expression. Approximating the exact result given by the generating function amounts to approximating to a Poisson distribution, by adjusting the two first moments. This allows one to represent the complex behaviour of a multirate system by a much simpler Poisson process with a unique bitrate δ. In the sequel, several applications to the traffic engineering of IP networks will illustrate the interest of such results.

4

Probability Laws

Random phenomena are clearly of various kinds, and their probabilistic description makes use of numerous probability laws. In this chapter, we present the most important, and most commonly used of these laws, giving their main characteristics (distribution function, transforms, moments) and commenting the "circumstances" under which they are likely to appear.

The specialist is to apply these laws in various situations. For instance, an experimental distribution of a character has been obtained in a measurement campaign, and the goal is to build a theoretical model able to represent the result. This amounts to *adjusting* a mathematical model using the measurements. Another case is in the choice *a priori* of the model of a process in an ongoing project (typically for a simulation experiment). Here, the choice of law is made by invoking intuitive or mathematical arguments sustaining a specific law. Needless to say, such a choice relies mainly on the specialist's experience. Lastly, the need occurs to *analyze* the behaviour of a phenomenon of given probability distribution (e.g. obtained through one of the previous steps). The analysis is based upon the properties of the law (moments, transforms) and makes it possible to draw various figures of interest (e.g. loss probabilities, average delays, etc.).

Depending on circumstances, observations are represented using discrete laws (number of events arriving in an observation window, number of failures, etc.) or continuous ones (e.g. service duration). Denoting the random variable as X, the law is thus either discrete, defined by a distribution function:

$$p_i = P\{X = i\}, \tag{4-1}$$

or continuous, defined by the distribution or the *probability density function* (PDF):

$$P(x) = P\{X \le x\}\,,\ f(x) = \frac{d}{dx}P(x)\,. \tag{4-2}$$

4.1. The (discrete) uniform distribution

The uniform law is the simplest one can imagine. Let X be a discrete random variable (the outcomes are represented as 0, 1, 2, etc., without loss of generality), such that each outcome has the same probability of occurrence (this is a frequent modelling assumption). Tossing an unloaded die is the most obvious illustration, each face having on the long term the same frequency, the probabilistic model simply assigns a probability 1/6 to each outcome.

A discrete random variable taking values between a and b has $b-a+1$ possible values. It obeys the uniform distribution, denoted as $U(a,b)$, if its probability law is:

$$P(X = k) = \frac{1}{b-a+1},\ a \le k \le b\,. \tag{4-3}$$

Its mean value is easily derived:

$$m = \sum kP(X=k) = \sum_{k=a}^{k=b} \frac{k}{b-a+1} = \frac{1}{b-a+1}\left[\sum_1^b k - \sum_1^{a-1} k\right] =$$
$$\frac{1}{b-a+1}\left[\frac{b(b+1)-(a-1)a}{2}\right] = \frac{a+b}{2}\,.$$

$$\text{Mean: } m = \frac{a+b}{2} \tag{4-4}$$

A calculation of the same kind, making use of the formula giving the sum of squares ($\sum_1^n k^2 = \frac{n(n+1)(2n+1)}{6}$) allows one to derive the variance, through tedious manipulations.

$$\text{Variance: } Var = \frac{(b-a)(b-a+2)}{12} \tag{4-5}$$

The continuous version of this law will be addressed later on in a later section.

4.2. The binomial law

The binomial law is encountered in numerous applications of probability theory. Let us consider the operation of quality control of a production line. Assume that each item has probability p to be defective. The circumstances making the item faulty or not are such that the status of an item does not depend on the status of the previously produced ones. Any element extracted at random is faulty with probability p. Now, assume that a batch of N items is controlled. How many of them are faulty? More precisely, as the experiment is of random nature, what is the probability of observing 0, 1, 2... faulty items among the N ones?

This is one of the classical problems addressed by combinatorial analysis. Each trial corresponding to the event "k faulty elements" consists in extracting, in any order, k faulty items and $N - k$ good ones. Several drawings lead to this result. For instance, denoting as "0" faulty elements and as "1" good ones, the sequences 110011, 111100, 001111, etc. yield the same event "2 faulty among 6". Each of these sequences has probability $p^k(1-p)^{N-k}$ (k and only k defaults have been observed). Moreover, as the order in which faulty elements are observed is irrelevant, the probability of the event is simply the sum of the probability of individual sequences (which are clearly exclusive). Classical counting consists of putting k "balls" inside N "urns", so that each urn contains at most one ball. There are N ways to choose the first urn where a ball is put, $N - 1$ ways to choose the second one, etc. so that there are $N(N-1)(N-2)...(N-k+1) = N!/(N-k)!$ possible ways. In this operation, however, the same sequence has been counted $k!$ times (as the k balls are not distinguishable, this is the number of permutations of k objects).

Finally, there were $\dfrac{N!}{k!(N-k)!} = \dbinom{N}{k}$ ways of observing the event, and the probability of the event is:

$$P(k) = P(k \text{ among } N) = \binom{N}{k} p^k (1-p)^{N-k}. \tag{4-6}$$

This is the so-called binomial distribution, since $P(k)$ is the term of rank k in the development of $[p+(1-p)]^N$. Transforms are the easiest way to obtain its moments, and we derive them using its characteristic function (see Chapter 3).

The characteristic function is obtained by the following argument: for a single drawing, the characteristic function is $\phi(z) = E(e^{zx}) = q + pe^z$, since the result is $x = 1$ with probability p, and 0 otherwise. Thus, for N independent drawings, since the characteristic function of the sum of independent variables is the product of the individual functions:

$$\phi(z) = (q + pe^z)^N, \tag{4-7}$$

from which the final result is:

$$m = \sum kP(k) = \phi'(z)_{(z=0)} = N(1 - p + pe^z)^{N-1} pe^z{}_{(z=0)} = Np. \tag{4-8}$$

Variance: $Var = \sum k^2 P(k) - m^2 = \varphi''(0) - (\phi'(0))^2 = Np(1-p)$. (4-9)

Figure 4.1 displays the typical shape for the probability density function of the binomial distribution.

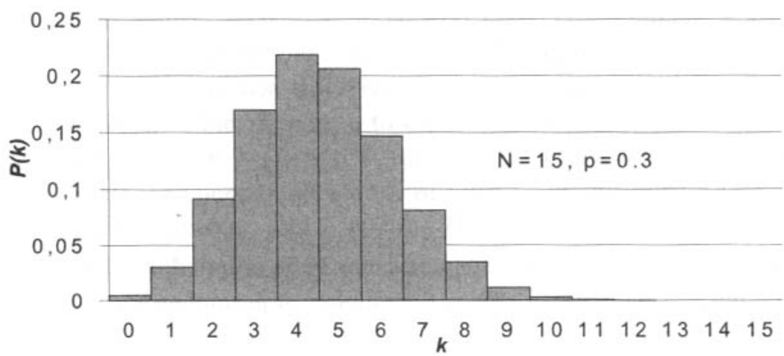


Figure 4.1. *Probability density function of the binomial distribution*

The binomial distribution is of repeated use each time a trial involves superposing independent elementary trials. In the field of traffic, the *Engset* problem introduces it as a limiting distribution (see Chapter 7) of the number of busy lines in subscriber concentrators.

The binomial distribution enjoys the following property, concerning the sum of several variables:

THEOREM.– If two discrete random variables X and Y have binomial distributions with parameters respectively (N, p) and (M, p), the variable $X + Y$ has a binomial distribution with parameters $(N+M, p)$.

The proof is straightforward, by noting that the characteristic function of the sum is the product of the individual functions.

4.3. The multinomial distribution

This is the generalisation of the binomial law. Assume that m types can be distinguished in the population (m kinds of faults, to take the previous example), so that the population has a proportion p_k of elements of type k (with naturally $\sum p_k = 1$). The question is: what is the probability of observing, when extracting N items, n_1 of the type 1, etc., n_m of type m, with $n_1 + n_2 + ... + n_m = N$. The result is:

$$P(n_1, n_2, ..., n_m) = \frac{N!}{n_1! n_2! ... n_m!} p^{n_1} . p^{n_2} ... p^{n_m} . \tag{4-10}$$

This result has numerous applications. Imagine for instance observing a network element (e.g. a concentrator) to which various sources are connected. What is the distribution of the traffic the sources generate? Individual traffic intensity, expressed in erlangs, is distributed between 0 and 1. One usually defines categories, according to the type of subscriber (professional, residential, in urban area, etc.), corresponding to traffic intensities. With two categories, the binomial distribution gives the answer. For several categories (typically: less than 0.03, between 0.03 and 0.05, ..., higher than 0.12), the distribution of the number of customers among the categories is given by the multinomial law. Finally, knowing the distribution of the different customers among the categories allows dimensioning the subscriber concentrator, using the multinomial distribution. More generally, this result holds whenever a population, composed with different sub-populations, is observed.

4.4. The geometric distribution

A discrete variable obeys a geometric distribution if its distribution is given by:

$$p_k = (1 - p)p^k, \; 0 < p < 1, \; k = 0, 1, ... \tag{4-11}$$

Consider for instance a data layer protocol which detects and corrects transmission errors (such as the celebrated X.25). Assume that each packet has a probability p of being erroneously received. Thus a retransmission is needed with probability p. An error is corrected if the following retransmission is successful: probability $1-p$. In the general case, the packet may be successively erroneous during the k first attempts, the $k+1^{\text{th}}$ being correct. This event has probability:

$$P(k \text{ retransmissions}) = (1-p)p^k$$

The moments of the distribution can be estimated directly:

$$m = \sum kp_k = \sum kp^k(1-p) = p(1-p)\sum kp^{k-1} = p(1-p)\frac{d}{dp}\left(\sum p^k\right) =$$

$$p(1-p)\frac{d}{dp}\left(\frac{1}{1-p}\right) = \frac{p}{1-p}.$$

The same kind of calculation gives the variance. Actually, this offers another opportunity to stress the efficiency of the transforms approach. We illustrate here the use of generating functions.

For the geometric law, the generating function is given by:

$$B(z) = \sum z^k p_k = \sum (zp)^k(1-p) = \frac{1-p}{1-pz}, \tag{4-12}$$

from which the results come immediately:

Mean value: $m = \sum kp_k = B'(z)_{z=1} = \dfrac{p}{1-p}$. (4-13)

Variance: $Var = \sum (k-m)^2 p_k = \dfrac{p}{(1-p)^2}$. (4-14)

As a comparison, here is the same calculation using the characteristic function:

$$\phi(z) = \sum e^{zk} p_k = \frac{1-p}{1-pe^z}, \tag{4-15}$$

from which the derivatives taken at $z = 0$ (see Chapter 3) give:

$m = \phi'(z)_{(z=0)}$ and $Var = \varphi''(0) - (\phi'(0))^2$.

(The reader is encouraged to proceed through the calculation.)

The law is sometimes presented in a slightly different form. In the above example, one could have been concerned by the distribution of the *total* number of transmissions. This time:

$$P(k \text{ transmissions}) = (1-p)p^{k-1}$$

4.5. The hypergeometric distribution

Consider once again the configuration leading to the binomial distribution: random extraction from a population having a given proportion of a certain character. This time, assume the population is of finite size H, of which a proportion p is distinguished (faulty elements, in the previous example): there are $M = Hp$ such elements. The experiment consists here in drawing N items without replacement (i.e. the items already drawn are not replaced in the population, which is also called an "exhaustive" drawing). At each trial, the size of the remaining population decreases and the proportion evolves. The binomial distribution holds only if the proportion remains constant. The hypergeometric distribution is to be used in this case.

The probability of drawing k elements of type T while extracting N of them in a population of size H containing M elements of type T (proportion $p = M/H$) is:

$$P(k) = \frac{\binom{M}{k}\binom{H-M}{N-k}}{\binom{H}{N}} \tag{4-16}$$

for $\max(0, N+M-H) \le k \le \min(M, N)$

This is the ratio of the number of ways to realise the event (choosing k among M, and the N–k others of the other type) to the total number of possible drawings. The moments are:

$$\text{Mean value: } N\frac{M}{H} = Np \tag{4-17}$$

Variance: $\sigma^2 = N\dfrac{M(H-M)(H-N)}{H^{\acute{e}}(H-1)} = Np(1-p)\dfrac{H-N}{H-1}$ (4-18)

The comparison with the binomial distribution of same parameter p shows that the average values are identical, while for variances:

$$\frac{Var(HyperG)}{Var(Binom)} = \frac{H-N}{H-1} \qquad (4\text{-}19)$$

As the population size keeps growing ($H \to \infty$), the ratio goes to 1: for a population of large size, extracting N elements brings no significant change to the composition of the remaining elements. In other words, the hypergeometric distribution goes to the binomial law as the population size increases indefinitely.

4.6. The Poisson law

A discrete random variable taking unbounded positive values obeys a Poisson law with parameter λ if its distribution is given by:

$$P(k) = \frac{\lambda^k}{k!}e^{-\lambda} \qquad (4\text{-}20)$$

The dimensionless parameter λ characterises the law. This distribution is encountered in various circumstances. Especially, if a flow arrives according to a Poisson process (see Chapter 7), then the number of arrivals observed in a window of width T is distributed according to the Poisson law with parameter $\Lambda = \lambda T$. This gives to this law a fundamental role in teletraffic studies, especially for describing arrival of calls, sessions, etc., in communications systems. The conditions explaining the occurrence of such a law are detailed in Chapter 7.

The characteristic function, from which the moments are derived, is written:

$$\phi(z) = \sum_i e^{kz}\frac{\lambda^k}{k!}e^{-\lambda} = e^{\lambda(e^Z-1)} \qquad (4\text{-}21)$$

Mean value: $m = \sum kp_k = \phi'(z=0) = \lambda e^z e^{\lambda(e^z-1)} = \lambda$ (4-22)

Variance: $Var = \sum (k-m)^2 p_k = \lambda$ (4-23)

The central moments of higher order are:

$$\mu_3 = \lambda,\ \mu_4 = \lambda + 3\lambda^2 . \tag{4-24}$$

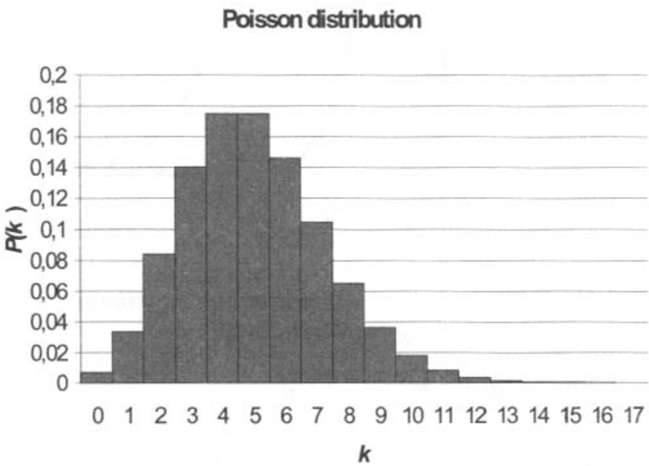


Figure 4.2a. *The histogram of the Poisson distribution (λ = 5)*

Figures 4.2a and 4.2b illustrate the general shape of the law. As this is a discrete distribution, the histogram resembles Figure 4.2a. In Figure 4.2b a smoothed variant shows the effect of changing the parameter. Variables obeying a Poisson distribution enjoy the following property, allowing combining them easily.

THEOREM.– Let X and Y be two Poisson variables with parameters respectively λ and μ.. Then, $X + Y$ obeys a Poisson distribution with parameter $\lambda + \mu$.

The proof is immediate, using the transform approach explained in Chapter 3, the transform of the sum being the product of transforms (Laplace, characteristic function, etc.). This property is of the higher importance, as one has often to treat the case of a mix of Poisson streams.

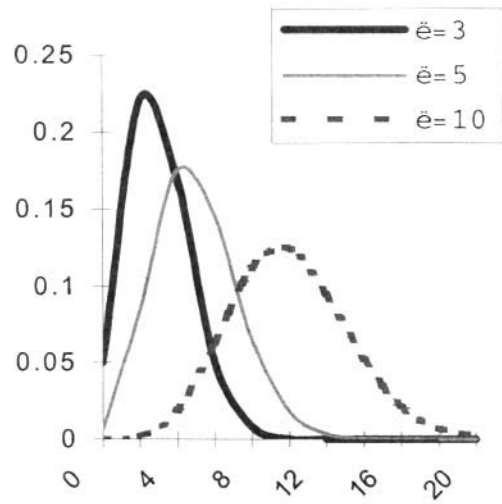


Figure 4.2b. *Smoothed shape of the Poisson distribution for different parameter values*

4.6.1. *Relationship with the binomial law*

When the size of the population grows indefinitely, the Poisson law provides a useful approximation. Consider a binomial law, with average A and with growing population size N (so that $p = A/N$ goes to 0).

$$P(X=k)=\binom{N}{k}\left(\frac{A}{N}\right)^{k}\left(1-\frac{A}{N}\right)^{N-k}.$$

Developing the binomial coefficient and rearranging the terms, one gets:

$$P(X=k)=\frac{A^k}{k!}\left(1-\frac{A}{N}\right)^{N}\left(1-\frac{A}{N}\right)^{-k}\times\frac{N}{N}\times\frac{N-1}{N}\times\ldots\times\frac{N-k+1}{N}.$$

We make N going to infinity, keeping A constant (thus p going to 0). Then:

$$\frac{A}{N}\to 0;\ \left(1-\frac{A}{N}\right)^{N}\to e^{-A};\ \left(1-\frac{A}{N}\right)^{k}\to 1;\ \frac{N}{N}\times\frac{N-1}{N}\times\ldots\times\frac{N-k+1}{N}\to 1$$

And so:

$$P(X=k) \to \frac{A^k}{k!}e^{-A}.$$

As the sample size increases, the average remaining constant, the limiting distribution conforms to the Poisson law. This is an appealing result, as the Poisson approximation is much easier to tabulate. Practically, the approximation holds as soon as $N > 40$ and $p < 0.1$.

Actually, the Poisson distribution may serve as a limit for several other laws, such as Erlang distribution, as will be seen later on. This explains its use in a lot of domains, such as statistics, reliability (e.g. for estimating spare parts), and also naturally traffic studies. We will present in Chapter 7 an interesting interpretation of a system with N customers and R servers, which allows explicating the natural relation between Poisson, binomial and Engset laws.

4.7. The continuous uniform distribution

This is the simplest kind of continuous distribution, which corresponds to the limiting case of the discrete one. A random variable is uniformly distributed between a and b if the probability of finding it in any interval of length l is $l/(b-a)$. As the law is absolutely continuous, the probability density exists:

$$f(x)dx = P(x \le x < x+dx) = \frac{dx}{b-a}; \tag{4-25}$$

and the distribution is:

$$\begin{aligned} P(x) &= 0 && x < a, \\ &= \frac{x-a}{b-a} && a \le x \le b, \\ &= 1 && x > b. \end{aligned} \tag{4-26}$$

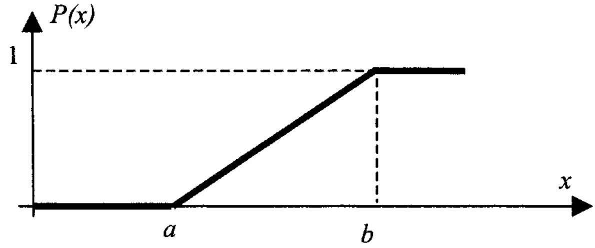


Figure 4.3. *The uniform distribution*

Such a distribution has a great variety of usages. This may represent a service duration, for instance if the service epoch occurs periodically, with a constant interval T, the customers arriving randomly (packeting of speech samples). This is also the law which will serve as a basis for generating random variables with arbitrary distributions in simulation (see Chapter 8), and more generally in all Monte-Carlo methods.

The characteristic function of the uniform distribution is:

$$\phi(z) = \frac{e^{bz} - e^{az}}{(b-a)z}, \tag{4-27}$$

from which the moments are obtained (e.g. by developing the exponential functions):

Mean value: $E(X) = \phi'(0) = \dfrac{a+b}{2}$, and (4-28)

$$Var(X) = \phi''(0) - [\phi'(0)]^2 = \frac{(b-a)^2}{12}. \tag{4-29}$$

4.8. The normal (Gaussian) distribution

The normal (or Gaussian) distribution is probably the most celebrated among the laws that the engineers and scientists manipulate. Its popularity rests to a large extend on work which has been devoted to measures and the theory of error measurements. Its justification relies on the central limit theorem, presented in Chapter 5, and which states that the normal distribution is the limiting distribution of the sum of a large number of independent and identically distributed variables. For instance, the causes of measurement errors are numerous and independent, and they add so that the measured value is normally distributed around the "true" value.

In the field of performance studies, its applications are numerous, for reliability, queueing, or simulation, where it allows estimation of the "precision" of the measurements as well as approximating probability distribution. Actually, the analysis of most physical phenomena involves the addition of a large number of independent variables – number of waiting customers, number of machines down, etc. – making it possible to invoke the central limit theorem.

A random variable is said to be normally distributed with mean m and standard-deviation σ, and usually denoted as $N(m,\sigma^2)$ if its probability distribution is given by:

$$P(X \le x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} e^{-(x-m)^2/2\sigma^2} \, dx, \; -\infty < x < \infty. \tag{4-30}$$

The coefficient before the integral guarantees the "normalisation", so that $P(X \le \infty) = 1$ as needed. The term σ, the standard-deviation, is responsible of the flattening of the density around the mean (see Figure 4.4). The expression $\sigma_r = \frac{\sigma}{m}$ is referred to as the *relative dispersion,* or *coefficient of variation* (also denoted as c).

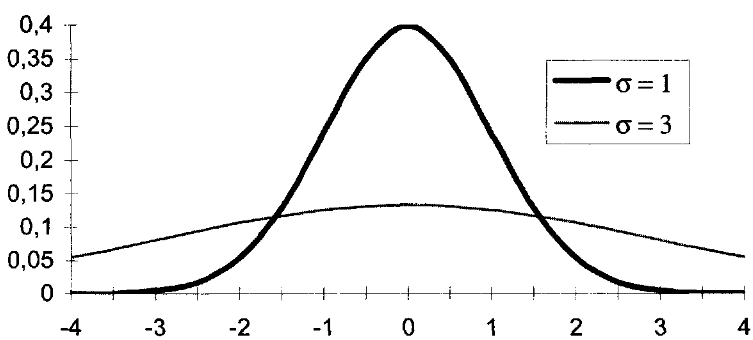


Figure 4.4. *Probability density function of the normal law*

4.8.1. *The sum of normal random variables*

With reference to the properties of the sum of independent variables, stated in Chapter 3, the following result allows easy manipulation of the sum of normal variables.

THEOREM.– Let $X = N(m_1, \sigma_1)$ and $Y = N(m_2, \sigma_2)$ be two independent random variables. The sum $X + Y$ is distributed according to $N(m_1 + m_2, \sqrt{\sigma_1^2 + \sigma_2^2})$.

This clearly generalises to the sum of an arbitrary number of variables. Of special interest is the case of *n* variables identically distributed: the coefficient of variation of the sum is $\sigma_r = \frac{\sigma}{m\sqrt{n}}$. As *n* grows, the relative dispersion vanishes so that the sum goes to a variable almost constant.

The curves in Figure 4.4 show how the dispersion varies as the standard deviation increases. Figure 4.5 illustrates the symmetrical shape of the distribution function.

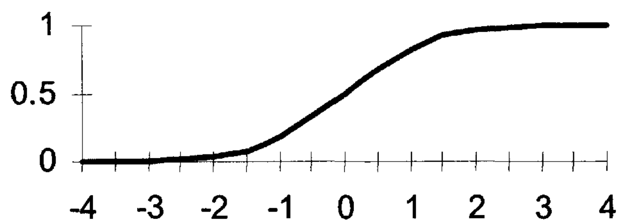


Figure 4.5. *Distribution of the normal law (standard-deviation = 1)*

In fact, the normal distribution is scarcely used this way. If X is normally distributed according to $N(m, \sigma^2)$, then clearly $X - m$ has a mean = 0, and the same variance as X. Moreover, $(X - m)/\sigma$ is a variable with variance equal to 1. This is the *standard* or *reduced normal distribution* (Laplace law). This form is "universal", in that it does not depend on any parameter, which allows tabulation of it. It is usually written as:

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{x^2}{2}} dx, \quad \varphi(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}, \qquad \text{(4-31)}$$

and the characteristic function:

$$\varphi(z) = e^{\frac{z^2}{2}}. \tag{4-32}$$

A direct numerical evaluation of $\Phi(x)$ happens to be difficult, except when using specific mathematical libraries, and its estimation is most often based upon numerical tables. The following approximation is quite efficient:

$$\Phi(x) = 1 - \varphi(x)\left[au^2 + bu + c\right],\ u = \frac{1}{1 + 0.33267x}, \tag{4-33}$$

$$\text{with}\ \begin{array}{l} a = -0.1201676 \\ b = 0.4361836 \\ c = 0.9372980 \end{array}$$

This expression deviates from the exact value by a relative error of about 10^{-5}.

4.8.2. *Statistical tables*

All reference textbooks in statistics provide comprehensive tables for the standardized normal distribution. An example of such tables is given in Appendix 2.

The table gives the bounds of the interval $\left[-a, a\right]$, such as, 'the value of a normal random variable has the given probability P to be outside the interval'. This can be written as $P = P(|X| > a)$. In other words, the value is within the interval with probability 1 – P. For instance, 1% of the values are likely to be outside the range [–2.576, 2.576] and about half of them are outside [–0.674, 0.674]. As the law is symmetrical, the probabilities for each side of the bounds are the same:

$$P(X < -2{,}576) = P(X > 2{,}576) = 0{,}005$$

Such tables are built only for the standard distribution. Values for an arbitrary law may be derived, if needed. For instance, let X have a normal distribution with $m = 10, \sigma = 15$. What is the probability of observing $X > 30$? The idea is to come back to the reduced variable:

$$P(X>30)=P\left(\frac{X-10}{15}>\frac{30-10}{15}=2\right)$$

The value we are looking for is the probability that the reduced variable is larger than 2. The table says that the probability is around 0.045/2 $\approx$ 0.023 (as $P(|X|>2)=0.045$).

4.8.3. *The normal law as limiting distribution*

The normal law happens to be the limiting distribution of numerous distributions, either continuous or discrete, when one of their parameter increases. For discrete distributions already encountered:

– Binomial law: Z = $B(n,p)$, then $(Z-np)/\sqrt{np(1-p)}$ goes to $N(0,1)$ as $n \to \infty$; practically, one admits generally that the approximation holds as soon as p > 0,5 and np > 5, or p < 0,5 and $n(1-p)$ > 5;

– Poisson law: Z = $P(\lambda)$, then $(Z-\lambda)/\sqrt{\lambda}$ converges towards $N(0,1)$ as $\lambda \to \infty$. The condition $\lambda > 10$ ensures the validity of the approximation.

Other continuous distributions, introduced in the next paragraphs, also have the normal law as limiting distribution.

4.9. The Chi-2 distribution

We are given a set of independent and identically distributed variables $X_1, X_2, \ldots, X_n$, following the reduced normal distribution.

The variable $\chi^2 = X_1^2 + X_2^2 + \ldots + X_n^2$ is distributed to as a Chi-2 with n degrees of freedom (written χ_n^2). This distribution enjoys the following property: the sum of two Chi-2 variables with respectively n and m degrees of freedom is a Chi-2 with $n+m$ degrees of freedom. This variable is often used in statistics, especially for hypothesis testing or estimating confidence intervals (see Chapter 5). Especially, it is useful for the estimation of operational failure rates of the components, when monitoring reliability of equipments in operation. The density function has the form:

$$f_n(\chi^2) = \frac{e^{-\frac{\chi^2}{2}} \chi^{2^{\frac{n}{2}-1}}}{2^{\frac{n}{2}} \Gamma(\frac{n}{2})} \text{ for } \chi^2 \geq 0 . \tag{4-34}$$

Mean value: $m = n$, (4-35)

Variance: $Var = 2n$. (4-36)

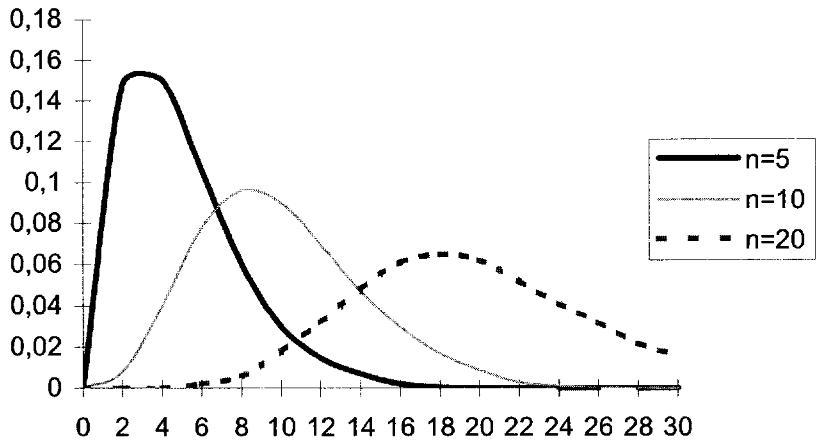


Figure 4.6. *Probability density function of the Chi-2 law*

4.9.1. *Limiting behaviour*

Since a Chi-2 variable is the sum of independent variables, it clearly should go to a normal distribution as the number of degrees of freedom increases. A more precise result can be stated:

THEOREM.– Let X_n be a variable distributed according to a Chi-2 with n degrees of freedom. As n increases:

$$\frac{X_n - n}{\sqrt{2n}} \to N(0,1) \text{, as } n \to \infty \tag{4-37}$$

4.10. The Student distribution

We are given a set of $n + 1$ independent random variables, $(X, X_1, X_2, ..., X_n)$, each normally distributed with mean $m = 0$ and the same variance. Let:

$$Y=\sqrt{\frac{1}{n}\sum X_i^2}\,,\ t=\frac{X}{Y} \tag{4-38}$$

The variable t is distributed according to the *Student law with n degrees of freedom*. This distribution appears in the theory of estimation, especially for samples of limited size.

The density function is given by:

$$f_n(t)=\frac{1}{\sqrt{n\pi}}.\frac{\Gamma(\frac{n+1}{2})}{\Gamma(\frac{n}{2})}\left(1+\frac{t^2}{n}\right)^{-\frac{n+1}{2}} \tag{4-39}$$

For $n>2$, the moments exist and are given by:

Mean value: $m=0$, (4-40)

Variance: $\mathrm{var}=\dfrac{n}{n-2}$ (4-41)

The variance does not depend on the common variance of the X_i. As the number of degrees of freedom increases, the Student distribution approaches the reduced normal distribution (intuitively Y goes to a constant equal to unity).

4.11. The lognormal distribution

A variable is lognormally distributed if its logarithm has a normal distribution. If Y is $N(m,\sigma^2)$-distributed, then $X=e^Y$ is lognormally distributed with density function:

$$f(x)=\frac{1}{\sigma\ x\sqrt{2\pi}}e^{-(\ln x-m)^2/2\sigma^2}\,,\ \text{for}\ 0\le x<\infty \tag{4-42}$$

The moments of the two distributions X and Y are related:

$$E(X)=e^{m+\sigma^2/2} \tag{4-43}$$

$$Var(X) = \sigma_X^2 = (e^{\sigma^2} - 1)e^{2m+\sigma^2} \tag{4-44}$$

One usually proceeds the inverse way: the parameters of X are known and one looks for the (m,σ) needed, e.g. to generate a sample in a simulation experiment. The correspondence is simply:

$$\sigma^2 = Log\left(1 + \frac{Var(X)}{E(X)^2}\right)$$
$$m = Log\left(E(X)\right) - \frac{\sigma^2}{2}$$

The reason why the lognormal distribution is encountered is described as the principle of *multiplicative accumulation*. The normal distribution appears naturally when a phenomenon results in the *sum* of independent perturbations. Assume now the amplitude of the phenomenon is caused by the *product* of independent causes. Taking the logarithm transforms the products into sums, on which the arguments of the central limit theorem apply. The lognormal distribution is thus invoked in the analysis of a large number of economic phenomena related to income or consumption, or in life sciences.

4.12. The exponential and related distributions

4.12.1. *The exponential distribution*

The exponential distribution has a quite particular position in the queueing and reliability domains. The corresponding chapters will explain the reason for its endless use. The distribution depends on a single parameter, traditionally denoted as μ in the teletraffic field, and the density function is given by:

$$f(x) = \mu e^{-\mu x}, \text{ for } x \geq 0. \tag{4-45}$$

The distribution function is:

$$F(x) = 1 - e^{-\mu x}. \tag{4-46}$$

Remember its characteristic function (see Chapter 3):

$$\phi(z) = \frac{\mu}{\mu - z}, \tag{4-47}$$

for which the moments are easily obtained:

Mean: $E(x) = \dfrac{1}{\mu}$ (4-48)

Variance: $Var(x) = \dfrac{1}{\mu^2}$ (4-49)

The curve $k = 1$ in Figure 4.7 shows the shape of its density function (as this is a special case of the Erlang-k distribution, see below).

4.12.2. *The Erlang-k distribution*

A variable which is the sum of k independent variables having the same exponential distribution is said to have an *Erlang-k* distribution. It can serve for approximating an unknown distribution, having a coefficient of variation lower than 1. This helps in building a model of service durations, having dispersion between the constant and exponential distributions.

For the simplest case let us take $k = 2$. Let X denote the variable sum of the two variables X_1, X_2, having probability distributions B_1 and B_2. The distribution B of X is the *convolution* of B_1 and B_2 (see Chapter 3):

$$P(X \le x) = P(X_1 + X_2 \le x) = \int_{u=0}^{x} B_1(x-u) b_2(u) \mathrm{d}u .$$

As X_1, X_2 are exponentially distributed, it becomes:

$$P(X \le x) = 1 - e^{-\mu x} - \mu x e^{-\mu x} .$$

More generally, one has shown in Chapter 3 that cascading k exponential variables having the same parameter μ leads to the distribution:

$$B(x) = P(X_1 + X_2 + \ldots + X_k \le x) = 1 - e^{-\mu x} \sum_{j=0}^{k-1} \frac{(\mu x)^j}{j!} . \qquad (4\text{-}50)$$

This distribution is known as the *Erlang-k* distribution. As it is the sum of independent variables, the mean and variance are simply the sums of the mean and variance of each component:

Mean value: k/μ. (4-51)

Variance: k/μ^2. (4-52)

Coefficient of variation: $1/\sqrt{k}$. (4-53)

Using the Laplace transform makes it easy to retrieve these results. As the distribution is the convolution of k exponential distributions of independent variables, its transform is directly obtained:

$$B^*(s) = \left(\frac{\mu}{\mu+s}\right)^k. \tag{4-54}$$

From the distribution, the moments are obtained by taking derivatives for s = 1. Moreover, coming back to the original function provides the probability density:

$$f(x) = \mu e^{-\mu x}\frac{(\mu x)^{k-1}}{(k-1)!}. \tag{4-55}$$

Note that the coefficient of variation $c = 1/\sqrt{k}$ is always less than 1, except for k = 1, which corresponds to the exponential function. This explains the use of this distribution as a means to represent a phenomenon having a low dispersion. It can even be used with a large value for k, keeping k/μ constant, providing then an approximate representation of constant duration under the form of a markovian system, thus keeping the memoryless property, the appeal of which will be made obvious in Chapter 7 devoted to Markov processes (it can be verified that the limit of equation 4.55 as k grows is indeed the constant distribution).

The Laplace transform provides an elegant way to prove this limit. Let $a = k/\mu$ kept constant as k increases:

$$B^*(s) = \left(\frac{k/a}{k/a+s}\right)^k = \left(1+\frac{as}{k}\right)^{-k}, \text{ and so:}$$

$$\lim_{k\to\infty} B^*(s) = \lim\left(1+\frac{as}{k}\right)^{-k} = e^{-as},$$

which is actually the transform of the constant duration with parameter a.

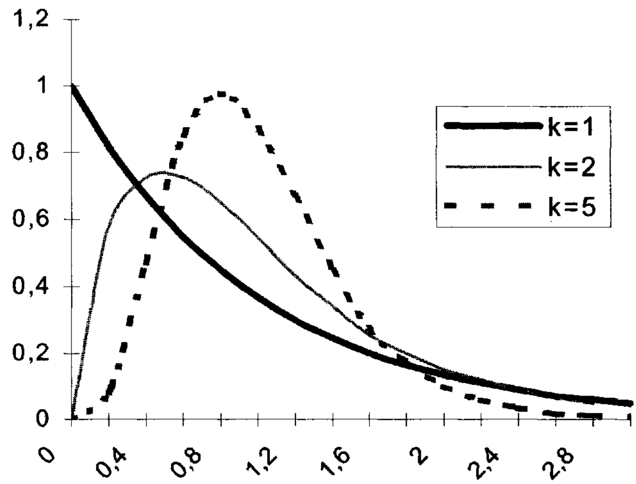


Figure 4.7. *Probability density function of the Erlang-k distribution for various k*

For practical purposes, one can introduce the new parameter $X = \mu x / k$, such that the new variable has a mean value equal to 1. One thus gets the reduced law, just as was done for the normal distribution:

$$f(X) = ke^{-kX} \frac{(kX)^{k-1}}{(k-1)!}. \qquad (4\text{-}56)$$

having the characteristic function:

$$\phi(z) = (1 - \frac{z}{k})^{-k}, \qquad (4\text{-}57)$$

and moments:

$$E(X) = 1, \; Var(X) = 1/k \qquad (4\text{-}58)$$

Figure 4.7 exhibits the shape of this distribution, for different values of k.

Taking for k a real number instead of a positive integer generalizes to the *Gamma distribution.* This is presented hereafter. Note also that for statisticians Erlang-k distribution is a χ^2 distribution with k degrees of freedom.

4.12.3. *The hyperexponential distribution*

When it comes to represent service distributions having a coefficient of variation larger than 1, the hyperexponential distribution is introduced, as it will be discussed in Chapter 7. The basic configuration depends on 3 parameters, usually denoted as α, μ_1, μ_2. The distribution is:

$$P(X \le x) = \alpha\left(1 - e^{-\mu_1 x}\right) + (1-\alpha)\left(1 - e^{-\mu_2 x}\right) \tag{4-59}$$

The Laplace transform is readily obtained, as the sum of the transforms of the 2 exponential functions:

$$B(s) = \frac{\alpha\mu_1}{s+\mu_1} + \frac{(1-\alpha)\mu_2}{s+\mu_2}. \tag{4-60}$$

From the transform, mean and variance are:

$$m = \frac{\alpha}{\mu_1} + \frac{1-\alpha}{\mu_2}, \quad \mathrm{var} = 2\left[\frac{\alpha}{\mu_1^2} + \frac{1-\alpha}{\mu_2^2}\right] - \left[\frac{\alpha}{\mu_1} + \frac{1-\alpha}{\mu_2}\right]^2.$$

Coefficient of variation: $c^2 = \dfrac{2\left[\dfrac{\alpha}{\mu_1^2} + \dfrac{1-\alpha}{\mu_2^2}\right]}{\left[\dfrac{\alpha}{\mu_1} + \dfrac{1-\alpha}{\mu_2}\right]^2} - 1$

In the general case, the distribution is a combination of n exponential functions:

$$P(X \le x) = \sum_k \alpha_k\left(1 - e^{-\mu_k x}\right), \text{ with } \sum_k \alpha_k = 1, \tag{4-61}$$

the moments of the distribution are:

$$\text{Mean: } m = \sum_k \frac{\alpha_k}{\mu_k}, \text{ variance: } \mathrm{var} = 2\sum_k \frac{\alpha_k}{\mu_k^2} - \left(\sum_k \frac{\alpha_k}{\mu_k}\right)^2. \tag{4-62}$$

$$\text{Coefficient of variation: } c^2 = \frac{2\sum \frac{\alpha_k}{\mu_k^2}}{\left[\sum \frac{\alpha_k}{\mu_k}\right]^2} - 1 \tag{4-63}$$

The Cauchy inequality states that $\left(\sum a_i b_i\right)^2 \le \sum a_i^2 . \sum b_i^2$. Applying it to the above expression, taking $a_i = \sqrt{\alpha_i}$, $b_i = \sqrt{\alpha_i} / \mu_i$ shows that the coefficient of variation is always larger than 1. This makes this law a representative of distribution more dispersed than the exponential one.

4.12.4. *Generalising: the Cox distribution*

The Cox distribution provides a generalisation of the Erlang and hyperexponential families. The following picture (Figure 4.8) visualises this law: a network of exponential servers, the "service" being composed of a random travel through the network.

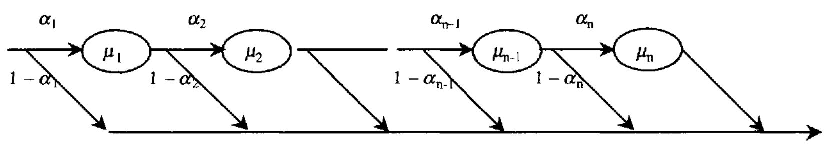


Figure 4.8. *A network of exponential servers generating the Cox distribution*

One can show that this formulation provides the most universal combination of exponential distributions, and that it can represent exactly any distribution for which the Laplace transform can be written as the ratio of two polynomials. As a consequence, any probability distribution can be approximated by such a law, the accuracy depending on the number of terms. The Laplace transform is easily written:

$$B(s) = 1 - \alpha_1 + \sum_{i \le n} \prod_{j=1}^{i} \frac{\alpha_j \mu_j}{s + \mu_j} (1 - \alpha_{i+1}) . \tag{4-64}$$

4.12.5. *The Gamma distribution*

A variable having a Gamma distribution $\gamma(\alpha, \beta)$ has a probability density given by:

$$f(x) = e^{-x/\beta} \frac{x^{\alpha-1}}{\beta^{\alpha}\Gamma(\alpha)}, \ \alpha, \beta > 0, \ x \geq 0, \tag{4-65}$$

in which Γ is the Gamma function: $\Gamma(\alpha) = \int_0^{\infty} x^{\alpha-1} e^{-x} dx$ (for integer values of n, $\Gamma(n) = (n-1)!$). The reduced form with $\beta = 1$ can be used: it is obtained by scaling the distribution: if X is distributed according to $\gamma(\alpha, \beta)$, then $Y = X / \beta$ is distributed according to $\gamma(\alpha, 1)$ – usually denoted as $\gamma(\alpha)$.

$$\text{Mean: } m = \alpha\beta. \tag{4-66}$$

$$\text{Variance: } Var = \alpha\beta^2. \tag{4-67}$$

Reciprocally, given the moments, the parameters are obtained as $\beta = Var / m$, $\alpha = m^2 / Var$.

For $\alpha = 1$, it reduces to the exponential distribution. More generally, $\gamma(k, \beta)$ with integer k is the Erlang-k distribution. For k integer and $\beta = 2$, this is a Chi-2 with $2k$ degrees of freedom.

If two variables X and Y, are Gamma distributed with the same parameter β – respectively $\gamma(\alpha, \beta)$ and $\gamma(\alpha', \beta)$ –, then $Z = X + Y$ is Gamma distributed as $\gamma(\alpha + \alpha', \beta)$.

The Gamma distribution allows one to represent various distributions, for which the analytical expression is unknown or too complex to be numerically manipulated. It provides an extension of the Erlang family to configurations with non integer parameters, giving arbitrary variances (larger or smaller than the mean).

Taking n as an integer, one verifies that the expression reduces to the Erlang-k already presented, making the obvious change of variable $k \to n$ $x \to \theta = kx/n$. It is under this form that the Gamma distribution is often calculated.

$$f(\theta) = ne^{-n\theta} \frac{(n\theta)^{n-1}}{(n-1)!} \tag{4-68}$$

This is the reduced form. One has:

$$\phi(z) = (1 - z/n)^{-n} \tag{4-69}$$

$$m = 1, \ Var = 1/n \tag{4-70}$$

4.13. The Weibull distribution

A variable obeys the Weibull distribution if its probability density function is given by:

$$f(x) = \frac{\beta}{\delta}\left(\frac{x}{\delta}\right)^{\beta-1} e^{-\left(\frac{x}{\delta}\right)^{\beta}}, \ F(x) = 1 - e^{-\left(\frac{x}{\delta}\right)^{\beta}} \tag{4-71}$$

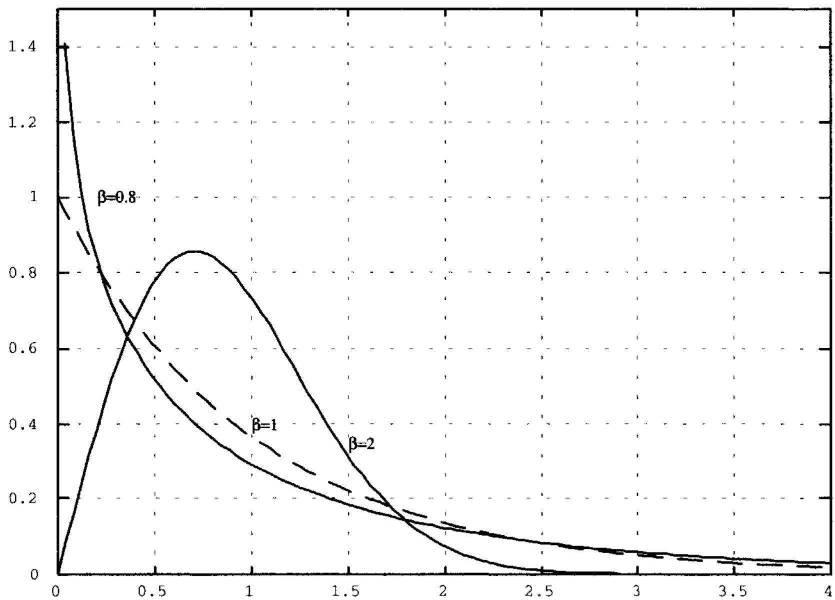


Figure 4.9. *Probability density of the Weibull distribution*

It is of common use in reliability studies, where the *reliability function R(t)* is defined: this is the probability that the system is working correctly at time *t*. *R(t)* is the complement of the distribution (see Chapter 6):

$$R(t) = 1 - F(t) = e^{-\left(\frac{t-\gamma}{\eta}\right)^{\beta}}. \tag{4-72}$$

It represents distributions with reliability parameters being a function of time. For example: the reliability increases in the beginning of the life, $\beta < 1$ (failure rate decreasing as the time increases), constant failure rate during the operational period, taking $\beta = 1$, $\gamma = 0$ (the exponential distribution), and the failure rate increasing at the end of the lifetime, with $\beta > 1$ (see Chapter 6). It can be of help in traffic studies too, for describing complex arrival processes, such as interarrival duration between successive IP flows.

The moments of the Weibull distribution are:

Mean value: $m = \delta \ \Gamma(1+1/\beta)$, (4-73)

Variance: $Var(x) = \delta^2 \left[\Gamma(1+2/\beta) - \Gamma(1+1/\beta)^2 \right]$. (4-74)

4.14. The logistic distribution

The logistic distribution appears when modelling the evolution of a character of a population, and more specifically here when deploying new services. Indeed, this is the classical model for the introduction of any consumer goods. To have a precise example, let us consider a communication service (e.g. cellular phone). Let x be the equipment ratio, i.e. the density of the equipment at time t (ratio of the number of equipped customers to the total population). Let s stand for the *saturation density* (it can be 100%, or even more if the households are the reference population: e.g. 2 telephones in a single family)

The derivative dx/dt accounts for the growth rate. Assume the demand can be immediately satisfied: the equipment ratio is equal to the demand. Assume now that the demand is proportional to the number of customers already equipped. Assume also that it is proportional to the size of the unequipped population. The first assumption is related to the *service usefulness*, as more and more customers can be reached; the second assumption amounts to defining an individual need of the service, the demand being thus proportional to the size. The same kind of assumptions is made in epidemiological studies (x being here the density of contaminated subjects)

Under these assumptions, and with k being a coefficient assumed constant, the phenomenon is governed by the following equation:

$$\frac{dx}{dt} = kx(s-x).$$

The solution is readily obtained:

$$x = \frac{s}{1 + Ae^{-kst}} \text{ or } x = \frac{s}{1 + e^{-k(t-t_0)}}. \tag{4-75}$$

In this equation, A stands for the integration constant. The second form, equivalent to the first one, introduces t_0, time of median equipment (epoch where $x = s/2$).

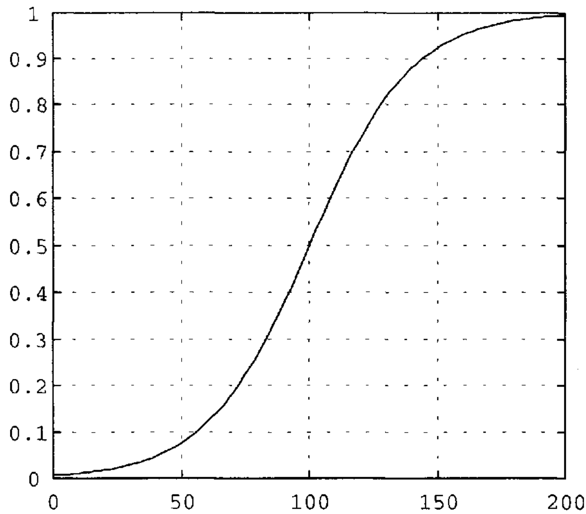


Figure 4.10. *The logistic curve*

A change in the coordinates is most often use, taking $\mathrm{Log}(\frac{x}{s-x})$ instead of x. This is the *logit transformation*, which reduces the typical logistic curve of Figure 4.10 to a straight line. This provides a framework of description, rather than a model of explanation. In practice, several observations allow one to estimate the best values for the parameters. The validity of this model depends actually on the (questionable) validity of all the assumptions (for instance, k is taken as constant, which is probably erroneous for society phenomena). Figure 4.10 displays the shape of the curve, for $s = 1$, $k = 0{,}05$, $t_0 = 100$.

4.15. The Pareto distribution

The Pareto distribution is given by the following law:

$$F(x) = 1 - \left(\frac{b}{x}\right)^{a}, \text{ with } a,b > 0,\ x \geq b. \tag{4-76}$$

Taking the derivative gives the density function:

$$f(x) = \frac{dF}{dx} = \frac{ab^{a}}{x^{a+1}}. \tag{4-77}$$

If the curve is drawn on a logarithmic scale the parameters are readily identified:

$$\log(F(x)) = a\log(b) - a\log(x),$$

so the curve appears as a straight line with slope $-a$.

$$\text{Mean: } \frac{ab}{a-1}, \tag{4-78}$$

$$\text{Variance: } \frac{ab^{2}}{(a-2)(a-1)^{2}}. \tag{4-79}$$

The Pareto distribution has been initially used in statistics. Recent studies about internet traffic has lead to applying it to traffic description. It has been shown that some of the characteristics of these flows are fairly well modelled using this law. This is especially the case for the length of the files exchanged on the network, and the *self-similarity* one observes on internet traffic could originate in this remark. Self similarity accounts for the observation that the statistical properties of the observation are conserved through changes of scale. The term *fractal* traffic is also in use.

Figure 4.11 displays the shape of the complementary distribution function, plotted on a logarithmic scale (the unit length on the abscissa is proportional to x). The curve "b = 2350, a = 1.04" corresponds to the adjustment with a series of observations of file sizes observed on the internet (reported in [CRO 97]).

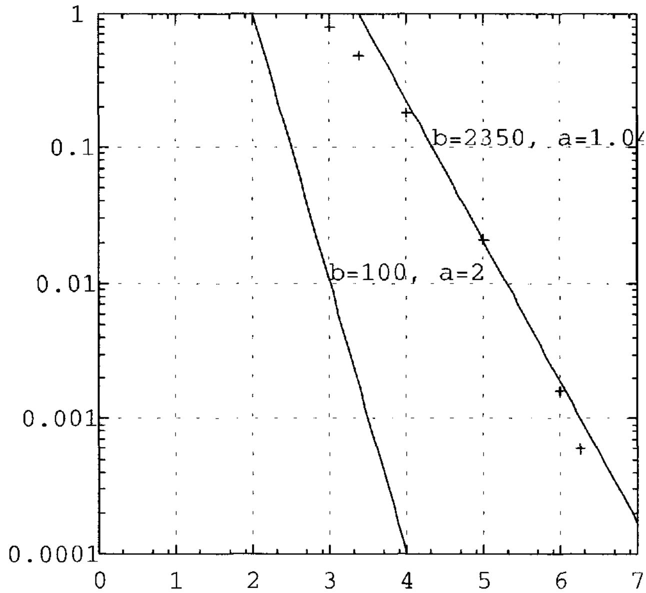


Figure 4.11. *The Pareto distribution*

4.16. A summary of the main results

Mean values are denoted as m and variances as *Var*. μ_3, μ_4 are the central moments of order 3 and 4. The generating function *B(z)* and the characteristic function $\phi(z)$ are given. Most of these laws are tabulated: the reader is referred to Appendix 2.

4.16.1. *Discrete distributions*

Uniform (discrete) distribution U(a,b)

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

$$B(z)=\frac{z^a-z^{b+1}}{(b-a+1)(1-z)},\ \phi(z)=\frac{e^{az}-e^{(b+1)z}}{(b-a+1)(1-e^z)}$$

$$m=\frac{a+b}{2},\ Var=\frac{(b-a)(b-a+2)}{12},$$

$$\mu_3 = 0,\ \mu_4 = \frac{(b-a)^4}{80} + \frac{(b-a)^3}{20} + \frac{(b-a)(b-a+1)}{30}$$

Binomial distribution B(p, N)

$$P(k) = P(k\, among\, N) = \binom{N}{k} p^k (1-p)^{N-k}, \quad k = 0,\ldots,N$$

$$B(z) = (1-p+pz)^N,\ \phi(z) = (1-p+pe^z)^N$$

$$m = Np,\ Var = Np(1-p),\ \mu_3 = Np(1-p)(1-2p),$$

$$\mu_4 = Np(1-p)\left[1+(3N-6)p-(3N-6)p^2\right]$$

Geometric distribution

$$p_k = (1-p)p^k,\ 0<p<1,\ k=0,1,\ldots$$

$$B(z) = \frac{1-p}{1-pz},\ \phi(z) = \frac{1-p}{1-pe^z}$$

$$m = \frac{p}{1-p},\ Var = \frac{p}{(1-p)^2}$$

Hypergeometric distribution

$$P(k) = \frac{\binom{M}{k}\binom{H-M}{N-k}}{\binom{H}{N}} \qquad \text{for } \max(0, N+M-H) \le k \le \min(M,N)$$

$$m = Np,\ Var = Np(1-p)\frac{H-N}{H-1}$$

Poisson distribution

$$P(k) = \frac{\lambda^k}{k!}e^{-\lambda}$$

$$B(z)=e^{\lambda(z-1)},\ \phi(z)=e^{\lambda(e^{z}-1)}$$

$$m=\lambda,\ Var=\lambda,\ \mu_3=\lambda,\ \mu_4=\lambda+3\lambda^2$$

4.16.2. *Continuous distributions*

The density and the distribution are given when the expressions are simple enough to be of some help.

Uniform distribution U[a,b)

$$f(x)dx=P(x\le x<x+dx)=\frac{dx}{b-a};\ \text{with } P(x)=\begin{cases}0 & x<a\\ \dfrac{x-a}{b-a} & a\le x<b\\ 1 & x\ge b\end{cases}$$

$$\phi(z)=\frac{e^{bz}-e^{az}}{(b-a)z}$$

$$m=\frac{a+b}{2},\ Var=\frac{(b-a)^2}{12},\ \mu_3=0,\ \mu_4=\frac{(b-a)^4}{80}$$

Normal (Gauss) distribution $N(m,\sigma)$

$$P(X\le x)=\frac{1}{\sigma\sqrt{2\pi}}\int_{-\infty}^{x}e^{-(x-m)^2/2\sigma^2}\ dx,\ -\infty<x<\infty$$

$$\phi(z)=e^{zm+\frac{(z\sigma)^2}{2}}$$

(mean and variance are the parameters of the law), $\mu_3=0,\ \mu_4=3\sigma^4$

Chi-2 distribution with n degrees of freedom

$$f_n(\chi^2)=\frac{e^{-\frac{\chi^2}{2}}\chi^{2\left(\frac{n}{2}-1\right)}}{2^{\frac{n}{2}}\Gamma(\frac{n}{2})}\ \text{ for } \chi^2\ge 0$$

$$\phi(z) = (1-2z)^{-n/2}$$

$$m = n \,,\ Var = 2n$$

Student distribution

$$f_n(t) = \frac{1}{\sqrt{n\pi}} . \frac{\Gamma(\frac{n+1}{2})}{\Gamma(\frac{n}{2})} \left(1+\frac{t^2}{n}\right)^{-\frac{n+1}{2}}$$

$$m = 0 \,,\ Var = \frac{n}{n-2} \,,\ \mu_3 = 0$$

Log-normal distribution

$$f(x) = \frac{1}{\sigma \ x\sqrt{2\pi}} e^{-(\ln x - m)^2/2\sigma^2}$$

$$m = e^{m+\sigma^2/2} \,,\ Var = (e^{\sigma^2} - 1)e^{2m+\sigma^2}$$

Exponential distribution

$$f(x) = \mu e^{-\mu x} \,,\ F(x) = 1 - e^{-\mu x}$$

$$\phi(z) = \frac{\mu}{\mu - z}$$

$$m = \frac{1}{\mu} \,,\ Var = \frac{1}{\mu^2} \,,\ \mu_3 = \frac{2}{\mu^3} \,,\ \mu_4 = \frac{9}{\mu^4}$$

Erlang-k distribution

$$F(x) = P(X_1 + X_2 + \ldots + X_k \le x) = 1 - e^{-\mu x} \sum_{j=0}^{k-1} \frac{(\mu x)^j}{j!} \quad f(x) = \mu e^{-\mu x} \frac{(\mu x)^{k-1}}{(k-1)!}$$

$$\phi(z)=\left(\frac{\mu}{\mu-z}\right)^{k}$$

$$m=k/\mu\,,\ Var=k/\mu^{2}$$

Hyperexponential distribution

$$P(X\le x)=\sum_{k}\alpha_k\left(1-e^{-\mu_k x}\right),\ \phi(z)=\sum_{k}\frac{\alpha_k\mu_k}{\mu_k-z},\ \text{with } \sum_{k}\alpha_k=1$$

$$m=\sum_{k}\frac{\alpha_k}{\mu_k},\ \text{var}=2\sum_{k}\frac{\alpha_k}{\mu_k^2}-\left(\sum\frac{\alpha_k}{\mu_k}\right)^2\ c^2=\frac{2\sum\frac{\alpha_k}{\mu_k^2}}{\left[\sum\frac{\alpha_k}{\mu_k}\right]^2}-1$$

Gamma distribution

$$f(x)=e^{-x/\beta}\frac{x^{\alpha-1}}{\beta^{\alpha}\Gamma(\alpha)},\ \alpha,\beta>0,\ x\ge 0$$

$$m=\alpha\beta\,,\ Var=\alpha\beta^{2}$$

Weibull distribution

$$f(x)=\frac{\beta}{\delta}\left(\frac{x}{\delta}\right)^{\beta-1}e^{-\left(\frac{x}{\delta}\right)^{\beta}},\ F(x)=1-e^{-\left(\frac{x}{\delta}\right)^{\beta}}$$

$$m=\delta\ \Gamma(1+1/\beta),\ Var(x)=\delta^{2}\left[\Gamma(1+2/\beta)-\Gamma(1+1/\beta)^{2}\right]$$

Pareto distribution

$$f(x)=\frac{ab^{a}}{x^{a+1}},\ F(x)=\left(\frac{b}{x}\right)^{a},\ \text{with } a,b>0,\ x\ge b$$

$$m=\frac{ab}{a-1},$$

$$Var=\frac{ab^{2}}{(a-2)(a-1)^{2}}$$

5

Statistics

In formulating any statement about a "system", the engineer needs to acquire as precise as possible knowledge about it and about the environment in which it is to work. Such knowledge cannot result from pure reasoning only, and has to rely on observation and measurement. In the domain of performance evaluation, service times, as well as traffic offered, are examples of such input data. For reliability studies, equipment lifetime, system availability, etc. are measured. In a real network or during lab experiments, response delays or load levels can be observed. Similar measurements can be made in the preliminary design step, during simulation studies.

It results from this that a certain amount of data is to be collected on the system. The methods of *descriptive statistics* help in choosing the parameters of interest and in presenting in a synthetic way the set of results – in short, how to *visualise* them.

Now, exhaustive measurements are clearly impossible to carry out. The methods of *mathematical statistics* aim at providing tools to analyze data in order to extract all the possible information from them. For instance, *estimation theory* helps in estimating the confidence level to be associated with the prediction of a parameter of interest. At last, *hypothesis testing* helps in making decisions about the population under study, such as comparing two different samples or deciding the conformance of the measurements with a given theoretical distribution function.

Statistics are concerned with a set of elements, called the *population*. On each of the elements a *character* is observed, which varies from one element to another. Typical examples here are the duration of a communication, the length of a message, the number of busy elements in a pool of resources, the time between two failures of the equipment, etc. The implicit idea sustaining the statistical approach is that there exists a kind of regularity behind the apparent randomness of the observations, and that the population is characterized by a (unknown) well-defined value of the parameter, and that the observations are distributed around this value according to some probability law.

5.1. Descriptive statistics

Interpreting the huge amount of raw data collected during a campaign of measurements always happens to be difficult, and the first step towards their understanding is in visualizing them properly: how to summarize the information, how to stress the relations between them, etc. The methods of the descriptive statistics are of invaluable help in this task, the importance of which must not be underestimated. The detail of the analysis the specialist may conduct is only convincing for his/her peers. For the others, synthetic and convincing illustrations (charts, tables or other) must be a preferred communication media: *a picture paints a thousand words.*

5.1.1. *Data representation*

Numerous data have been gathered, and the point is to display them, so as to give a synthetic view, allowing one to visualise the whole results "in a glance", at least in a qualitative way. Numerous techniques have been conceived, which are easily available through modern specific software tools.

The most popular tool is the *histogram*. To begin with, assume that a discrete character is observed (to help in the presentation, it takes integer values 0, 1, 2, etc.). Let n_k be the frequency of value k, i.e. the number of times it is observed. Figure 5.1 shows the histogram.

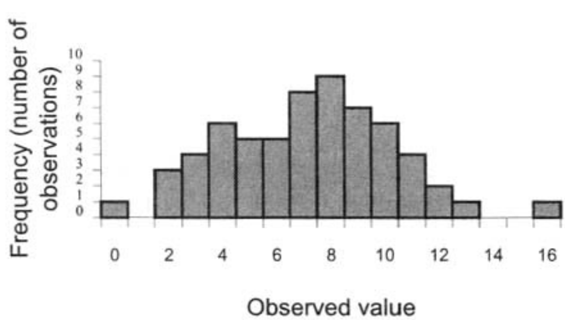


Figure 5.1. *Histogram, showing results of observations for a discrete variable*

The outcome has been "0" for 1 element, while 5 elements exhibit the value "6", and an average around "8" can be guessed. Abnormal measurements may be discovered in this way (here, perhaps "16", but the sample size is far too small for such a firm statement).

The case of continuous variables calls for a more careful analysis. The previous approach is no longer valid, as the outcomes are real numbers, scattered throughout the axis, with no two identical values. The method is to group the observations in classes and to draw *cumulated frequencies*. I intervals are defined: $[x_i, x_{i+1}[, i = 1,\dots,I$. They are usually referred to as *bins*. Class i bin gathers all observations within interval i, and contains n_i observations: n_i =# *Elts*(x, such that $x_i \le x < x_{i+1}$). With N for the size of the whole population, the relative frequency for class i is the ratio $f_i = n_i / N$. The curve of cumulated frequencies displays the whole set of measurements:

$$F(x) = \sum_{j=1}^{i} f_j \quad x_i \le x < x_{i+1}, \; et \; F(x_{i+1}) = \sum_{j=1}^{i+1} f_j. \tag{5-1}$$

The curve is made up of a series of horizontal segments (Figure 5.2):

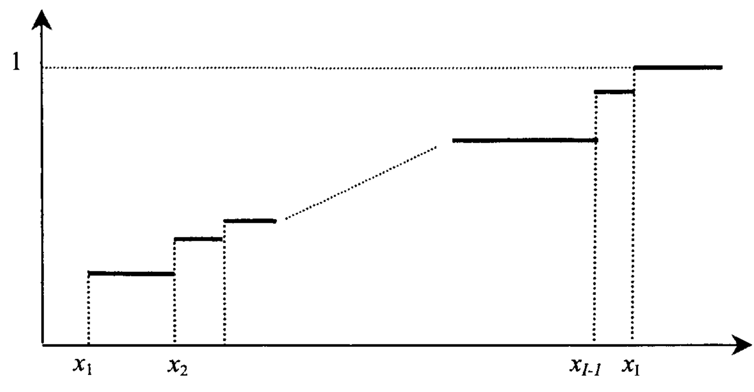


Figure 5.2. *The empirical distribution of a continuous variable*

The histogram is another representation, just as for discrete value observations, provided that the "*area rule*" is observed. For each interval i, a rectangle is constructed with a base length equal to its range $x_{i+1} - x_i$ and an area proportional to the number of observations within the interval. This guarantees the validity of the interpretation given to the graph:

Class i is represented on the histogram by a rectangle with area (instead of height) proportional to n_i. For bins of equal width, the distinction makes no sense. In the general case, the rule is motivated by the following arguments:
– It allows interpretation of the ordinate as an "empirical density".
– It allows an unbiased comparison between frequencies of the different classes to be made.
– It makes the histogram insensitive to class modifications (especially, the shape remains the same if several bins are grouped).

Other representations

Numerous other description tools have been imagined, the goal of which is to provide a synthetic, intuitive, access to the set of data. Various software packages make them easy to use. The reader is referred to the bibliography.

5.1.2. *Statistical parameters*

The direct visualization of the data is a first and important step. At that point, the need arises for a quantitative characterization for the parameter of interest. Here again, the assumption is that statistical laws, often difficult to identify, govern the phenomena under study. To begin with, one makes use of global parameters that introduce no specific assumption about the underlying probabilistic models.

5.1.2.1. *Fractiles*

Fractiles are read directly on the cumulated frequency curve. Let F be the cumulated frequency of variable X: $F(x)$ is the ratio of measured values equal to or less than x. The *α-fractile* (also referred to as *quantile* or *percentile*) is the number u such that $F(u) \le \alpha, \quad F(u+\varepsilon) > \alpha$.

This notion may help eliminate extreme values, most often abnormal. In the common situation where the measured character has no upper bound, the range of interest can be chosen as the interval between the 0.001-fractile and 0.999-fractile. In economics use is made of quartiles (values for α = 25, 50 or 75%). The 0.5-fractile is called *median*. It divides the whole population into two subsets of the same size.

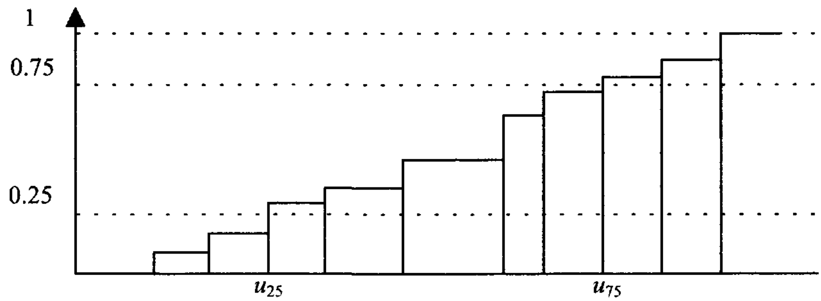


Figure 5.3. *Fractiles of a distribution function*

5.1.2.2. *The sample mean*

Among the global numeric indicators that are derived from the raw data, the mean value (or average) is certainly the most popular. The term *empirical mean* is sometimes used to emphasize the fact that this quantity is estimated, as opposed to the theoretical mean value of a probabilistic distribution. It is most commonly denoted as $\bar{x}$:

$$\bar{x} = \frac{\sum_{i=1}^{N} x_i}{N}, \tag{5-2}$$

N being the sample size.

5.1.2.3. *The sample variance*

In order to represent the dispersion (intuitively the distance from the average behaviour given by the sample mean), the sample variance V is introduced:

$$V = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2. \tag{5-3}$$

The standard deviation σ is the square root of the variance.

5.1.2.4. *The moments*

The k^{th}-moment of the distribution (moment of order k) is usually denoted as μ_k. Most often, the moments are taken about the mean:

$$\mu_k = \frac{\sum (x_i - \bar{x})^k}{N} \tag{5-4}$$

(the first moment is $\mu_1 = \bar{x}$).

5.1.2.5. *The mode*

The mode is the most frequently observed value. It is given by the maximum of the histogram. Distributions with a single mode are referred to as being *unimodal.*

Mode, median and mean values must be carefully distinguished. The mode is visible on the histogram; the median separates the population in two subsets of equal size, and the mean in two subsets of equal "weight". Unimodal distributions fit in one of the two categories:

- Mode < median < mean (the distribution is spread to the right);
- Mean < median < mode (the distribution is spread to the left).

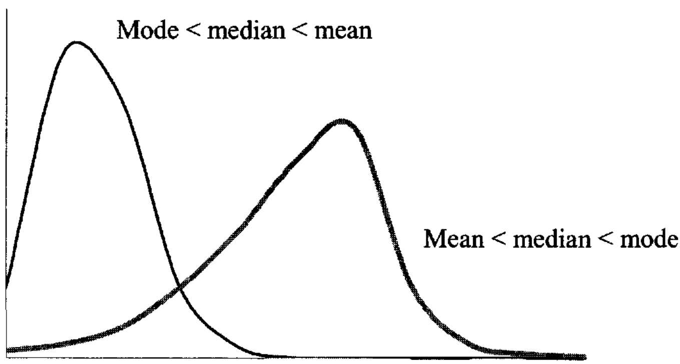


Figure 5.4. *Mean, mode, median of a density function*

5.1.2.6. *Other characterisations*

The mean and variance do not capture all features of the distribution. Other indicators have been proposed, which take account of certain classes of shapes of the distribution functions. These indicators are derived from the moments. Here too these indicators are to be seen as "empirical" (i.e. defined from the sample). In order to simplify the notations, the "–" are omitted in what follows.

$$\gamma_1 = \frac{\mu_3}{\sigma^3} \text{ or } \beta_1 = \gamma_1^2 .$$

These coefficients (respectively, Fisher and Pearson coefficient) reflect the *skewness* of the distribution. They take account of the *symmetry* the function may exhibit; they vanish when the curve is symmetric.

$$\beta_2 = \frac{\mu_4}{\mu_2^2},\ \gamma_2 = \beta_2 - 3 .$$

These coefficients allow comparison of the peakedness of the distribution (the "*kurtosis*") with the normal distribution, which is such that $\mu_3 = 0$, $\mu_4 = 3\sigma^4$ (see Chapter 4), and so $\gamma_1 = \gamma_2 = 0$.

5.2. Correlation and regression

When a phenomenon implies more than one random variable, the possible relationships between them are an important issue. Observing a sample of people, one can expect a kind of "fuzzy" relation between their size and weight. In a similar way, the socio-economic category of a family, as given by its annual income, is related to its telephone or internet usage (and more precisely with the level of usage, i.e. the *traffic*).

The same observation is made in reliability studies. For instance when characterizing some production process one is interested in the possible relation between the rejection ratio and some condition of the production (e.g. the date, or identity of the production line). A first approach is to illustrate the possible relation by displaying the set of all pairs (x, y) of observed data.

The "Jipp curve" (see Figure 5.5) is a classical example of such a *correlation*. It shows the relation between the revenue per inhabitant and the level of equipment for fixed telephony. There is no strict mathematical link, but only a quite typical trend. Interpreting this trend is beyond the scope of the statistical tools. There may be a direct causal relationship (phenomenon X provokes Y) or a more complex scheme (X and Y being two consequences of a third, hidden, cause).

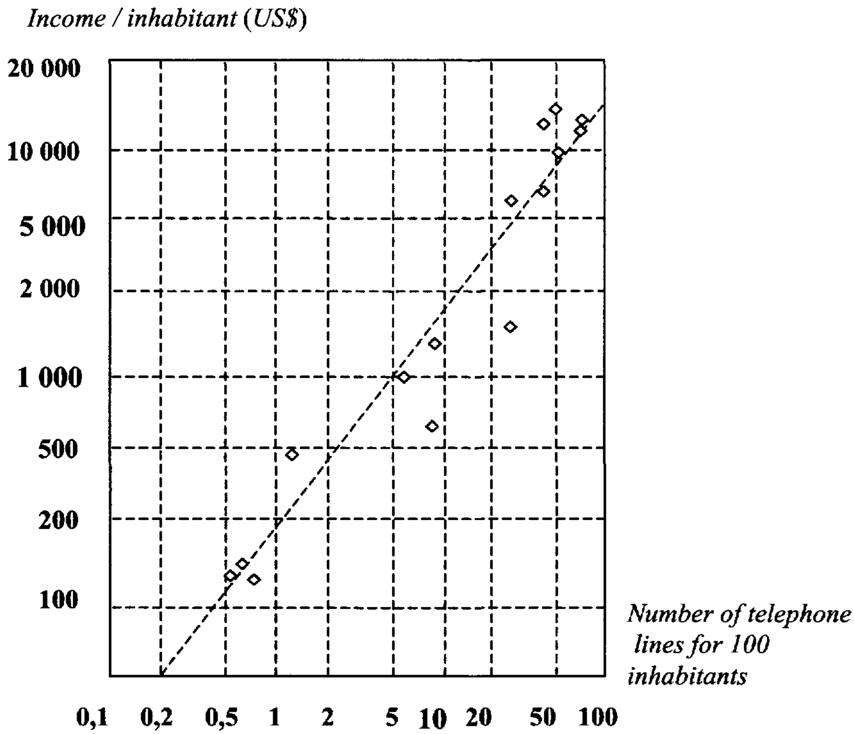


Figure 5.5. *Relation between income and equipment level for the telephone service, for various countries*

5.2.1. *Correlation coefficient*

The relationship between two random variables is numerically represented by the correlation coefficient. This is a number, between −1 and 1, defined as:

$$\hat{\rho}=\frac{\sum(X_i-\overline{X})(Y_i-\overline{Y})}{\sqrt{\sum\left(X_i-\overline{X}\right)^2}\cdot\sqrt{\sum\left(Y_i-\overline{Y}\right)^2}}. \tag{5-5}$$

The reader may compare this with the covariance (equation 3-53): the above quantity is nothing more than the "empirical" version (normalised). When the coefficient is equal to 0, the variables are said to be independent.

5.2.2. *The regression curve*

When two variables X and Y are correlated, their mutual influence may be visualized through a causal relation $Y = f(X)$, although however that correlation *does not imply* any direct causal relation. This operation is called a *regression*. In the general case the relation could be quite complex, in the following the analysis is restricted to a linear model $Y = aX + b$. Note that one can always restrict to a linear model by the appropriate transformation of the variables. It is also possible to define more elaborated models (e.g. quadratic relations).

The least squares method

Let us assume a linear model. The coefficients have to be estimated from the set of measurements providing N couples (x_i, y_i). As the linear model does not capture the whole phenomenon, since other factors are neglected, the linear relation is approximate and the "error" is represented by an additional variable e:

$$y_i = \hat{a}x_i + \hat{b} + e_i\,, \tag{5-6}$$

(we are really looking for *estimators* of the parameters; as we will see later, they are often symbolized using "^"). The method consists in allocating values to parameters so as to minimize the distance between the model and the values actually observed. The distance is written:

$$\Delta = \sum \left(y_i - \hat{a}x_i - \hat{b}\right)^2 .$$

The optimal coefficients are such that they cancel partial derivatives:

$$\frac{\partial \Delta}{\partial \hat{a}} = 0 \text{ i.e. } \sum x_i(y_i - \hat{a}x_i - \hat{b}) = 0\text{, and } \sum x_i y_i - N\hat{a}\overline{x^2} - N\hat{b}\overline{x} = 0\,,$$

$$\frac{\partial \Delta}{\partial \hat{b}} = 0 \text{ i.e. } \sum (y_i - \hat{a}x_i - \hat{b}) = 0\text{, and } N\overline{y} - N\hat{a}\overline{x} - N\hat{b} = 0\,.$$

This yields the solution we are looking for:

$$\hat{a} = \frac{\frac{\sum x_i y_i}{N} - \overline{x}.\overline{y}}{\overline{x^2} - \overline{x}^2} = \hat{\rho}\frac{S_y}{S_x}. \tag{5-7}$$

$$\hat{b} = \overline{y} - \hat{a}\overline{x}$$

(S_x, S_y stand for the empirical standard deviations of the variables). The slope of the line is directly related with the regression coefficient. The variance of the e_i's gives a measure of the residual error of the model:

$$S_e^2 = \frac{1}{N-2}\left(\overline{y^2} - \overline{y}^2 - \hat{a}^2\left(\overline{x^2} - \overline{x}^2\right)\right) \tag{5-8}$$

The coefficient $N - 2$ ensures that the estimator is unbiased (this concept will be clarified later on).

NOTE– USING SPECIFIC SCALES. The Jipp curve is drawn with a logarithmic scale for the ordinate axis. In the general case of a non linear relation between two quantities, the graphical representation of the cloud of measures can be made using a specific scale, thanks to which a linear trend reappears.

This is especially the case with the celebrated "gausso-arithmetic" scale, where the scale is such that any normal distribution appears as a straight line. For an empirical distribution, this gives a simple visual method to estimate the accuracy of a normal approximation and to give a rough idea of the moments: the mean is the abscissa corresponding to 50%, the standard deviation is half the distance between 16 and 84%, or the reverse of the slope. The graph (see Figure 5.6) is referred to as the *Normal Plot*.

5.3. Sampling and estimation techniques

Visualizing data, apart from giving them a synthetic characterization through global parameters, covers by no means the needs of the analyst. Especially, the main issue is to predict general properties of the whole population from data coming from samples of limited size, for economical or technical reasons (and most often because the population is of infinite size). The question is thus: how to estimate the value of the (true) parameters of the population from the incomplete series of measurements, i.e. the *sample*. A basic assumption is needed here, which allows use of the probability theory and its tools: each element being tested has to be chosen *independently* of the previous ones (called a *random* drawing). Most often the assumption is impossible to assert.

Drawing random variables is actually an important aspect of our activity, for reliability studies, traffic measurements or simulation. We address this point in more detail in Chapter 8.

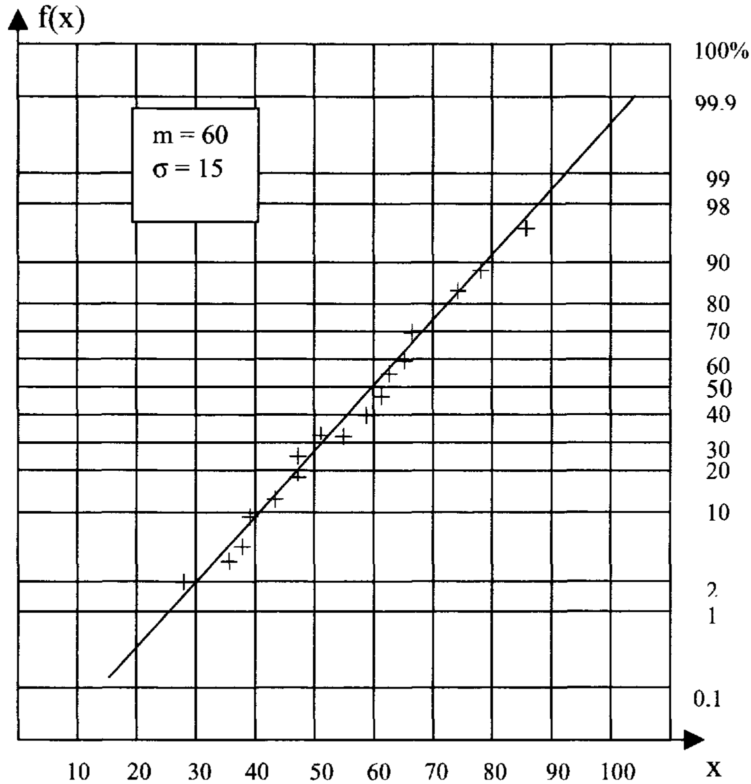


Figure 5.6. *Normal plot showing that the empirical measures are roughly normally distributed*

Two basic results of *mathematical statistics* help in understanding the classical techniques related with sampling.

a) Sample mean, and variance: Consider a sample of size *N*, drawn from population of infinite size, or from a finite size population with replacements (then the draw is said *non exhaustive*) where the character of interest has average *m* and variance σ^2. The sample mean $\overline{m}$ given by Equation 5-2 is a random variable, with average *m*, and with variance $\sigma^2(\overline{m})$ equal to σ^2/N. Symbolically,

$$E(\overline{m}) = m, \tag{5-9}$$

$$\sigma(\overline{m}) = \frac{\sigma}{\sqrt{N}}, \tag{5-10}$$

or, $E(\overline{m} - m)^2 = \sigma^2(\overline{m}) = \dfrac{\sigma^2}{N}$.

This last result is easy to verify: let the sample contain N elements X_i. The classical results concerning the sum of independent random variables (see Chapter 3) provide:

$$\overline{m} = \frac{1}{N}(X_1 + X_2 + ... + X_N),$$

and thus:

$$E(\overline{m}) = \frac{1}{N}\left[E(X_1) + E(X_2) + ... + E(X_N)\right] = \frac{1}{N} Nm = m.$$

In the same way:

$$\sigma^2(\overline{m}) = Var\left[\frac{1}{N}(X_1 + ... + X_N\right] = Var\left[\frac{X_1}{N} + ... + \frac{X_N}{N}\right] =$$

$$\frac{1}{N^2}\left[Var(X_1) + ... + Var(X_N)\right] = \frac{N}{N^2} Var(X) = \frac{\sigma^2}{N}.$$

b) Central limit theorem.

The previous result concerns the two first moments but not the probability distribution of the sample mean. The next result concerns the *distribution* of the sample mean.

The central limit theorem is of major importance when using statistical methods, as it introduces and motivates the endless references to the normal distribution. Also, it justifies most of the approximations presented hereafter. Its proof goes beyond the scope of the present book.

THEOREM.– Consider a set of N variables $X_1, X_2, \ldots X_N$, independent and distributed according to a given, arbitrary distribution with average μ and finite variance σ^2. Then, the variable:

$$\overline{X} = \frac{1}{N}\sum_{i=1}^{N} X_i$$

goes, as N increases, to a limiting distribution which is *normal*, with average μ and with variance σ^2 / N.

This is a major improvement, as compared with the previous result: drawing a sample of size N, from a population with a character distributed according to an *arbitrary* distribution with mean m and standard-deviation σ, the sample mean goes to a normal law with mean m and standard-deviation $\sigma / \sqrt{N}$, as the sample size increases.

Practically, the approximation is good when N is greater than 30 in most configurations. However the independence assumption is mandatory.

These two important results are the basis on which the remainder of this section is built. Two major techniques allow extracting information from the sample: *estimation* and *hypothesis testing*. Estimating the value of a parameter, based upon the assumption of an underlying probabilistic model, may not be enough to support a decision such as "accept/reject". One can also need to justify the initial assumptions. The tools of hypothesis testing give answers to these issues.

At last, observations or tests (e.g. during simulations) are of limited duration, the question thus arises of taking account of the last time interval, or of the last test. This leads to distinguish between *truncated tests*, for which the duration is chosen independently of the number of events observed, and *curtailed tests*, in which the total number of events is fixed in advance, the duration then having no limitation.

5.4. Estimation

The problem can be stated as follows:

Given a population, with a given character distributed according to a probability distribution function depending on some parameter p, estimate the numerical value of the parameter from the observations made on a sample of limited size.

There are two classes of techniques related to estimation: *point estimation* and *interval estimation.* In *point estimation* the goal is to give the "best possible" estimate of the parameter, assuming the form of the probability distribution is known.

Since the sample is of limited size, the estimation is likely to be accompanied by some error. This introduces the notion of *confidence* in the result and the result will be stated under the form of a *confidence interval* inside of which is the true value, with a given probability: this is *interval estimation*, the answer of which is stated:

With a *confidence level* $1-\alpha$ (i.e. a risk of error α), the parameter p lies in the interval $[p_{\min}, p_{\max}]$.

Clearly, the narrower the confidence interval, the higher the risk of error. This is the fundamental dilemma of every sampling and estimation activity.

5.4.1. *Point estimation*

Assume the character is distributed according to some probability distribution function, the form of which is known, except for a certain parameter p, which one intends to estimate from the data.

Let the sample of size n be denoted $(x_1, x_2,..., x_n)$.The estimator is usually denoted as "$\hat{p}$", to stress the fact that it is actually a random variable, and to distinguish it from the true value. The properties desirable for the estimator are:

- As the sample size increases, the estimated value ought to get closer to the true value: $Lim_{n\to\infty}(\hat{p}_n) = p$ (where $\hat{p}_n$ stands for the estimate built on a sample of size n): the estimator is said to be *convergent (or consistent).*
- The estimator ought to be *unbiased*, that is: for all n, $E(\hat{p}) = p$ (strictly speaking, one should put the same requirement on higher moments).

An estimator which is convergent and unbiased is said to be *absolutely correct.*

Several methods have been proposed to derive a point estimator. In the following we concentrate on the method of *maximum likelihood.* Let $f(x_1, x_2,...x_n,)$ – or $f(x_i)$ for short – denote the probability density of the sample statistics (i.e. the probability to draw x_1, x_2,...x_n, in any order). The maximum likelihood estimator is the one which gives the higher value for $f(x_1, x_2,...x_n,)$, i.e. such that

$$\frac{dLog(f(x_1,x_2,..x_n)}{dp} = 0 \text{ (likelihood equation)} \tag{5-11}$$

It can be shown that such an estimator is convergent. However, the method does not guarantee it is unbiased. The estimation will be said to be *correct* but not absolutely correct. In order to make it absolutely correct the estimator will be adjusted, as is shown hereafter.

The two following results are of main importance for estimation:

Average

The sample mean is an absolutely correct estimator for the average (mean):

$$\hat{m} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{5-12}$$

It is easy to verify that, whatever the value for n, $E(\hat{m}) = m$: the estimator is unbiased.

Variance

The quantity s^2 is an absolutely correct estimator for the variance:

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \hat{m})^2 \tag{5-13}$$

s^2 is known as the "empirical variance". Actually, the term $(n-1)$ instead of n takes account of the bias which would be observed with the sample variance:

$$\hat{\sigma}^2 = \frac{\sum (x_i - \hat{m})^2}{n}, \text{ since } E(\hat{\sigma}^2) = \frac{n-1}{n}\sigma^2 .$$

It is thus a biased estimator. In order to correct the bias, statisticians introduce the substitution:

$$s^2 = \frac{n}{n-1}\hat{\sigma}^2 .$$

Let us verify this result:

$$(n-1)s^2 = \sum (x_i - \hat{m})^2 = \sum \left[(x_i - m) - (\hat{m} - m)\right]^2.$$

Developing and taking the mean of the right hand side expression, one gets, using the results (5-9) and (5-10) about the mean and the variance of a sample:

$$E\sum (x_i - m)^2 = \sum E(x_i - m)^2 = n\sigma^2,$$

$$2E\sum (x_i - m)(\hat{m} - m) = 2E\left[(\hat{m} - m)\sum (x_i - m)\right] =$$

$$2E\left[(\hat{m} - m)(n\hat{m} - nm)\right] = \frac{2}{n}E(n\hat{m} - nm)^2 = 2\sigma^2,$$

$$E\sum (\hat{m} - m)^2 = E\left[n(\hat{m} - m)^2\right] = nE(\hat{m} - m)^2 = \sigma^2,$$

and at last:

$$E(n-1)s^2 = n\sigma^2 - 2\sigma^2 + \sigma^2 = (n-1)\sigma^2, \text{ thus } E(s^2) = \sigma^2.$$

Note that these results do not depend on any assumption about the distribution probability of the parameter under study.

Now, one can show that, for the case of normal variables, these estimators are actually the best possible.

Estimating the mean and variance of a normal distribution

a) The problem is to estimate the mean value m of a normal distribution with parameters (m, σ), where σ is known. The method of the maximum likelihood is used to derive the best estimator:

$$f(x_i) = \frac{1}{(\sigma\sqrt{2\pi})^n} e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - m)^2}$$

From this expression, one derives:

$$\frac{dLog(f)}{dm} = \frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i - m).$$

And for the maximum likelihood:

$$\left(\frac{dLog(f)}{dm}\right)_{m=\hat{m}} = 0 .$$

Thus $\hat{m} = \frac{1}{n}\sum_{i=1}^{n} x_i$.

One can verify that $E(\hat{m}) = m$ whatever n. The estimator is unbiased and convergent thus absolutely correct.

b) Similarly, when m is known, the best estimator for the variance is:

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - m)^2 .$$

c) In the general case, however, both m and σ are unknown.

The mean is estimated as above, but the estimator of the variance $\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{m})^2$ is biased, although convergent. Instead, one makes use of the parameter already introduced $s^2 = \frac{n}{n-1}\hat{\sigma}^2$.

Same results hold for the exponential distribution as considered in the following example.

Example: estimating the average lifetime of an equipment

As an example of application, we show how to apply these results to estimate the average lifetime of equipment, the mortality having an exponential distribution (see Chapter 6). The same reasoning holds for other observations.

Consider a trial on a sample of n elements. The average lifetime θ is estimated by:

$$\hat{\theta} = \frac{1}{n}\sum_{i=1}^{n} t_i = \frac{T}{n} , \tag{5-14}$$

where t_i is the lifetime of item i, and T is the total duration of the test. For practical purposes, the duration of the trial, as well as the number of failures, is to be limited. This refers to the notions of truncated tests and curtailed tests. For truncated tests, the test duration is limited to t_{max}. If, upon the n elements under test, $n-k$ are still operating at time t_{max}, then:

$$\hat{\theta} = \frac{1}{k}\left(t_1 + t_2 + \ldots t_k + (n-k)t_{\max}\right) = \frac{T}{k}, \tag{5-15}$$

where T is the cumulated duration of the test.

If no failure at all has been detected (so that $\hat{\theta} = 0$), it is mandatory to estimate a confidence interval. The number of elements to test is estimated using the simple formula:

$$k = n\left(1 - \exp(-\frac{t_{\max}}{\theta})\right)$$

As an example, estimating a failure rate around $\lambda = 1/\theta = 10^{-6}$/h, by observing 30 failures during 8760 hours (one year) requires 3450 elements to be tested. This clearly stresses the difficulty of characterising elements having high reliability – or, otherwise stated, in considering the case of rare events.

For curtailed tests, the test is stopped upon failure number k. Then:

$$\hat{\theta} = \frac{1}{k}\left(t_1 + t_2 + \ldots t_k + (n-k)t_k\right) = \frac{T}{k}, \tag{5-16}$$

where T is the cumulated duration of the test at failure k.

NOTE. The maximum likelihood is by no means the only approach when looking for estimators. For instance, the *least squares* method provides interesting estimators. In fact, in most cases these methods lead to the same estimator. The section about correlation has already shown an example illustrating the principles of the least squares approach.

5.4.2. *Estimating confidence intervals*

The maximum likelihood method provides a "good" way to estimate an unknown parameter, when the underlying probability distribution is known. However, if the same trial is repeated, or lengthened, it will give a different result. As already mentioned, *the estimator is a random variable*, as opposed to the parameter to be estimated, assumed to have a constant value. This leads one to question the "precision" of the prediction. Not only has a value be given for the parameter p, but a range within which its actual value is likely to be found. As above, the estimator of p is denoted as $\hat{p}$.

Let H be the probability distribution of the estimator. Let us choose two real numbers α_1 and α_2 ($\alpha = \alpha_1 + \alpha_2$). These numbers stand for the *risk* in the prediction. More precisely, they provide upper and lower bounds for the estimation, i.e. a *confidence interval I* such that:

$$P(\hat{p} \in I) = 1 - \alpha = 1 - (\alpha_1 + \alpha_2)$$

The approach consists, from the distribution H (or the density h associated), in finding two quantities u_1, u_2, such that $P(\hat{p} < u_1) = \alpha_1$ and $P(\hat{p} > u_2) = \alpha_2$ (see Figure 5.7). Any choice for the risk is possible (for instance $\alpha_1 = \alpha_2 = \alpha/2$, or $\alpha_1 = 0$, etc.).

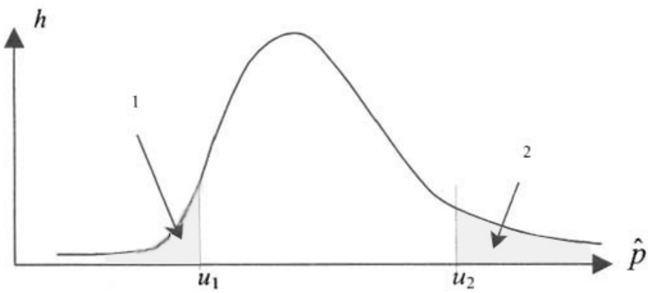


Figure 5.7. *Risks associated with the bounds of the confidence interval*

The exact shape of the curve depends on the unknown value p, and so do u_1, u_2: $u_1 = H_1(p)$, $u_2 = H_2(p)$.

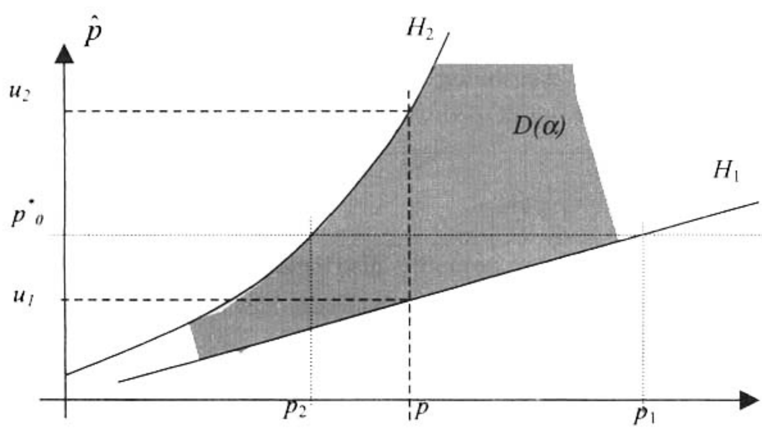


Figure 5.8. *Limits of the confidence interval*

The curves H_1, H_2 delimit an area in the plane denoted as $D(\alpha)$ (it depends clearly on the risk, written simply α for short). Let us choose an arbitrary point and the ordinate axis p_0^*. The ordinate cuts the curves at the abscissas p_1 and p_2, given by the inverse functions $p_2 = H_2^{-1}(p_0^*), p_1 = H_1^{-1}(p_0^*)$. These functions are most often impossible to derive explicitly, but can be obtained numerically.

Now, let us consider the three following relations:

$$(p, \hat{p}) \subset D(\alpha)$$
$$u_1 < \hat{p} < u_2$$
$$p_1(\hat{p}, \alpha) < p < p_2(\hat{p}, \alpha)$$

All these relations describe the same set D. The elements have thus the same probability:

$$P(p_1 < \hat{p} < p_2) = 1 - (\alpha_1 + \alpha_2) = 1 - \alpha$$

The estimation proceeds by executing the trial, giving a series of measurements $(x_1, x_2, ..., x_n)$, from which the estimate $\hat{p}$ is derived, and also the bounds (p_1, p_2), through the set of curves obtained above.

The examples below will help in the understanding of the principle of the method and will illustrate it on typical cases.

Example: estimating the mean of a normal distribution

A random variable is distributed according to a normal distribution, with a known variance. The point estimator for the mean is, as explained in a previous section,

$$\hat{m} = \sum_{i=1}^{n} x_i .$$

Moreover, the estimator is normally distributed, with mean m and variance σ^2 / n (remember that the sum of normal variables has a normal distribution). So, the parameter " u ", such that

$$u = \frac{|\hat{m} - m|}{\sigma / \sqrt{n}},$$

is distributed according to the normalized normal distribution. Thus, given the risks α_1 and α_2:

$$P\left(-u_1 < u = \frac{\hat{m} - m}{\sigma / \sqrt{n}} < u_2\right) = 1 - \alpha\,, \tag{5-17}$$

and finally:

$$P\left(\hat{m} - u_1 \frac{\sigma}{\sqrt{n}} < m < \hat{m} + u_2 \frac{\sigma}{\sqrt{n}}\right) = 1 - \alpha\,. \tag{5-18}$$

For example, with a risk symmetrically distributed: $\alpha_1 = \alpha_2 = 0.025$, one gets $u_1 = -u_2 = 1.96$, i.e. an interval around $\pm 2\sigma / \sqrt{n}$.

Note that in the case where the variable is not normally distributed, the method can still be used, as the central limit theorem ensures that the estimator behaves approximately as a normal variable, with mean m and variance σ^2 / n. In the general case, however, σ is unknown, and has to be estimated too.

Example: The Chi-2 distribution in reliability

We revisit the problem of estimating the lifetime of items having a lifetime exponentially distributed. We assume that the number of failures within an interval obeys the Poisson distribution. This corresponds to a truncated test, in which a failed item is immediately replaced.

Let θ be the average lifetime, let n be the number of items under test, and $t_{\max}$ the test duration. With the Poisson distribution, the probability of having more than r failures during the interval θ is:

$P(i \le r;\theta) = \sum_{i=0}^{r} \frac{(k)^i}{i!} e^{-k}$, with $k = T/\theta$, and $T = n\ t_{max}$, the cumulated test duration. (the notation r is motivated by the usage for the Chi-2 distribution, that we are to obtain).

Now, the following relation holds:

$$P(i \le r;\theta) = \sum_{i=0}^{r} \frac{(k)^i}{i!} e^{-k} = e^{-k}(1 + k + \ldots \frac{k^r}{r!}) = \int_k^{\infty} \frac{t^r}{r!} e^{-t} dt$$

And thus, with $t = u/2$:

$$P(i \le r;\theta) = \int_{2k}^{\infty} \frac{u^r}{r!2^{r+1}} e^{-u/2} du$$

Referring to Chapter 3, one recognizes the Chi-2 distribution:

Let $u = \chi^2$ and $\nu = 2r + 2$, one gets:

$$P(i \le r;\theta) = \int_{2k}^{\infty} \frac{1}{2^{\nu/2}\Gamma(\nu/2)} (\chi^2)^{\frac{\nu}{2}-1} e^{-\chi^2/2} d\chi^2 \tag{5-19}$$

This is the probability that a χ^2 with $2(r+1)$ degrees of freedom is larger than $2k$.

Finally,

$$P(i \le r;\theta) = P(\chi^2_{2r+2} > 2k). \tag{5-20}$$

This yields the upper and lower bounds at the levels α_1 and α_2:

$$\frac{2T}{\chi^2_{2r+2;\alpha_1}} < \theta < \frac{2T}{\chi^2_{2r;1-\alpha_2}}. \tag{5-21}$$

Especially, for a symmetrical risk $\alpha_1 = \alpha_2 = \alpha/2$:

$$\frac{2T}{\chi^2_{2r+2;\alpha/2}} < \theta < \frac{2T}{\chi^2_{2r;1-\alpha/2}}. \tag{5-22}$$

Table 5.1 summarizes the main results for this important kind of tests (including the case of curtailed tests).

An example of application to realistic situations will be developed in Chapter 9. Tables of the Chi-2 distribution are also given in Appendix 2.

Estimation of a proportion

Suppose one intends to estimate the proportion p of faulty elements in a given population of infinite size. The experiment proceeds by choosing a sample of size N, the successive drawings being independent of each other, and by counting the number k of faulty items. In accordance with the intuition, the calculation confirms that the maximum likelihood estimator for p is the ratio $\hat{p} = k/N$.

Table 5.1. *Truncated and curtailed test: confidence intervals with the Chi-2 distribution*

	Truncated test at t, N equipments, r failures		Curtailed test at t_r, r failures, N equipments	
Cumulated time T	With replacement $T = Nt$	Without replacement $T = t_1 + t_2 \ldots + (N-r)t$	With replacement $T = Nt_r$	Without replacement $T = t_1 + t_2 \ldots + (N-r+1)t_r$
Point estimator	$\theta = T/r$ (*If* $r = 0$, *take* $\theta = 3T$)		$\theta = T/r$ (*If* $r = 0$, *take* $\theta = 3T$)	
Lower limit at confidence level $(1-\alpha)$	$\theta_i = \dfrac{2T}{\chi^2_{2r+2;1-\alpha}}$		$\theta_i = \dfrac{2T}{\chi^2_{2r;1-\alpha}}$	
Confidence interval at confidence level $(1-\alpha)$	$\dfrac{2T}{\chi^2_{2r+2;\alpha/2}} < \theta < \dfrac{2T}{\chi^2_{2r;1-\alpha/2}}$		$\dfrac{2T}{\chi^2_{2r;\alpha/2}} < \theta < \dfrac{2T}{\chi^2_{2r;1-\alpha/2}}$	

Under the independence assumption, the number of faulty items is distributed according to a Bernoulli distribution (for a population of finite size, the distribution would be hypergeometric). The probability distribution is written:

$$P(k \text{ faulty among } N) = \binom{N}{k} p^k (1-p)^{N-k}. \tag{5-23}$$

The probability of observing less than k faulty elements is:

$$P(k,j) = \sum_{j=0}^{k} \binom{N}{j} p^j (1-p)^{N-j}$$

Once the risk α_1 is chosen, the curve H_1 is obtained by "inverting" the relation (which is possible through exact numerical calculation, or using an approximate method, see below).

In most cases, the experiment is such that the normal approximation holds: this is the case, if $p > 5$, and $Np > 5$ (or $p < 0.5$, and $N(1-p) > 5$). The binomial distribution is replaced by the normal law with mean $m = Np$ and variance $Np(1-p)$, for which the mean is to be estimated. If the risk is taken symmetrically, it amounts to calculate $u(\alpha)$ and the confidence interval is:

$$\hat{p} - u_\alpha \sqrt{\frac{\hat{p}(1-\hat{p})}{N}} \le p \le \hat{p} + u_\alpha \sqrt{\frac{\hat{p}(1-\hat{p})}{N}} . \qquad (5\text{-}24)$$

If the conditions for using the normal approximation are not fulfilled, the Poisson distribution is used instead: it has been shown that the binomial distribution goes to the Poisson distribution with parameter Np, as soon as $N \ge 40$, and $p < 0.1$ (see Chapter 4).

How to build the curves H1 and H2

The solution can only be derived numerically. In the case where the normal approximation holds, the confidence interval is given by relation (5-24). The limits H_1 and H_2 are the points such that:

$$\hat{p} - u(\alpha)\sqrt{\frac{\hat{p}(1-\hat{p})}{N}} = p;\; p = \hat{p} + u(\alpha)\sqrt{\frac{\hat{p}(1-\hat{p})}{N}},$$

that is the solutions of the quadratic equation:

$$(p-\hat{p})^2 = u^2 \frac{\hat{p}(1-\hat{p})}{N}.$$

Figure 5.9 displays the confidence interval versus the observed value, for a confidence level 95%.

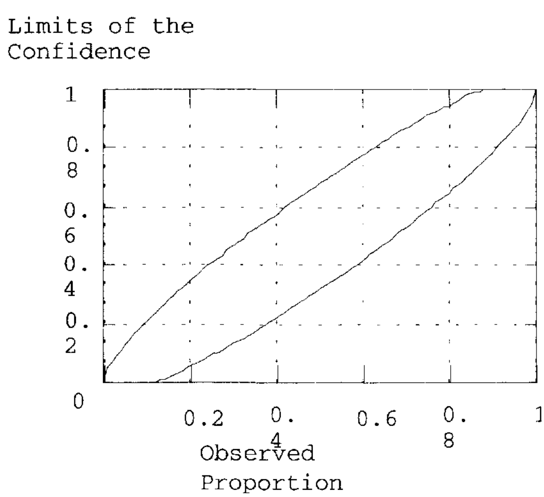


Figure 5.9. *Limits of the confidence interval for estimating a proportion*

Estimating the parameter of a Poisson distribution

Here, the problem is to estimate the parameter λ of the Poisson distribution, given by:

$$P(k) = \frac{\lambda^k}{k!} e^{-\lambda}. \tag{5-25}$$

The typical example can be in the observation of faulty elements, in the configuration of low rejection ratios, so that the Poisson distribution is a good approximation. Also, this is the distribution corresponding to the observation of the number of arrivals within a window of length T for a Poisson traffic with rate a: here, $\lambda = aT$.

The experiment consists in repeating the counting (observing m windows of size T, e.g., providing the values $k_1, k_2, ... k_m$). The maximum likelihood estimator is given by:

$$\hat{\lambda} = \frac{k_1 + k_2 + ... + k_m}{m}. \tag{5-26}$$

The function H^{-1} can be derived numerically, or by using tables.

5.5. Hypothesis testing

The goal of hypothesis testing is to help in deciding if an hypothesis made on a population can be retained, based upon the available observations. Depending on the nature of the hypothesis, and especially if it can be formulated under a qualitative form, the *parametric* or *non-parametric* methods are used. For instance, testing that an arrival process obeys the Poisson assumption or that the service durations are exponentially distributed is a non-parametric test. On the other hand, testing that they conform to the Poisson distribution with parameter m and standard deviation σ, is a parametric test.

5.5.1. *Example: testing the value of the mean of a normal distribution*

This simple example provides a good introduction to the method. We have a population with a parameter X normally distributed with parameters m and σ. The standard deviation σ is known, and the test is related with m. For instance, the hypothesis is "m larger than 10".

One draws a sample of size *N*, from which the sample mean $\overline{X}$ is estimated. Remembering that the sample mean is normally distributed, with mean *m* and standard deviation $\sigma/\sqrt{N}$, we can write, see (5.24):

$$P\left\{\overline{X} < m - u_1 \frac{\sigma}{\sqrt{N}}\right\} = \alpha_1 \text{ and } P\left\{\overline{X} > m + u_2 \frac{\sigma}{\sqrt{N}}\right\} = \alpha_2 . \tag{5-27}$$

Assume for instance $\alpha = 0.01$, i.e. $x_\alpha \approx -2.32$. Assume the sample has size *N* = 100, and the standard deviation is $\sigma = 5$. The above result gives: $P\{\overline{X} < 10 - 1.16 = 8.84\} = 0.01$.

Imagine first that the experiment results in the value $\overline{X} = 12$. This value seems coherent with the hypothesis. Now, imagine that the result is $\overline{X} = 7.5$. The probability of observing such a value is low, if the hypothesis is true: this leads to *rejection* of the hypothesis. However, even with $m > 10$ this value has a positive probability to be observed, meaning that the rejection is accompanied with a risk.

The general scheme of the test is as follows:

Let *X* be the variable under test. One is given a first hypothesis H_0, e.g. $X = X_0$, or $X < X_0$, etc. As the experiment is of statistical nature, the possibility of an error cannot be eliminated. A first risk α is thus introduced, the "*error of type I*". This is the probability to reject H_0 although it is exact.

But there is a risk of accepting the hypothesis while it is false. This introduces the "*error of type II*", β. More precisely, we introduce a second hypothesis H_1, and the test is H_0 versus H_1. The error of type II is the possibility to reject H_1 although it is true. The following table summarizes the procedure.

Table 5.2. *The procedure of the hypothesis test*

"Truth" Decision	**H_0 true**	**H_1 true**
Accept H_0	Correct decision	Type II error
Reject H_0 and accept H_1	Type I error	Correct decision

The "power function" of the test

The principle is as follows: given a population in which a character is subject to the test. The population is conforming if the proportion of values greater than some limit is lower than some probability P_{max}. Typical situations are a response time lower than T_{max} for 95% of the calls. The power function gives the probability to accept the test, given the sample of size N, for a population having a given parameter p. Figure 5.10 displays the general shape of the power curve.

In the case of an exhaustive sampling, the test would yield a result with certainty:

– probability of acceptance = 1 if $p > P_{\min}$;

– probability of acceptance = 0 if $p < P_{\min}$.

This is symbolized by the theoretical curve. Now, the experiment introduces both a risk of erroneous acceptance β, and a risk of erroneous rejection α. This corresponds to the effective power function.

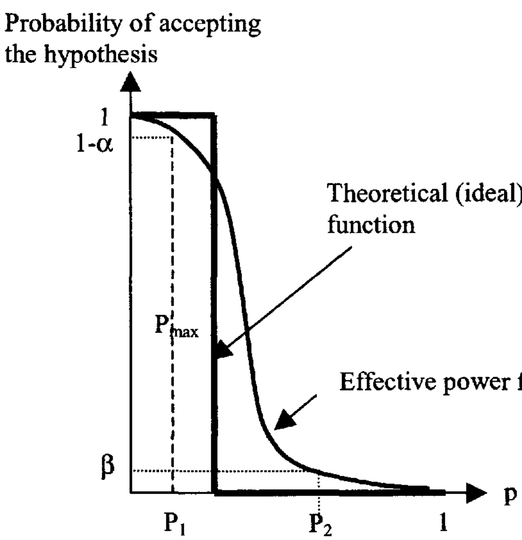


Figure 5.10. *The power function of the test*

To summarise: a customer encounters a risk (said of type II) of making a wrong decision in accepting the product under test, although its actual ratio p of bad elements is higher than the threshold P_2 that has been agreed. Similarly, the provider encounters a risk (of type I) that the product is rejected, although the actual ratio is lower than the limit P_1. Statisticians sometimes refer to these risks respectively as the *buyer risk* and the *provider risk.* In telecommunications, it would typically describe the relations between a network operator and an equipment manufacturer.

5.5.2. *Chi-2 (Chi square) test: uniformity of a random generator*

A discrete-event simulation makes use of random numbers, uniformly distributed in the interval (0, 1) provided by a *pseudo-random* generator, see Chapter 8 for more details.

The problem arises of testing the quality of the generator, and more especially its uniformity. The Chi-2 test brings an answer to this problem. The interval (0, 1) is divided in n sub-intervals of equal length (e.g., n=10, each interval having a length 0.1). N numbers are drawn and are positioned in the intervals (only for a histogram). If the generator conforms to the uniformity hypothesis, the probability that a number falls within any interval is simply $1/n$, so that an average of N/n elements are expected in each bin. The hypothesis is "the generator is uniform".

If this is true, the number of drawings in each bin conforms to the Bernoulli distribution:

$$P(k)=\binom{N}{k}p^k(1-p)^{N-k}, \text{ with } p=\frac{1}{n}. \tag{5-28}$$

The mean is Np, the variance $Np(1-p)$. Under the usual assumptions, the normal distribution is used as a convenient approximation. Let k_j be the number of drawings in the interval j, then:

$\dfrac{k_j-Np}{\sqrt{Np(1-p)}}=\dfrac{k_j-N/n}{\sqrt{\dfrac{N}{n}\left(1-\dfrac{1}{n}\right)}}$ is distributed according to the standard normal distribution. Thus the variable:

$$Z=\sum_j\frac{(k_j-N/n)^2}{\frac{N}{n}\left(1-\frac{1}{n}\right)}\cong\frac{\sum_j(k_j-N/n)^2}{N/n} \tag{5-29}$$

is distributed as a Chi-2 with $n-1$ degrees of freedom. A risk α is adopted (for instance 5%), the test is re-stated as $Z < u_\alpha$. For instance, with $\alpha = 5\,\%$ and $n = 10$, the Chi-2 with 9 degrees of freedom gives 16.92 as the limit: the hypothesis is accepted as long as Z is below 16.92.

5.5.3. *Test of the correlation*

The problem here is to test whether two variables are correlated or not. The sampling allows estimation of the empirical correlation coefficient given by equation (5-5). Assume a trial gives a value for $\hat{\rho}$ near 0. How to accept the independence property?

The test is "no correlation", which translates to "$\rho = 0$". If the hypothesis is true, it can be shown that as the sample size increases, $\hat{\rho}$ goes to a normal distribution with mean 0 and variance $1/n$: this is nothing more than the central limit theorem – and this illustrates once again its importance in statistics.

The hypothesis is accepted, with a risk α, if:

$$\left|\hat{\rho}\sqrt{N}\right| < x_\alpha \tag{5-30}$$

(x_α is naturally the quantile corresponding to α).

This page intentionally left blank

6

Reliability

This chapter presents the major elements of *reliability theory*, and more generally what is called *dependability*.

Dependability encompasses all these aspects of reliability, availability, maintainability, which play a major role in communication systems and networks. Indeed ensuring the continuity of service is one of their first missions, especially in case of emergency (e.g. calling the fire brigade), and in any disaster situation. To reach such a goal, operators must deploy secured architectures. As this increases the cost of the system, it is worth evaluating their efficiency. Reliability theory is the set of techniques to apply in order to perform this evaluation. This chapter presents these techniques, giving numerous illustrating examples.

6.1. Definition of reliability

According to the standardization and particularly to the IEC (International Electrotechnical Commission), *reliability* is "*the probability that a system will perform its intended function for a specified time interval under stated conditions*". This will correspond to the "success" of the mission. On the opposite, the mission fails once the system breaks down. We will compute the *probability of success at time t*, or *probability of working for a given time interval t*.

In the telecommunication domain, and particularly for terrestrial equipments that are generally repairable, another important feature is the long-term proportion of time the service is available; one then refers to this concept as *availability*. The availability function is defined as the probability that the system is working (into service) at any instant of time t, whatever could be the preceding states (failures and repairs may have occurred).

A system is thus characterized both by its reliability (probability of uninterrupted activity up to t) and by its availability (probability of being active at time t). As the availability is usually high, the probability being very close to one, 0.99 and more, *unavailability* is used preferably. This is the complement to 1, usually expressed under the form 10^{-x}.

6.2. Failure rate and bathtub curve

The *instantaneous failure rate* $\lambda(t)$ is defined as follows: $\lambda(t)dt$ is the probability that a system, working correctly at time t fails within the time interval $[t, t + dt]$.

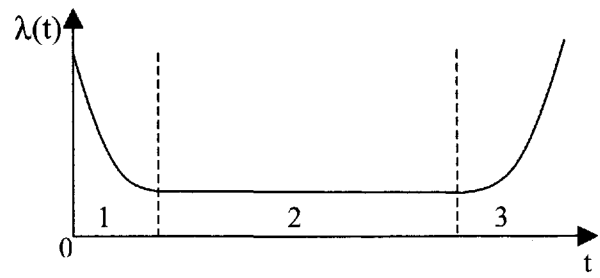
During the lifetime of a repairable system, or of a population of identical equipments, the observation leads to distinguish three periods, according to the behaviour of the failure rate:

The first period corresponds to the *early life period* (also called *infant mortality period*, or *burn in period)*. It is characterized by the decrease of the failure rate as time increases. This corresponds to the period during which remaining (hardware or software) defaults, which have not been detected while debugging the system, are corrected after its deployment.

The second period, or *middle life* period, corresponds to what is also called the *useful life period* and is characterized by an approximately constant failure rate. During this phase, the longest of the lifetime, failures occur – either hardware or software – which are referred to as "*catalectic*" or "*random*" i.e. unpredictable. Actually, it is not possible, nor needed, and not even profitable to identify and correct residual defaults as the failure rate becomes negligible.

The third period corresponds to the so-called *old age period*, also called *wear out period*, during which the failure rate increases with time. This corresponds to irreversible degradation of the system, either due to a material degradation of its components, or due to a lack of maintenance in the case of repairable systems. For terrestrial telecommunication systems this period is generally not considered because of the replacement of old equipments by new ones, while it can be of importance for satellite systems.

The curve of Figure 6.1 illustrates these three modes of behaviour. It is conventionally called the "bathtub curve" because of its shape.



1 = early life period, 2 = useful life period, 3 = wearout period

Figure 6.1. *Bathtub curve (mortality curve)*

In the following, the focus is mainly on the useful life period. Moreover, since repairable systems are our main concern, the concept of MTBF (*mean time between failures*) will be of prime importance. This is the inverse of the constant failure rate λ of the system during the useful life period.

Prior to begin modelling the major reliability structures, it is worth defining the main functions used in reliability studies.

6.3. Reliability functions

$R(t)$ is the reliability function and represents the probability of survival at time t.

One frequently characterizes the reliability by

$$M = \theta = \int_0^\infty R(t)dt. \tag{6-1}$$

M is the MTTF (mean time to failure), or MTBF (mean time between failures) for a repairable system where it represents the average time between failures. M is often denoted by θ.

$F(t)$ is the failure distribution function, $F(t) = 1 - R(t)$.

$\lambda(t)$ is the instantaneous failure rate (also called hazard function). It corresponds to the probability that a system breakdown occurs between times t and $t + dt$ provided that the system was on service at time t. Hence we have:

$$\lambda(t)=\frac{dF(t)}{dt}.\frac{1}{R(t)},$$

and thus:

$$\lambda(t)dt=-\frac{dR(t)}{R(t)}, \tag{6-2}$$

and also:

$$R(t)=e^{-\int_0^t \lambda(\tau)d\tau}. \tag{6-3}$$

In the useful life period $\lambda(t)=\lambda$ is constant, hence:

$$R(t)=e^{-\lambda t}, \text{ and } M=\frac{1}{\lambda}. \tag{6-4}$$

In what follows, our interest is mainly concentrated on that period.

Note however that more complex functions can be used, in order to describe the reliability of the equipment during its whole lifetime. Especially the Weibull distribution is often used (see Chapter 4). Indeed, let us express the reliability distribution under the form of a Weibull law:

$$R(t)=e^{-\left(\frac{t-\gamma}{\eta}\right)^{\beta}} \tag{6-5}$$

Identifying this expression with the general definition (6-3), one has:

$$R(t)=e^{-\int_0^t \lambda(\tau)d\tau}=e^{-\left(\frac{t-\gamma}{\eta}\right)^{\beta}},$$

and thus:

$$\int_0^t \lambda(t)dt=\left(\frac{t-\gamma}{\eta}\right)^{\beta};$$

then:

$$\lambda(t)=\frac{\beta}{\eta}\left(\frac{t-\gamma}{\eta}\right)^{\beta-1}. \tag{6-6}$$

Denoting $\lambda_0=\frac{1}{\eta^\beta}$ one has:

$$\lambda(t)=\lambda_0\beta\left(t-\gamma\right)^{\beta-1}, \text{ and } R(t)=e^{-\lambda_0(t-\gamma)^\beta}. \tag{6-7}$$

Clearly these expressions show that:

– the failure rate is decreasing with time for $\beta < 1$,

– the failure rate is constant for $\beta = 1$, and one finds again the exponential distribution if moreover $\gamma = 0$,

– the failure rate is increasing with time for $\beta > 1$.

6.4. Systems reliability

6.4.1. *Reliability of non repairable systems*

In the following any element of a system (component, card, rack...) will be referred to as E_i. Any element will be itself decomposable in other elements.

We will assume that an element may be in only two states: it is *up*, ie. it *works (E)*, or it is *down* ($\overline{E}$). We will also call these states *good* or *bad* respectively.

E and $\overline{E}$ are two mutually exclusive events:

$P(E) + P(\overline{E}) = 1.$

The reliability R is defined as the probability of being up, or probability of success (the time index is omitted, for short):

$R = P(E) = 1 - P(\overline{E}).$

Conforming to the usage, we will denote Q its complement, i.e. the probability of being down or probability of failure: $Q = 1 - R$.

6.4.1.1. *Reliability of the series configuration*

A system is said to be a *series* system if it fails *as soon as* any of its elements fails. It is symbolised by a *reliability diagram* represented in Figure 6.2.

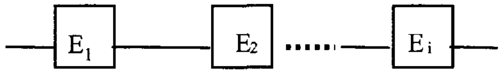


Figure 6.2. *Series configuration*

Let $\overline{E}$ denote the event "breakdown of the system" (or total failure) and $\overline{E_i}$ the event "failure of element i", we have:

$$\overline{E} = \overline{E_1} \cup \overline{E_2} \cup ... \cup \overline{E_i} \cup ...$$

Event algebra, and especially De Morgan rule, allows one to write:

$$\overline{E} = \overline{E_1 \cap E_2 \cap .... \cap E_i \cap ..},$$

and thus:

$$E = E_1 \cap E_2 ...$$

Going from the events to their probabilities, and considering the events as independent,

$$P(E) = \Pi P(E_i),$$

and thus for reliability:

$$R = \Pi R_i. \tag{6-8}$$

Which may be merely formulated by the following rule: the reliability of a series configuration is the product of the reliabilities of its components.

In the case where all the components have a failure rate exponentially distributed with parameter λ_i, one gets:

$$R(t) = e^{-\sum \lambda_i t}. \tag{6-9}$$

The MTBF of the system is:

$M = \int_0^\infty R(t)dt$, and thus:

$$M = \frac{1}{\sum \lambda_i} \tag{6-10}$$

6.4.1.2. *Reliability of the parallel configuration*

A system is said to be *parallel* if the system fails *only if* all its elements fail. The reliability diagram is represented in Figure 6.3.

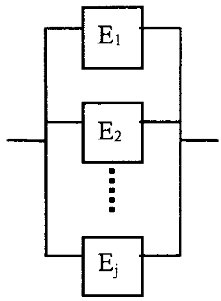


Figure 6.3. *Parallel configuration*

Then we have:

$$\overline{E} = \overline{E_1} \cap \overline{E_2} \cap .. \cap \overline{Ej} \cap ...$$

Hence:

$$P(\overline{E}) = \Pi P(\overline{Ej}),$$

or, since Q denotes the probability of failure:

$$Q = \Pi Qj \tag{6-11}$$

In words, we have the following simple rule: the probability of failure of a parallel system is the product of the probabilities of failure of its components.

Remember that we denote $Q = 1 - R$. Then we may write:

$$R = 1 - \Pi(1 - Rj). \tag{6-12}$$

The system is working as long as *at least* one of its elements is working. (Note that this kind of manipulation, consisting in operating either on an event or on its complement is of great help in modelling such systems, and, more generally, each time the event of interest is a combination of elementary events: see the analysis of switching networks in Chapter 9).

In the case where all elements have a failure law exponentially distributed with parameter λ_i, then:

$$R(t) = 1 - \Pi(1 - e^{-\lambda_i t}). \tag{6-13}$$

In the simple case of two elements in parallel, we have the obvious result:

$$R(t) = R_1(t) + R_2(t) - R_1(t)R_2(t),$$

$$R(t) = e^{-\lambda_1 t} + e^{-\lambda_2 t} - e^{-\lambda_1 t}e^{-\lambda_2 t}. \tag{6-14}$$

The MTBF of the system is:

$M = \int_0^\infty R(t)dt$, and then:

$$M = \frac{1}{\lambda_1} + \frac{1}{\lambda_2} - \frac{1}{\lambda_1 + \lambda_2}. \tag{6-15}$$

6.4.1.3. *Reliability of the series – parallel configuration*

This configuration corresponds to a set of groups of elements in parallel, each one formed of elements in series: k branches of p elements for instance.

The reliability diagram is represented in Figure 6.4.

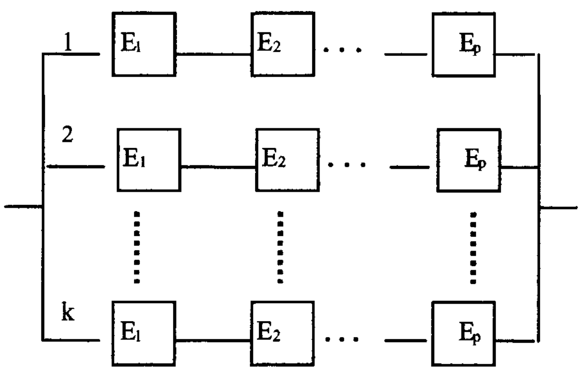


Figure 6.4. *Series – parallel configuration*

Proceeding as above yields immediately:

The reliability of one branch is:

$$Rj = \prod_{i=1}^{p} R_i .$$

The probability of failure of the whole system with k branches is:

$$Q = \prod_{j=1}^{k} Qj ,$$

and its reliability is $R = 1 - Q$, an thus:

$$R = 1 - \prod_{j=1}^{k}(1 - \prod_{i=1}^{p} R_i) . \tag{6-16}$$

6.4.1.4. *Reliability of the parallel-series configuration*

The reliability diagram is represented in Figure 6.5.

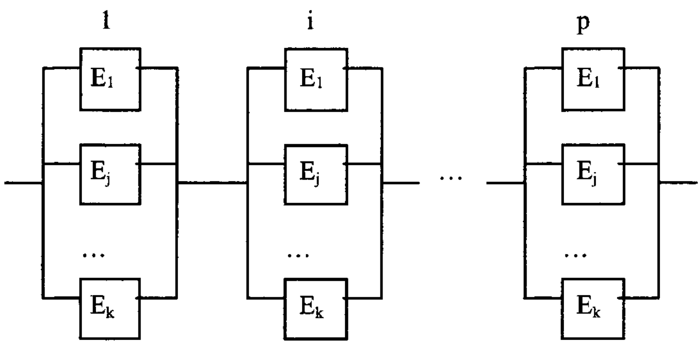


Figure 6.5. *Parallel-series configuration*

The reliability of one set of k elements in parallel is:

$$R = 1 - \prod_{j=1}^{k}(1 - Rj) ,$$

and the reliability of p sets in series is:

$$R = \prod_{i=1}^{p}\left[1 - \prod_{j=1}^{k}(1 - Rj)\right] . \tag{6-17}$$

6.4.1.5. *Complex configurations*

The method is briefly developed through an example, illustrated in Figure 6.6. The generalisation is straightforward.

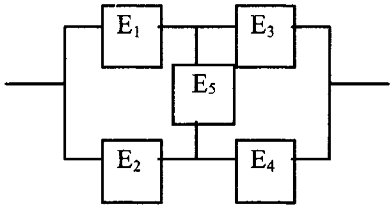


Figure 6.6. *"Meshed" configuration*

The more direct approach distinguishes two cases, according to the state of E_5 : E_5 is good, or E_5 is bad.

If E_5 has failed, the system is *up,* provided that *at least* one of the two branches (E_1, E_3) or (E_2, E_4) is working, i.e. the complement of the event "the two branches are down", the rule on elements in series giving the corresponding probability.

The probability of being *up*, corresponding to this case is:

$$(1-R_5)[1-(1-R_1R_3)(1-R_2R_4)].$$

If on the other hand E_5 works, the system is *up,* provided that in each couple (E_1, E_2) and (E_3, E_4) at least one element is working, the probability of which is 1 minus the probability that both elements are down.

That is:

$$R_5[1-(1-R_1)(1-R_2)][1-(1-R_3)(1-R_4)]$$

The reliability of the system is the sum of probability of these two mutually exclusive events:

$$\begin{aligned} R = &(1-R_5)[1-(1-R_1R_3)(1-R_2R_4)] \\ &+ R_5[1-(1-R_1)(1-R_2)][1-(1-R_3)(1-R_4)] \end{aligned} \tag{6-18}$$

6.4.1.6. *Non repairable redundant configurations*

The goal of such organisations is to ensure a higher reliability by inserting additional components in parallel. These structures are deployed especially in the case where it is impossible to replace failed components.

Simple redundancy

A breakdown of the system occurs either if all the elements in parallel are down (this is called *total simple redundancy*), or if a given proportion of them has failed (*partial simple redundancy*). For instance, a telecommunication system (a switching or transmission equipment, a router...), is built with many processing elements sharing the load. This leads to consider different degraded states as a function of the number of failed elements. Notice however that terrestrial systems will generally be repairable, so that for them one is mainly concerned with availability, as we will see below.

The total simple redundancy case corresponds directly to the parallel configuration: the system is down if *all* its components are down.

$$R = 1 - \prod_{1}^{n} (1 - Rj),$$

and thus with n identical elements:

$$R(t) = 1 - (1 - e^{-\lambda t})^{n}. \tag{6-19}$$

Hence the MTBF of the system is:

$$M = \int_{0}^{\infty} R(t)dt,$$

$$M = \frac{1}{\lambda} \sum_{1}^{n} \frac{1}{i} \tag{6-20}$$

For the simple case of two elements in parallel, we have:

$$M = \frac{3}{2\lambda}$$

For the partial simple redundancy, it corresponds to the case where a part of the components is up. This is a special case of majority redundancy, which we examine here.

Partial redundancy and majority voting

In that case, the system is in service as long as at least r out of n elements are in service.

The system is said a majority redundancy system, or a majority voting system, if furthermore there exists a decision element in series (for the reliability standpoint) making possible to detect whether at least r out of n elements are working, or not. Rather than sharing the load, the system replicates the same vital function (e.g. power supplies or clocks). The decision element simply makes the comparison and issues a signal corresponding to the "majority opinion". Here too the mission will be said successful as long as, at least, r elements out of the n elements in parallel are good, provided however that the voter does not fail.

The corresponding reliability diagram is represented in Figure 6.7.

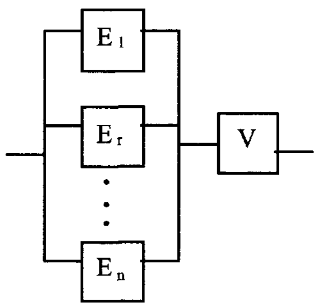


Figure 6.7. *Majority voting*

First let us consider the system without the majority voter (V). Its reliability function may be easily written using the binomial law:

$$R = \sum_{k=r}^{n} \binom{n}{k} R_j^k (1 - R_j)^{n-k},$$

with $\binom{n}{k} = \dfrac{n!}{k!(n-k)!}$.

Thus, with an exponential failure probability distribution:

$$R(t) = \sum_{k=r}^{n} \binom{n}{k} e^{-k\lambda t} (1 - e^{-\lambda t})^{n-k}. \tag{6-21}$$

Now let us consider the system with its majority voter. From the reliability standpoint, the voter, with failure rate λ_V, is an element in series. Thus the reliability of the whole system is:

$$R(t) = e^{-\lambda_V t} \sum_{k=r}^{n} \binom{n}{k} e^{-k\lambda t} (1 - e^{-\lambda t})^{n-k} \tag{6-22}$$

In the simple but common case of a three elements majority system with $r = 2$ (at least two elements must work), we have:

$$R(t) = e^{-\lambda_V t} e^{-2\lambda t} (3 - 2e^{-\lambda t}) .$$

6.4.2. *Reliability and availability of repairable systems*

These systems are such that as soon as one of their components fails, the system itself or an external repairperson corrects the defaut after a certain amount of time. The equipment is then said to be returned to service. Here the notion of interest is the (mean) *availability*, i.e. the stationary probability of finding the system in service. The simplest example is a set of n components in total simple redundancy, on which a technician intervenes to re-establish the state with n components in service. The system alternates between "in service" and "degradated" states. The long-term proportion of the time during which the system is operational, i.e. with n components active, is the availability. Availability is conventionally denoted A. Its complement, termed the *unavailability* and denoted U, is also frequently used, as we will see below.

The period during which less than n elements are active is called the *degradated regime duration*. Clearly, this duration is shorter as repairs are less frequent and faster.

6.4.2.1. *State equations*

The process of going from state to state is nothing else that a *birth and death* process, and is given a thorough attention in Chapter 7. The present analysis introduces this notion, further developed later on.

Let denote as $i = 1, 2, \ldots, n$ the different system states, corresponding to i active elements, and let $P(i, t)$ the probability that the system is in state i at time t. Whatever the specific structure of the system, a general basic set of equations describes the evolution of the state equations:

$$P(i,t+dt) = P(i+1,t)d(i+1,t)dt + P(i,t)[1-d(i,t)dt][1-r(i,t)dt] + P(i-1,t)r(i-1,t)dt.$$

This equation translates the fact that $P(i,t+dt)$, probability of finding the system in state i at time $(t+dt)$, is the sum of:

a) the probability to be in state $(i+1)$ at time t, $P(i+1,t)$, and a failure has occurred, moving the system to state i in the interval dt, i.e. $d(i+1,t)dt$;

b) increased by the probability of being in state i at time t, $P(i,t)$, and no event has occurred during dt: no failure $[1-d(i,t)dt]$ and no repair $[1-r(i,t)dt]$;

c) and by the probability of being in state $(i-1)$ at time t, $P(i-1,t)$, and that a component has been repaired (rate $r(i-1,t)dt$) moving to state i during the interval dt.

This holds, as the events are clearly mutually exclusive and neglecting simultaneous event (probability proportional to dt^2, see Chapter 7).

The equation is transformed as follows:

$$\frac{P(i,t+dt)-P(i,t)}{dt} = P(i+1,t)d(i+1,t) + P(i-1,t)r(i-1,t) - P(i,t)[d(i,t)+r(i,t)].$$

Now, letting $dt \to 0$ and assuming the derivative exists, one gets:

$$P'(i,t) = P(i+1,t)d(i+1,t) + P(i-1,t)r(i-1,t) - P(i,t)[d(i,t)+r(i,t)]. \quad (6\text{-}23)$$

This result is quickly obtained, using the method of Markov graphs, also called state transition diagrams, as will be further developed in Chapter 7 (see Figure 6.8).

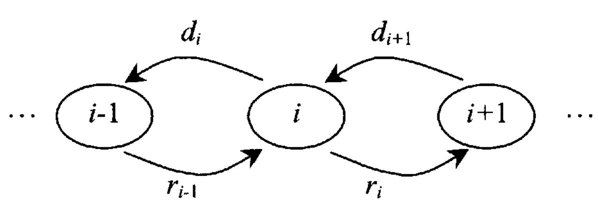


Figure 6.8. *Markov graph or state transition diagram*

As the states are mutually exclusive, one must have

$$\sum P(i,t) = 1 \text{, and thus } \sum P'(i,t) = 0 \,. \tag{6-24}$$

These equations must be completed by the initial conditions, i.e. the system state at $t = 0$. Generally, it is admitted that all elements are good at the origin, so that $P(n, 0) = 1$ if the system is composed with n elements and $P(i, 0) = 0$ for $i \neq 0$.

Lastly, one considers generally that both the failure and repair processes are stationary: the probabilities $d(i,t)$ and $r(i,t)$ do not depend on t. The system should thus reach a state of statistical equilibrium as the time increases, characterised by the condition:

$P'(i,t) = 0$ for every state i.

The set of these equations is a system of differential equations, which allows studying reliability for various structures of the redundant repairable systems class.

6.4.2.2. *Reliability of redundant repairable systems*

We consider the same configurations as above, for non-repairable systems, and apply the method for exponentially distributed failure time and repair duration. As will be explained in Chapter 10, this is in good accordance with actual observation.

Simple redudancy with two elements

The system consists of two elements in parallel: if one fails, the second keeps working and the system survives.

First, let us compute the *availability* of the system, i.e. the stationary probability to be in one of the states: one or two elements into service.

For repairable systems, this is a major QoS criterion, as it reflects the degree of availability from the user viewpoint (examples concern the availability of telecommunication service for emergency calls).

As above, the state i is the one with i elements into service. Each element has a failure rate λ, so when i elements operate, the probability of a failure of one of them is:

$d(i,t) = i\lambda$

The repair process is such that failed elements are restored with a constant repair rate (one repair and one at the same time):

$r(i,t) = \mu$

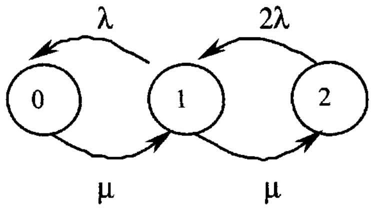


Figure 6.9. *Simple redundancy with two elements*

From the state transition diagram of Figure 6.9, we derive directly the set of differential equations:

$$P'(0,t) = \lambda P(1,t) - \mu P(0,t),$$

$$P'(1,t) = 2\lambda P(2,t) - (\lambda + \mu)P(1,t) + \mu P(0,t),$$

$$P'(2,t) = \mu P(1,t) - 2\lambda P(2,t).$$

At statistical equilibrium, probabilities *P(i,t)* become independent of time, so that $P'(i,t) = 0$ and thus:

$$\begin{cases} \mu P_0 = \lambda P_1 \\ (\lambda + \mu)P_1 = \mu P_0 + 2\lambda P_2 \\ 2\lambda P_2 = \mu P_1 \end{cases}$$

which yields:

$$P_1 = \frac{2\lambda}{\mu} P_2, P_0 = (\frac{2\lambda}{\mu})(\frac{\lambda}{\mu})P_2,$$

adding the condition:

$$P_0 + P_1 + P_2 = 1,$$

we obtain:

$$P_2 = \frac{1}{1+\dfrac{2\lambda}{\mu}+\dfrac{2\lambda^2}{\mu^2}},\ P_1 = \frac{\dfrac{2\lambda}{\mu}}{1+\dfrac{2\lambda}{\mu}+\dfrac{2\lambda^2}{\mu^2}},\ P_0 = \frac{\dfrac{2\lambda^2}{\mu^2}}{1+\dfrac{2\lambda}{\mu}+\dfrac{2\lambda^2}{\mu^2}}. \qquad (6\text{-}25)$$

The state associated to P_0 corresponds to the breakdown of the system. Hence, the availability A of the system, from a service viewpoint, is given by: $A = 1 - P_0$.

Thus:

$$A = \frac{\mu^2 + 2\mu\lambda}{\mu^2 + 2\mu\lambda + 2\lambda^2} \qquad (6\text{-}26)$$

In practise, the level of quality is such that the availability is close to the unity (see below). The characterisation is done preferably by the system unavailability, denoted as U, expressed under the form 10^{-x}.

Evidently, $U = 1 - A = P_0$.

Given the relative magnitude of λ (between 10^{-4}/h and 10^{-6}/h, i.e. MTBF of several years), and of μ (0.1 or 0.2 per hour, i.e. repair times of a few hours), one can simplify the above expressions. It comes:

$$U = P_0 \approx \frac{2\lambda^2}{\mu^2}$$

Often the procedures of repairments are such that the system, when down, will directly switch to the full availability state. The state transition diagram is then represented in Figure 6.10:

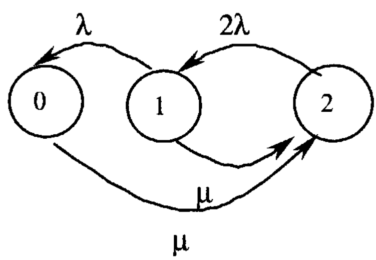


Figure 6.10. *Redundancy with return to total service*

From the graph, we may directly write the following equations:

$\mu P_0 = \lambda P_1$,

$(\lambda + \mu) P_1 = 2\lambda P_2$,

$2\lambda P_2 = \mu P_1 + \mu P_0$,

which, combined with:

$P_0 + P_1 + P_2 = 1$

yields:

$$P_2 = \frac{1}{1 + \frac{2\lambda}{\lambda + \mu} + \frac{2\lambda^2}{\mu(\lambda + \mu)}}, \quad P_1 = \frac{\frac{2\lambda}{\lambda + \mu}}{1 + \frac{2\lambda}{\lambda + \mu} + \frac{2\lambda^2}{\mu(\lambda + \mu)}},$$

$$P_0 = \frac{\frac{2\lambda^2}{\mu(\lambda + \mu)}}{1 + \frac{2\lambda}{\lambda + \mu} + \frac{2\lambda^2}{\mu(\lambda + \mu)}},$$

thus:

$$P_0 = \frac{2\lambda^2}{3\lambda\mu + \mu^2 + 2\lambda^2}, \tag{6-27}$$

and finally:

$$A = \frac{3\lambda\mu + \mu^2}{3\lambda\mu + \mu^2 + 2\lambda^2}. \tag{6-28}$$

Referring once again to the orders of magnitude, as here again $\mu >> \lambda$, the unavailability is approximately:

$$U = P_0 \approx \frac{2\lambda^2}{\mu^2}.$$

From a practical viewpoint, the two models are equivalent.

k out of n elements redundancy

We consider here the system "*k* out of *n*" already discussed, assuming now repairable elements. The system is considered as working correctly as long as *k* (at least) elements are operational.

We take account of the previous remarks about the practical equivalence between the repair procedures, and we only address the case with simple repair. The state transition diagram is represented in Figure 6.11.

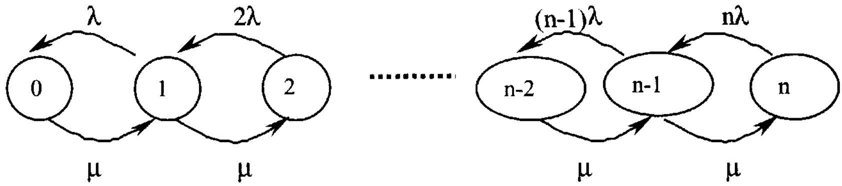


Figure 6.11. ***k*** *out of* ***n*** *elements redundancy*

From the graph, we then easily derive the following relationships between the state probabilities:

$$P_{n-1} = \frac{n\lambda}{\mu} P_n,$$

$$P_{n-2} = \frac{n\lambda}{\mu}\frac{(n-1)\lambda}{\mu} P_n = \frac{n!}{(n-2)!}\frac{\lambda^2}{\mu^2} P_n,$$

and more generally:

$$P_{n-i} = \frac{n!}{(n-i)!}\frac{\lambda^i}{\mu^i} P_n, \qquad (6\text{-}29)$$

which may also be written:

$$P_j = \frac{n!}{j!}\left(\frac{\lambda}{\mu}\right)^{n-j} P_n,$$

and thus, for a *k* out of *n* redundant system:

$$A = \sum_{j=k}^{n} P_j ,$$

$$U = \sum_{j=0}^{k-1} P_j .$$

Taking account here again of the current orders of magnitude:

$$U \approx P_{k-1} \approx \frac{n!}{(k-1)!} \left(\frac{\lambda}{\mu}\right)^{n-k+1} . \qquad (6\text{-}30)$$

Many real redundant systems consist of n elements, with only one of them for spare. For instance, it will be the case of several processors sharing the load, or even performing different tasks (different functions) but with a standby one, able to replace any other that failed.

For this type of system we then have $k = n - 1$. Thus:

$$U \approx P_{n-2} \approx n(n-1)\frac{\lambda^2}{\mu^2}$$

NOTE – This kind of system is often called $n + 1$-redundant. The expression for U becomes:

$U \approx (n+1)n\dfrac{\lambda^2}{\mu^2}$, this is just a question of notation.

Sequential systems

Sequential systems are such that a single element is working while the other ones are waiting and ready to replace it in case of failure. This organisation is called *standby redundancy*. The failure rate for all the states is:

$$d(i,t) = \lambda$$

and the transition diagram is given in Figure 6.12, for the case of two elements.

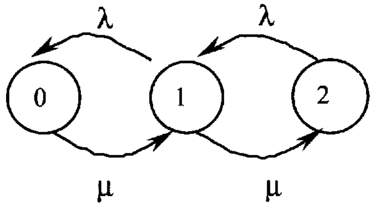


Figure 6.12. *Sequential redundancy*

We easily derive the following relationships:

$$\mu P_0 = \lambda P_1 ,$$

$$(\lambda + \mu) P_1 = \mu P_0 + \lambda P_2 ,$$

$$\lambda P_2 = \mu P_1 ,$$

$$P_0 + P_1 + P_2 = 1 ,$$

and thus:

$$A = \frac{\mu^2 + \mu\lambda}{\mu^2 + \mu\lambda + \lambda^2} . \qquad (6\text{-}31)$$

Thus for the unavailability:

$$U = P_0 = \frac{\dfrac{\lambda^2}{\mu^2}}{1 + \dfrac{\lambda}{\mu} + \dfrac{\lambda^2}{\mu^2}} \qquad (6\text{-}32)$$

That is, given the orders of magnitude:

$$U \approx \frac{\lambda^2}{\mu^2} .$$

That redundant organisation lowers unavailability, as can be expected since the standby unit is activated only if needed. However, its weakness is that all tasks in progress (call set-up, packet forwarding, etc.) are lost upon a failure of the active unit, as the spare unit does not process them in parallel.

Furthermore, the general case of n elements with one spare element (redundancy said of type $n + 1$), gives for the unavailability:

$$U \approx n^2 \frac{\lambda^2}{\mu^2},$$

which is to compare with the previous one for the simple redundancy, i.e. $U \approx (n+1)n\frac{\lambda^2}{\mu^2}$. The difference is not significant, as soon as n is large enough.

That kind of organisation generally does not apply for the hardware part of telecommunication equipment, as all elements, including standby ones, are permanently powered on and thus subject to failures. On the other hand, this model is quite well suited to software, as in the case of passive redundancy (see below), where the software modules are executed only when the element carrying them becomes active.

6.4.2.3. *Imperfect structures*

The term stands for systems having imperfections at the level of the defence mechanisms. In most cases, the defence mechanisms are rather simple. Moreover, the usual orders of magnitude, as we have already remarked, justify the application of simple and tractable models. However, one must always keep on mind certain critical aspects of real systems, and especially the imperfect nature of the defence mechanisms. Therefore, we have to deal with structures, simple in their principles, but really imperfect in their reality. This leads to complexify the models, since we need to take account of these imperfections, for reliability modelling.

This gives the opportunity to summarise the major structures of redundant systems. Aside the type of the structure: simple or partial redundancy, majority voting, one has to distinguish between *passive* (or *stand-by*) redundancy and the *active* (or *static*) redundancy.

Passive redundancy. Elements in standby are activated only in case of failure. This type of redundancy is referred to as *dynamic*, as it implies real time switching operation from standby to active state.

Active redundancy. All the components, the ones in use as well as the ones for redundancy, are simultaneously active. There is no explicit switching operation when a failure occurs, and one talks about static redundancy.

All these modes of redundancy are widely in use in communication systems in order to ensure high service availability. The whole set of hardware and software resources implied constitute the defence mechanisms. They include all the decision processes associated with fault detection, isolation of the faulty element and traffic redirection to the working elements. Unfortunately, all these mechanisms are most of the time imperfect.

The following example, typical of real systems, serves to address these aspects of reliability.

Redundant structure with silent failure

In this system, the spare component is in passive redundancy and the detection mechanism is imperfect. Failure rates of both the active component and the spare one are identical (the common situation, in telecommunication systems, however the analysis is easily generalized).

We assume that every fault on the active element is detected (a reasonable assumption, as the default appears rapidly at the application level). On the other hand, we will denote as c the coverage coefficient, which measures the efficiency of the detection mechanism on the passive element: this is the percentage of failures actually detected, for instance by periodic tests, routines etc..

Lastly, one assumes negligible the probability of two successive failures on the passive element, such as the first undetected and the second detected. The state transition diagram is given in Figure 6.13.

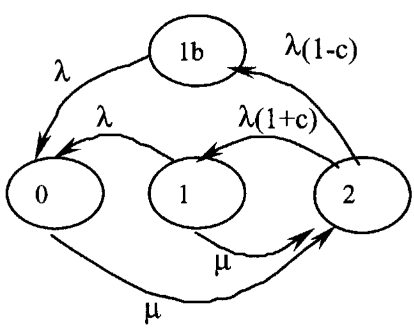


Figure 6.13. *Redundancy with silent failure*

State "1" corresponds to the failure of the active element, or a detected failure of the passive one. State "1b" follows an undetected failure of the passive element so that the system passes from state "1b" to state "0" upon the failure of the active element.

We then derive the following system of equations:

$$\mu P_0 = \lambda P_1 + \lambda P_{1b},$$

$$(\lambda + \mu) P_1 = \lambda (1 + c) P_2,$$

$$\lambda P_{1b} = \lambda (1 - c) P_2,$$

$$\lambda P_2 = \mu P_0 + \mu P_1,$$

$$P_0 + P_1 + P_{1b} + P_2 = 1,$$

and thus with $A = 1 - P_0$,

$$A = \frac{3\lambda\mu + 2\mu^2 - c\mu^2}{4\lambda\mu - \lambda c\mu + 2\mu^2 - c\mu^2 + 2\lambda^2}. \tag{6-33}$$

For $c = 1$, we obtain of course the same results as for the system without any silent failure:

$$A = \frac{3\lambda\mu + \mu^2}{3\lambda\mu + \mu^2 + 2\lambda^2}.$$

It is interesting to try better understanding the contribution of the coverage coefficient c. For this, let us take the expression of the unavailability:

$$U = 1 - A$$

$$U = \frac{\lambda\mu - \lambda c\mu + 2\lambda^2}{4\lambda\mu - \lambda c\mu + 2\mu^2 - c\mu^2 + 2\lambda^2}. \tag{6-34}$$

Then, observing that in practice (as we will see in Chapter 9), we have $\mu >> \lambda$:

$$U \approx \frac{\lambda\mu(1-c) + 2\lambda^2}{\mu^2(2-c)}.$$

Thus: when $c \to 1, U \approx \dfrac{2\lambda^2}{\mu^2}$ and for $c \to 0, U \approx \dfrac{\lambda}{2\mu}$

Clearly, the effect of the coverage factor on unavailability is considerable (it goes from a linear dependency to a quadratic one).

6.4.3. *Using Laplace transform*

Up to now, our concern has been the availability of repairable systems. Let us now estimate the reliability of these systems. Here one looks for the time-dependant probability to be in the different states, given that the system stops operating as soon as it reaches the state of total failure – state 0, where there is no possible repair. As previously, the system behaviour is modelled by a system of differential equations, but instead of statistical equilibrium state probabilities, we are now interested in the time-dependent state probabilities.

An example helps showing how the Laplace transform can help in solving the problem (see Chapter 3). We take again the configuration with two elements in simple redundancy. Figure 6.14 gives the state transition diagram.

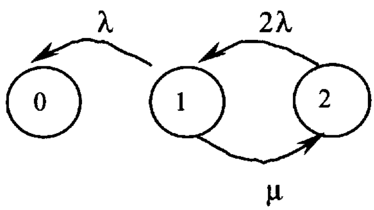


Figure 6.14. *Simple redundancy (reliability)*

The corresponding equations are:

$$P'(0,t) = \lambda P(1,t),$$

$$P'(1,t) = 2\lambda P(2,t) - (\lambda + \mu)P(1,t),$$

$$P'(2,t) = \mu P(1,t) - 2\lambda P(2,t).$$

As in Chapter 3, let us denote as $F^*(s)$ the Laplace transform of $f(t)$. Remember that:

$$F^*(s) = \int_{0-}^{\infty} f(t)e^{-st}dt .$$

This relationship is symbolised by:

$$f(t) \Leftrightarrow F^*(s) .$$

We have also:

$$\frac{df(t)}{dt} \Leftrightarrow sF^*(s) - f(0_-) ,$$

$$\int_{-\infty}^{t} f(t)dt \Leftrightarrow \frac{F^*(s)}{s} ,$$

and in particular for the exponential function:

$$Ae^{-at} \Leftrightarrow \frac{A}{s+a} ,$$

and for the unit step function:

$$\mu(t) \Leftrightarrow \frac{1}{s} .$$

Now let us apply the Laplace transform to the above set of differential equations, and let P_i* denote the transform of $P(i, t)$ for simplicity.

We first determine the initial conditions. Assuming (logically) that all the elements are good at time $t = 0$, we have for the initial states: $P_2(0) = 1$, $P_1(0) = P_0(0) = 0$.

Then the application of the Laplace transform combined with the above result yields the following set of linear equations:

$$sP_0^* = \lambda P_1^* ,$$

$$sP_1^* = 2\lambda P_2^* - (\lambda + \mu)P_1^* ,$$

$$sP_2^* - 1 = \mu P_1^* - 2\lambda P_2^* .$$

This gives:

$$P_0^* = \frac{2\lambda^2}{s(s^2 + s(3\lambda + \mu) + 2\lambda^2)}. \tag{6-35}$$

Or, if a_1 and a_2 are the roots of the denominator:

$$P_0^* = \frac{2\lambda^2}{s(s+a_1)(s+a_2)}, \text{ with}$$

$$a_1 = \frac{-(3\lambda + \mu) + \sqrt{\lambda^2 + \mu^2 + 6\lambda\mu}}{2}, \; a_2 = \frac{-(3\lambda + \mu) - \sqrt{\lambda^2 + \mu^2 + 6\lambda\mu}}{2}.$$

Then, observing that:

$$\frac{1}{(s+a_1)(s+a_2)} = \frac{-1}{a_1 - a_2}\left(\frac{1}{s+a_1} - \frac{1}{s+a_2}\right),$$

and that one can decompose $\frac{1}{s(s+a)} = \frac{-1}{a}\left(\frac{1}{s+a} - \frac{1}{s}\right)$,

we obtain:

$$\frac{1}{s(s+a_1)(s+a_2)} = \frac{-1}{a_1 - a_2}\left[\frac{-1}{a_1}\left(\frac{1}{s+a_1} - \frac{1}{s}\right) + \frac{1}{a_2}\left(\frac{1}{s+a_2} - \frac{1}{s}\right)\right].$$

This result allows coming back directly to the original (untransformed) function:

$$P(0,t) = \frac{-2\lambda^2}{a_1 - a_2}\left(\frac{-e^{a_1 t}}{a_1} + \frac{1}{a_1} + \frac{e^{a_2 t}}{a_2} - \frac{1}{a_2}\right).$$

This result is combined with:

$$a_1 a_2 = 2\lambda^2,$$

yielding:

$$P(0,t) = 1 - \frac{a_1 e^{a_2 t} - a_2 e^{a_1 t}}{a_1 - a_2}. \tag{6-36}$$

Thus finally the reliability function is:

$$R(t) = \frac{a_1 e^{a_2 t} - a_2 e^{a_1 t}}{a_1 - a_2}, \tag{6-37}$$

and the MTBF:

$$\theta = \int_0^\infty R(t)dt = -\frac{a_1 + a_2}{a_1 a_2},$$

or, $$\theta = \frac{3\lambda + \mu}{2\lambda^2}. \tag{6-38}$$

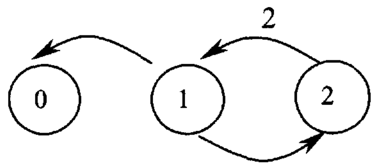
This last result could of course be easily obtained directly from the graph as seen in the preceding sections, but not the expression of $R(t)$ as a function of t.

This example shows clearly the power of the method, but also that its application to complex systems is rapidly tedious. We explain in the next section how to analyse numerically certain aspects of complex systems by means of matrix methods. However, whatever the approach, approximations are most often mandatory in order to derive explicit results, and especially when retrieving the original function. This emphasises the importance of being able to detect states with negligible probability, even for availability at statistical equilibrium.

6.4.4. *Matrix utilization*

In order to estimate the reliability – or the unavailability – we have proceeded, up to now, by directly solving the system of differential equations derived from the state transition diagram. Another approach is to represent the system under its matrix form and then to solve, either analytically or numerically, the matrix equation through appropriate software solver tools.

For instance, let us reconsider the case of two elements in simple redundancy, as already studied in the section on reliability. We had the following graph and equations:



$$P'(0,t) = \lambda P(1,t)$$

$$P'(1,t) = 2\lambda P(2,t) - (\lambda + \mu)P(1,t)$$

$$P'(2,t) = \mu P(1,t) - 2\lambda P(2,t)$$

Writing these equations into the matrix form, the system reduces to a single matrix equation, as follows:

$$\begin{bmatrix} P'(o,t) \\ P'(1,t) \\ P'(2,t) \end{bmatrix} = \begin{bmatrix} 0 & \lambda & 0 \\ 0 & -(\lambda+\mu) & 2\lambda \\ 0 & \mu & -2\lambda \end{bmatrix} . \begin{bmatrix} P(o,t) \\ P(1,t) \\ P(2,t) \end{bmatrix}$$

Equation which may also be directly derived from the graph, since the matrix coefficients, a_{ij} (i = row and j = column), are merely the transition rates from state i to state j.

Let us now consider the duration T of the first period of time (cycle) leading the system to the down state. The integration of the matrix equation from 0 to T yields:

$$\int_0^T P_i'(t)dt = P_i(T) - P_i(0),$$

and also:

$$\int_0^T P_i(t)dt = T_i,$$ average time spent in state i.

Then, one has the very general following matrix relationship:

$$\begin{bmatrix} P(0,T) - P(0,0) \\ P(1,T) - P(1,0) \\ P(2,T) - P(2,0) \end{bmatrix} = [\Lambda] . \begin{bmatrix} T_0 \\ T_1 \\ T_2 \end{bmatrix} \tag{6-39}$$

where $[\Lambda]$ denotes the transition rates matrix.

Now, taking account of the initial conditions, we obtain for our example:

$$\begin{bmatrix} 1-0 \\ 0-0 \\ 0-1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} = [\Lambda].\begin{bmatrix} T_0 \\ T_1 \\ T_2 \end{bmatrix}$$

whose resolution, through the corresponding system of equations, gives:

$T_2 = \dfrac{\lambda + \mu}{2\lambda^2}$, and $T_1 = \dfrac{1}{\lambda}$.

The average time up to the down state, i.e. the MTBF, is merely the sum of T_2 and T_1, thus:

$$\theta = \frac{3\lambda + \mu}{2\lambda^2}. \tag{6-40}$$

This is the same result as already obtained in equation (6-38). The main advantage of this approach is that it gives the possibility of solving numerically the matrix system (exact solution by matrix inversion, or approximate methods, such as Euler or Runge-Kutta methods). This is especially attracting for complex systems having a large number of states and thus of equations. We present now that kind of resolution.

6.4.4.1. *Exact resolution by inversion*

Before illustrating by an example, we recall briefly the method and particularly how to inverse matrices.

Let us call I the set of initial states, and T the unknown average durations in the different states. From the above general matrix equation (6-39) we have:

$$[I] = [\Lambda].[T],$$

and then:

$$T = [\Lambda]^{-1}.[I],$$

provided that the inverse matrix exists, which is fortunately the normal case for practical applications. Furthermore, in our problems, matrices are also generally square matrices and thus regular (every matrix that may be inversed is regular).

There are essentially two methodes for matrix inversion: the direct *determinant method* and the *triangulation*, or *Gauss elimination*. Although it needs more calculation, let us recall the simple determinant method, as it allows refreshing all the basic notions concerning matrices:

a) permute rows and columns,

b) replace each element by its cofactor,

c) divide by the determinant.

The method of course applies as well to availability as to reliability calculation, and we now describe an example of application in the case of statistical equilibrium.

First, let us remind how to obtain the determinant of a matrix. The determinant associated to a matrix A of elements a_{ij} (i = row and j = column) is defined by (expanding for instance with respect to the first row):

$$\det(A) = |A| = \sum_j (-1)^{j+1} \det(A_{1,j}), \tag{6-41}$$

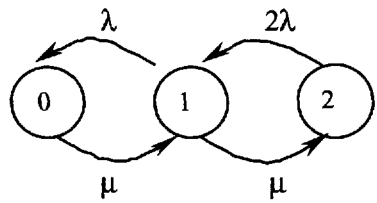
where $A_{1,j}$ is the "sub matrix" of A, and $\det(A_{1,j})$ the cofactor of a_{1j}, obtained when the first row and the j^{th} column are deleted. For instance for the basic case of a four terms matrix, we have:

$|A| = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21}$. Then for a nine terms matrix we have:

$$|A| = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11}\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12}\begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13}\begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}$$

Now, let us reconsider the case of a redundant repairable system composed of two elements, as in section 6.4.2.2.

We have:



$$P'(0,t) = \lambda P(1,t) - \mu P(0,t),$$

$$P'(1,t) = 2\lambda P(2,t) - (\lambda + \mu)P(1,t) + \mu P(0,t),$$

$$P'(2,t) = \mu P(1,t) - 2\lambda P(2,t),$$

and of course the conservation condition:

$$\sum_i P_i = 1.$$

This set of equations may be written:

$$\begin{bmatrix} 1 \\ P'(o,t) \\ P'(1,t) \\ P'(2,t) \\ \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ -\mu & \lambda & 0 \\ \mu & -(\lambda+\mu) & 2\lambda \\ 0 & \mu & -2\lambda \\ \end{bmatrix} . \begin{bmatrix} P(o,t) \\ P(1,t) \\ P(2,t) \end{bmatrix}.$$

In the steady state, or statistical equilibrium, the state probabilities are independent of time and derivatives are equal to zero, then we have:

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = [\Lambda] . \begin{bmatrix} P_0 \\ P_1 \\ P_2 \end{bmatrix}, \text{ with } [\Lambda] = \begin{bmatrix} 1 & 1 & 1 \\ -\mu & \lambda & 0 \\ \mu & -(\lambda+\mu) & 2\lambda \\ 0 & \mu & -2\lambda \\ \end{bmatrix}$$

Furthermore, one can easily verify that only two out of the three initial equations are independent. Thus, deleting for instance the last row (note that this is the sum of the two preceding rows), the system reduces to:

$$\begin{bmatrix}1\\0\\0\end{bmatrix}=[\Lambda].\begin{bmatrix}P_0\\P_1\\P_2\end{bmatrix}, \text{ with } [\Lambda]=\begin{bmatrix}1 & 1 & 1\\-\mu & \lambda & 0\\ \mu & -(\lambda+\mu) & 2\lambda\end{bmatrix},$$

which determinant is:

$$|\Lambda|=1\begin{vmatrix}\lambda & 0\\-(\lambda+\mu) & 2\lambda\end{vmatrix}-1\begin{vmatrix}-\mu & 0\\ \mu) & 2\lambda\end{vmatrix}+1\begin{vmatrix}-\mu & \lambda\\ \mu & -(\lambda+\mu)\end{vmatrix},$$

$$|\Lambda|=2\lambda^2+2\lambda\mu+\mu^2$$

Now we permute (transpose) the matrix:

$$t[\Lambda]=\begin{bmatrix}1 & -\mu & \mu\\1 & \lambda & -(\lambda+\mu)\\1 & 0 & 2\lambda\end{bmatrix}$$

Then, replacing each term by its cofactor, we obtain the cofactor matrix:

$$c[\Lambda]=\begin{bmatrix}\begin{vmatrix}(\lambda) & -(\lambda+\mu)\\0 & 2\lambda\end{vmatrix} & -\begin{vmatrix}1 & -(\lambda+\mu)\\1 & 2\lambda\end{vmatrix} & \begin{vmatrix}1 & \lambda)\\1 & 0\end{vmatrix}\\ -\begin{vmatrix}-\mu & \mu\\0 & 2\lambda\end{vmatrix} & \begin{vmatrix}1 & \mu\\1 & 2\lambda\end{vmatrix} & -\begin{vmatrix}1 & -\mu\\1 & 0\end{vmatrix}\\ \begin{vmatrix}-\mu & \mu\\ \lambda & -(\lambda+\mu)\end{vmatrix} & -\begin{vmatrix}1 & \mu\\1 & -(\lambda+\mu)\end{vmatrix} & \begin{vmatrix}1 & -\mu\\1 & \lambda)\end{vmatrix}\end{bmatrix}=\begin{bmatrix}2\lambda^2 & -(3\lambda+\mu) & -\lambda\\2\lambda\mu & 2\lambda-\mu) & -\mu\\ \mu^2 & \lambda+2\mu & \lambda+\mu\end{bmatrix}$$

And finally, dividing by the determinant:

$$\begin{bmatrix}P_0\\P_1\\P_2\end{bmatrix}=\frac{1}{|\Lambda|}c[\Lambda]\begin{bmatrix}1\\0\\0\end{bmatrix}$$

We obtain:

$$P_0=\frac{2\lambda^2}{\mu^2+2\lambda\mu+2\lambda^2},\ P_1=\frac{2\lambda\mu}{\mu^2+2\lambda\mu+2\lambda^2},\ P_2=\frac{\mu^2}{\mu^2+2\lambda\mu+2\lambda^2}$$

These are naturally the results already obtained. As already mentioned, these calculations are attractive only as they can be automated, allowing numerical resolutions of complex systems.

One can find numerous mathematical software packages performing such calculations. They rather make use of methods demanding less run time, as the Gauss elimination.

6.4.4.2. *Approximate solutions*

These methods are especially suited to evaluate such quantities as reliability at a given time t (as we have seen with Laplace transforms). Indeed, they operate directly on the differential system and look for the solution iteratively. The two major methods are the Euler and Runge-Kutta methods, both being based upon the same principle. Runge-Kutta method is clearly the most precise, at the price of more calculation. Here too, numerous software packages implement these methods.

We recall here the basic principle through the presentation of the Euler's method. We are given the equation $y' = f(x,y)$, with the initial conditions x_0, y_0. One searches a solution in the interval $[x_0, x]$. The interval is decomposed in elementary sub-intervals $[x_{n-1}, x_n]$, and for each interval boundary, one writes $y_n = y_{n-1} + f(x_{n-1}, y_{n-1})(x_n - x_{n-1})$, beginning from the initial point $y_1 = y_0 + f(x_0, y_0)(x_1 - x_0)$, and going step by step up to the target point x.

This amounts to approximate the curve f by its tangent at the x_i's (actually one writes $y'_1 = (y_1 - y_0)/(x_1 - x_0)$). This is Euler's method. The precision depends on the increment, and the error cumulates on each step from 0 to the target x. Runge Kutta method proceeds the same way, step by step, except that at each point it uses a Taylor development, generally at the order 4 (note that Euler's method makes no more than to use a Taylor series of order 1, see Appendix 1). This increases the precision, at the price of processing time. For a Runge Kutta of order 4, one has the following formulas:

$y_n = y_{n-1} + \frac{1}{6}(k_0 + 2k_1 + k_3)$, with:

$k_0 = hf(x_{n-1}, y_{n-1})$, $k_1 = hf(x_{n-1} + \frac{h}{2}, y_{n-1} + \frac{k_0}{2})$,

$k_2 = hf(x_{n-1} + \frac{h}{2}, y_{n-1} + \frac{k_1}{2})$, $k_3 = hf(x_{n-1} + h, y_{n-1} + k_2)$, $h = x_n - x_{n-1}$

6.5. Software reliability

Software reliability raises several specific issues, and we detail some of them in this section. We must first stress the importance of this point, as software complexity keeps increasing with the development of the services and becomes a major factor in system dependability.

The analysis of software reliability proceeds a similar way as was presented for hardware, and especially several behaviours may be distinguished: reliability growth and constant failure rate. The phenomenon is complex but however two main period may be indentified. During the developing and debugging period, the reliability is growing as errors are detected and corrected. Then during the operational (or useful life period), a failure rate almost constant is observed. More exactly, successive periods of increase and decrease then stability are observed, which correspond for telecommunication products to the introduction of new releases, which introduce new services and improve the performance of older ones, but which bring new early-life defaults. In fact, despite the continuous effort of correction during the test and operational phases, a certain number of defaults are likely to remain, associated with quite specific operating conditions. That is the reason why the focus will be put on redundancy, which as we will see, allows tolerating most of these defaults, and the effort will be concentrated in eliminating these faults provoking the total breakdown of the system.

At last, one does not observe any wear out period, at least for terrestrial equipments, as these products are generally replaced before reaching this phase.

6.5.1. *Reliability growth model, early-life period*

We are concerned here with the increase of reliability related with the progressive correction of the residual defaults in the system. Note that we address this issue in the context of software reliability, but this is a global characteristic for both hardware and software components of communication systems, where new releases are permanently issued, and similar considerations may apply as well to hardware as to software.

Numerous models for reliability increase have been proposed: Jelinski-Moranda, Shooman, Musa, Littlewood-Verral, etc. Musa's model seems the best suited to the cases we have in mind.

Let N_0 be the number of errors (defaults) remaining at $t = 0$, and $n(\tau)$ the number of errors corrected at date $t = \tau$, then we have:

$$\lambda(\tau) = K(N_0 - n(\tau)), \tag{6-42}$$

where K is a proportionality constant which relates error exposure frequency to code linear execution frequency (no error can occur as long as no code is executed, as already explained).

Assuming that no new errors are spawned in the error correction process, the error correction rate will be equal to the error exposure rate. Then one has:

$$\frac{dn}{d\tau} = \lambda(\tau), \tag{6-43}$$

which combined with the relationship above yields:

$$n(\tau) = N_0(1 - e^{-K\tau}), \tag{6-44}$$

and the MTTF (*mean time to failure*), average time up to the next fault occurrence (following the default exposure), is:

$$T = \frac{1}{\lambda(\tau)},$$

$$T = \frac{1}{KN_0}\exp(K\tau), \text{ or introducing } T_0 = \frac{1}{KN_0},$$

$$T = T_0 \exp(\frac{\tau}{N_0 T_0}). \tag{6-45}$$

Obviously, the MTTF increases with time, up to the useful lifetime period where it is approximately constant, with however a certain amount of variability (related, as explained, to the introduction of new releases).

This expression allows estimating practically the duration of the early-life period and the failure risks during this period.

For instance, the MTTF is multiplied by 3 after a time τ of the order of N_0T_0 (i.e. N_0 times the initial MTTF when the equipment is brought into service) and by 10 for a period 2.3 times longer, in accordance with the exponential distribution. Practise shows that often a new system reaches its operational (useful life) period when its initial MTTF has been mutiplied by 3. If one denotes as θ the MTBF (now equal to the MTTF) of the equipment during its useful-life period, the early-life duration is then $\tau = (N_0/3)\theta$. This emphasizes the impact the residual errors have in the system reliability.

6.5.2. *Useful-life period model*

Here the goal is to measure the impact of failures due to residual software errors on the system availability. During the useful-life period, the software failure rate is assumed as being constant: the system is assumed stable, the last version has been delivered and the early-life defaults have been corrected.

The effect of the failures related with software takes an increasing importance. However, most of the residual defaults lead to failures quite easy to recover, thanks to the defence mechanisms such as "hot restarts". To illustrate this, let us analyse on a simple example the relative influence of the three categories of failures: software defaults said *recoverable* simply by restarting the failed machine, *severe* defaults, giving rise to a full system reinitialization using remote elements, and lastly hardware failures asking for human intervention.

Consider a system composed of two elements in passive redundancy. The master component can break down, due to hardware of software failure. The passive element cannot breakdown, except due to hardware cause, as it executes no code (or hardly no: for instance a test routine).

For software failures, one distinguishes the ones recoverable through an automatic system restart, from the other ones, said "severe", asking for human intervention.

The two components being in operation, upon a recoverable software fault on the active element, the system switches the control to the standby one, which becomes active. Then, quasi instantaneously, the system comes back to normal operation (two elements in order of service), the faulty one being rapidly and automatically reloaded (by copy from the other for instance). This transition is absolutely negligible from an unavailability point of view.

If after a switching on the standby element due to a non recoverable or an hardware fault (ie. with only one element into service) a recoverable software default is detected, the system (out of service) is *re-initialised* by automatically reloading from a remote site. The duration of this operation is non-negligible but still moderate as compared to human intervention in case of equipment replacement.

Let us denote:

λ_h the hardware failure rate,

λ_{ss} the "severe" software failure rate, i.e. which cannot be repaired by an automatic restart of the station (this is equivalent to an hardware failure).

λ_{sr} the "recoverable" software failure rate: failures recoverable through automatic restarts of the station. This operation is almost instantaneous if a copy from the other element is possible; otherwise, the re-initialisation lasts a few minutes.

μ_h the hardware repair rate, corresponding to an intervention duration of a few hours,

μ_r the reinitialization rate, corresponding to a reinitialization duration of a few minutes.

Let us take:

$$\lambda_1 = \lambda_h + \lambda_{ss}$$

$$\lambda_2 = 2\lambda_h + \lambda_{ss}$$

$$\lambda_3 = \lambda_{sr}\,.$$

The state diagram is represented in Figure 6.15.

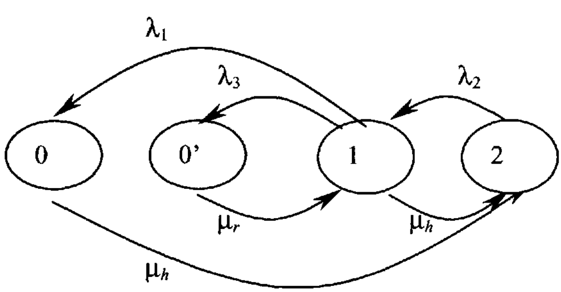


Figure 6.15. *Redundancy with software failures*

The set of equations is:

$$\lambda_2 P_2 = \mu_h (P_1 + P_0)\,,$$

$(\lambda_1+\mu_h)P_1=\lambda_2P_2+\mu_rP_0$,

$\mu_hP_0=\lambda_1P_1$,

$\mu_rP_{0'}=\lambda_3P_1$,

$P_0+P_{0'}+P_1+P_2=1$.

From the above we derive:

$$P_0=\frac{\dfrac{\lambda_1}{\mu_h}}{1+\dfrac{\lambda_1}{\mu_h}+\dfrac{\lambda_3}{\mu_r}+\dfrac{\lambda_1+\mu_h}{\lambda_2}}. \qquad (6\text{-}46)$$

$$P_{0'}=\frac{\dfrac{\lambda_3}{\mu_r}}{1+\dfrac{\lambda_1}{\mu_h}+\dfrac{\lambda_3}{\mu_r}+\dfrac{\lambda_1+\mu_h}{\lambda_2}}. \qquad (6\text{-}47)$$

Expressions that show clearly the importance of the "recoverable" software failures (terms in the order of λ_3/μ_r).

Indeed the availability of the system is:

$A=1-(P_0+P_{0'})$,

$$A=\frac{1+\dfrac{\lambda_1+\mu_h}{\lambda_2}}{1+\dfrac{\lambda_1}{\mu_h}+\dfrac{\lambda_3}{\mu_r}+\dfrac{\lambda_1+\mu_h}{\lambda_2}},$$

which may also be written:

$$A=\frac{(\lambda_2+\lambda_1)\mu_h\mu_r+\mu_h^2\mu_r}{(\lambda_2+\lambda_1)\mu_h\mu_r+\mu_h^2\mu_r+\lambda_2\lambda_1\mu_r+\lambda_3\lambda_2\mu_h}. \qquad (6\text{-}48)$$

One can first verify that, suppressing the possibility of software faults ($\lambda_3=0$, and $\lambda_2=2\lambda_1$), one finds again the basic system.

Then using the unavailability eases visualising the impact of the software failure rate:

$$U = \frac{\lambda_3\lambda_2\mu_h + \lambda_2\lambda_1\mu_r}{(\lambda_2+\lambda_1)\mu_h\mu_r + \mu_m^2\mu_r + \lambda_2\lambda_1\mu_r + \lambda_3\lambda_2\mu_h}. \qquad (6\text{-}49)$$

When severe software failures are the majority and the recoverable software failures are negligible ($\lambda_3 = 0$), one has:

$$U \approx \frac{\lambda_2\lambda_1}{\mu_h^2},$$

the system behaving just as if the hardware failure rate is worsened.

When severe software failures are negligible, which is the case if the defence mechanisms operate correctly:

$\lambda_2 = 2\lambda_1 = 2\lambda_m,$

and so:

$$U \approx \frac{2\lambda_h^2}{\mu_h^2} + \frac{2\lambda_h\lambda_{sr}}{\mu_h\mu_r}$$

The impact of re-initialisations is relatively moderate, as μ_r is much lower than μ_h. For instance, with a reinitialisation duration 10 times shorter than the human intervention, and software failure rates twice as large as hardware failure rates, the unavailability is increased by 20% only. This clearly shows the importance of implementing efficient defence mechanisms.

6.6. Spare parts calculation

This section is devoted to evaluating the quantity of spare parts to be provided for maintainance, in order to guarantee a given *stock out probability* (shortage). This is another important component of quality of service, as it contributes to quantify maintenance costs, for the operator. Remark also that shortage of the stock can severely impact service availability, especially if the *restocking delays* turn out to be non negligible (a point which must absolutely be checked).

6.6.1. *Definitions*

Call N the total number of equipments in the district to be secured by the spare parts store. Only the case of identical components is considered. Let P_s denote the stock out probability (probability of shortage of the stock), λ the failure rate of the component and μ the restocking rate (this rate corresponds to the time needed by the repair centre, local or central, to replace the missing equipments into the spare parts store, also called *turn around time*).

Two major strategies can be described. The first one consists in periodically restocking the store with a period T (generally large, e.g. one year). This corresponds to machines far from production or repair centres (for instance in a foreign country). The second strategy works by continuously restocking for each item in the stock. In that case one introduces the *restocking rate* μ. This fits to situations where the district has a repair centre in its vicinity, or it can be a first-level store, making use itself of a centralised stock of second level.

6.6.2. *Periodical restocking*

The stock being of size S, the stock out probability (or shortage) is the probability of having more than S failures during the period T. We assume the replacement delay on site much shorter than the restocking delay. In this case, as long as no shortage is observed, the failure rate on the district is equal to $N\lambda$, and it gives the decrease rate of the stock size.

Thus the Poisson law with parameter $N\lambda T$, gives the answer:

$$P_s = 1 - \sum_{k=0}^{S} \frac{(N\lambda T)^k}{k!} e^{-N\lambda T} \qquad (6\text{-}50)$$

One evaluate then the (integer) value for S allowing reaching the targeted value for P_s.

6.6.3. *Continuous restocking*

The system is composed of the equipments on site *and* those in the stock. Its behaviour is described by the transition diagram in Figure 6.16. It is a birth-death process with constant failure rate as long as there exists a spare part in the stock, and a decreasing failure rate as soon as the S spare parts of the stock have been used.

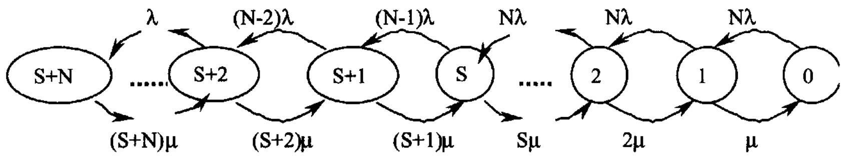


Figure 6.16. *Continuous restocking*

From the graph we obtain:

$$P_k = \frac{\left(\frac{N\lambda}{\mu}\right)^k}{k!} P_0, \qquad k \le S,$$

$$P_k = \frac{N!}{(N+S-k)!} \frac{S!}{k!} \left(\frac{\lambda}{\mu}\right)^{(k-S)} P_s \text{ for } S \le k \le N+S,$$

Then doing:
$\alpha = N$ for $k \le S$, and $\alpha = N+S-k$ for $k > S$,
we have the general expression:

$$P_k = \frac{N!}{\alpha!} N^{(\alpha-N+k)} \frac{\left(\frac{\lambda}{\mu}\right)^k}{k!} P_0, \text{ where } \begin{array}{l} \alpha = N, k \le S \\ \alpha = N+S-k, k > S \end{array} \tag{6-51}$$

with of course:

$$\sum_{k=0}^{N+S} P_k = 1 \text{ and } P_0 = 1 - \sum_{k=1}^{N+S} P_k .$$

The stock out probability is then defined as the long-term proportion of requests for replacements that are not immediately satisfied due to a shortage of spare parts in the stock:

$$P_s = \frac{\sum_{k=S}^{N+S} \alpha P_k}{\sum_{k=0}^{N+S} \alpha P_k}. \tag{6-52}$$

with α and P_k as defined above.

The size S of the stock will be defined as the lower value of S that yields a stock out probability P_s less than the specified one.

If N is large enough compared to S the failure rate is approximately constant $\alpha = N\lambda$, and the system may be modeled by a Poisson law with parameter $\dfrac{N\lambda}{\mu}$.

$$P_s \approx \sum_{k=S}^{N+S} P_k \approx \sum_{k=S}^{\infty} P_k , \tag{6-53}$$

and then:

$$P_k = \frac{\left(\dfrac{N\lambda}{\mu}\right)^k}{k!} e^{-\frac{N\lambda}{\mu}} . \tag{6-54}$$

An application of these formulae will be considered in Chapter 10.

This page intentionally left blank

7

Queueing Theory

In this chapter, we set out the main issues of *queueing theory* (or the theory of queueing networks). Stemming from probability theory, as with reliability and statistics already presented, its application to the field of telecommunications for the solving of traffic problems has given rise to a well-known discipline: *teletraffic*.

Queueing theory sets out to take account of the phenomena of waiting and congestion that are linked to the random and unforeseeable nature of the events encountered (in the case with which we are concerned, connection set-up requests, IP packets going through a router, tasks to be processed by the processor of a server, etc.). In view of the problems to be resolved as set out in the chapter on quality of service (waiting times to be complied with, loss probability in buffers, etc.), the importance of this theory will easily be understood from the viewpoint of our essential aim: to assess the performances of systems and networks.

Whenever a theory is proposed, it offers a schematic and simplified vision of observed reality. In point of fact, this theory will be used on models, abstractions from the actual world which capture the essence of the phenomenon to be studied. The activity of *modelling*, which leads from the actual system to the queueing model, takes on essential importance in this case.

Obviously, the theory does not take into account all the diversity of reality – or more accurately is not able to respond to it with perfect fidelity. Depending on the case in hand, we will have to make use of approximate results or sophisticated methods. In the most complex cases, we will have to associate with the theory methods such as simulation, which we will consider in Chapter 8.

7.1. The basic service station: clients and servers

Throughout this chapter, we will be dealing with two basic concepts: *clients (customers)* and *servers*. In queueing theory, the *clients* are entities that move in a network of *servers*, where they receive processing. When several clients try to obtain a service simultaneously, some of them must wait in *queues*, or *buffers*. This vocabulary is conventional and general. Clients may be people queueing up at a ticket office to obtain a ticket (the server is the employee issuing the tickets). In the field with which we are concerned, they may represent packets that are forwarded to a transmission line and awaiting the availability of the line (two packets may be competing). The server is then the transmission link, and the service time is the transmission time. They may also be programs (or "tasks") requesting processing in a real-time system.

Another type of client consists of requests for circuits, or bandwidth, in a telephone system. In this case, the servers form a group: the circuits or the links serve the direction requested.

The most general service station consists of a queue, of finite or non-finite capacity, emptied by one or more servers, and supplied by a flow of clients (see Figure 7.1).

To describe this system with greater precision, it must be possible to specify:

- The mechanism according to which clients arrive in the buffer (i.e. what probability law the arrival process does comply with);
- Service time (i.e. the probability distribution of its duration);
- Service discipline (when a server becomes idle, which client does it choose?).

7.2. Arrival process

To advance in the study of traffic properties, it is necessary to look at the two components on which it depends, i.e. client arrivals and their service.

Arrivals of clients at the system input are observed. To describe the phenomenon, the arrival law, the first idea is to use the time interval between successive arrivals, or the number of arrivals in a given time interval.

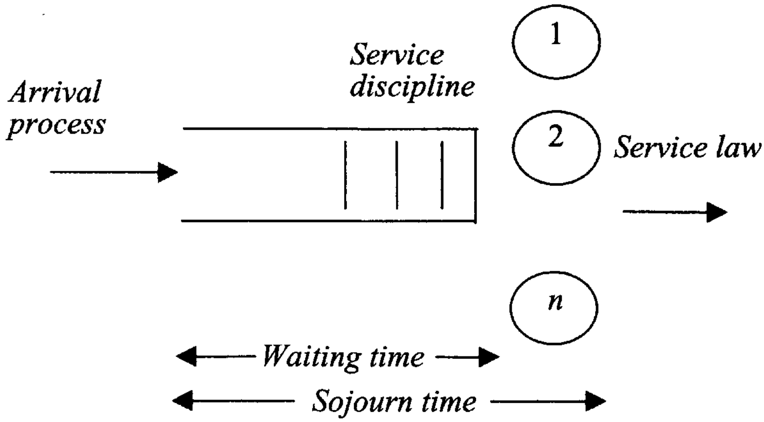


Figure 7.1. *The basic service station*

During the time T, $n(T)$ arrivals occur. The flow intensity is then expressed as a number, the arrival rate, whose intuitive definition is:

$$\lambda = \lim_{T\to\infty} \frac{n(T)}{T}$$

It is then also possible to estimate the average interval between arrivals, this being the inverse of the previous quantity.

7.2.1. *Renewal process*

In the most favourable case, the statistical description of the time between arrivals is very simple. Let us assume that the situation can be described as follows:

Each time an arrival takes place, the interval until the next arrival is drawn according to a given law, in such a way that successive intervals are independent of each other. This is in fact the special case of a *renewal process*. It is essential to understand the value of this concept, but also what is rare about it. For example (in Figure 7.2) let us observe the arrival process resulting from the superimposition of two independent renewal processes.

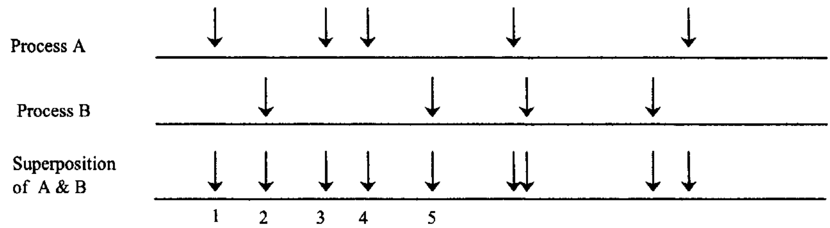


Figure 7.2. *Superimposing processes A and B does not result in a renewal process: the arrival of "4" is related to "3", but "5" is related to "2"*

Clearly, this is a very common situation, but the superimposition process *is not a renewal process.* To predict the arrival of the 4th client, reference must be made to the 3rd, by using the inter-arrival time. But the arrival of the 5th client can only be predicted by reference to the 2nd client, and not to the 4th.

7.2.2. *Poisson process*

Let us assume that the arrival process complies with the following rules:

- the probability of an arrival in an interval $[t, t+\Delta t[$ does not depend on what happened before the instant t. This is the so-called memoryless property.
- the probability of the arrival of a client is proportional to Δt, and the probability of more than one event is "negligible" (in the upper order of the infinitely small, in mathematical language). The proportionality factor is rated λ (process intensity).

These are the classical assumptions that lead to the Poisson process. How can the distribution of probability of the number of arrivals in a given time be estimated on the basis of the above axioms? The reasoning applied is typical of the theory:

Let us denote as $P_k(t)$ the probability of k arrivals in the interval $[0, t[$. One can describe how this probability varies over time: k clients will be observed in the interval $[0, t+\Delta t[$ if:

- k clients have been observed in $[0, t[$, and no arrivals have been observed in $[t, t+\Delta t[$;
- $k-1$ clients have been observed in $[0, t[$, and one arrival occurred in $[t, t+\Delta t[$;

- $k-n, n>1$ arrivals have been observed in $[0,t[$, and n arrivals in $[t, t+\Delta t[$, etc.

If these observations are put into an equation:

$$P_k(t+\Delta t) = P_k(t)\left[1-\lambda\Delta t + o(\Delta t)\right] + P_{k-1}(t)\left[\lambda\Delta t + o(\Delta t)\right]$$
$$+ P_{k-2}(t)\left[\lambda\Delta t + o(\Delta t)\right]^2 + \ldots$$

If $k = 0$, the terms $k - 1$, etc., disappear:

$$P_0(t+\Delta t) = P_0(t)\left[1-\lambda\Delta t + o(\Delta t)\right]$$

The development of the previous equation (making $\Delta t \to 0$, etc.) leads to:

$$\frac{d}{dt}P_0(t) = -\lambda P_0(t),$$

that is $P_0(t) = ae^{-\lambda t}$, where a is a constant, still unknown at this point.

The general equation will be similarly developed. It should be noted that in the passage to the limit that leads to the derivative (that is, $\Delta t \to 0$), the terms in $o(\Delta t)/\Delta t$ disappear:

$$P_k(t) + \frac{d}{dt}P_k(t)\Delta t = P_k(t) - \lambda P_k(t)\Delta t + \lambda P_{k-1}(t)\Delta t + o(\Delta t),$$

thus $\dfrac{d}{dt}P_k(t) = -\lambda P_k(t) + \lambda P_{k-1}(t)$.

The basic resolution method proceeds using a recurrence:

$\dfrac{d}{dt}P_1(t) = -\lambda P_1(t) + \lambda ae^{-\lambda t}$, and thus $P_1(t) = \lambda ate^{-\lambda t} + be^{-\lambda t}$

The obvious condition $P_1(0) = 0$ leads to $b = 0$; the reader will be able to write the second iteration, which gives the intuition of the general solution: $P_k(t) = a\dfrac{(\lambda t)^k}{k!}e^{-\lambda t}$. This can be verified with the general equation. It can now be noted that, as a certain number of arrivals have taken place in the interval, it is essential that:

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

which gives $a = 1$. Finally, the probability of observing k arrivals in an interval of length t amounts to:

$$P_k(t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t} \tag{7-1}$$

The above discrete distribution is the Poisson law, which is sometimes written, by noting $A = \lambda t$:

$$P(X = k) = \frac{(A)^k}{k!} e^{-A}$$

A is then the mean traffic offered during the period considered.

The distribution function of the probability law for the interval for two successive arrivals is derived from the distribution: the probability of an interval between arrivals greater than t is the probability that no arrival occurs between 0 and t:

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

The average number of arrivals observed in any interval of length t is:

$$m = \lambda t$$

and its variance is also

$\sigma^2 = \lambda t$ (see also Chapter 3).

It should be noted that the Poisson process is a renewal process.

7.2.3. *Where do the Poisson processes come into play?*

When one knows nothing about an arrival process, it is tempting to attribute to it the two properties set out above, which lead to the formulae of the POISSON process, as they are both generally applicable and reasonable for many systems. In point of fact this is not the case – and verifying these hypotheses will usually be a difficult process. In most cases, the justification for using the Poissonian hypothesis is based on the following result.

Theorem [FEL 71, CIN 75].– Let us assume k independent renewal processes, not necessarily Poissonian, with the arrival rates $(\lambda_i, i=1,\ldots,k)$. We denote as $\lambda = \sum \lambda_i$ the global arrival rate of the process resulting from their superimposition.

If the previous sum has a limit of λ^* when k increases indefinitely, then the superimposition process tends towards a Poisson process with the rate λ^*.

(Note: the use of the notation λ_i and of the concept of arrival rate does not in any way mean that a Poissonian hypothesis is adopted for individual processes.)

7.3. Service process

The service process may be extremely complex, but one usually limits oneself to assuming that each service time is independent of the others, and that all of them comply with the same distribution function: we speak of independent and identically distributed variables. This law, the service law, is described by its probability distribution:

$$B(x) = P\{\text{Service time} \leq x\}$$

The assumption about the independence and identity of distribution between successive services, although a simplification, in fact represents a very good approximation in most actual cases, such as calls or independent sessions times, etc. We will however need to deal with systems handling clients whose service laws obey different distributions. This is the case for example with multi-service systems, in which each type of service has its own characteristic.

We will give some examples of classical service laws. First, let us introduce the important concept of residual service time.

7.3.1. *Residual service time*

This concept will enable us to make explicit the "memoryless" property of some processes. The following question is asked: if a service is observed which has already reached the "age" y, what is the distribution of the time X still to elapse up to the end of the service?

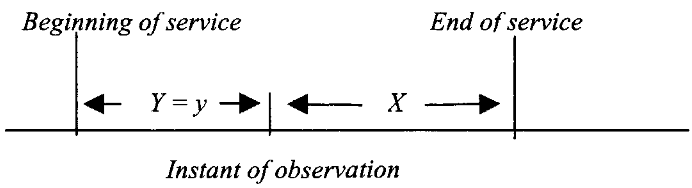


Figure 7.3. *Distribution of the residual service time*

The answer is very simple: saying that $X > x$ seconds are still to be accomplished is the same thing as saying that the total time will be greater than $y+x$, and as $Y=y$ this corresponds to calculating the conditional probability *knowing that* the total time is greater than y. This gives for the distribution of X:

$$\begin{aligned} P(X > x / Y = y) &= P(whole\ service > x + y / Y > y) \\ &= \frac{P(whole\ service > x + y)}{P(whole\ service > y)} \\ &= \frac{1 - B(x+y)}{1 - B(y)} \end{aligned} \qquad (7\text{-}2)$$

In an equivalent way, the density will be:

$$P(X \in [x, x+dx[/ Y) = b(x+y)dx/(1-B(y))$$

We will see an application of this below.

7.3.2. *The exponential law*

The most popular service law is the exponential law, which is traditionally written by using the letter μ as the "service rate":

$$B(x) = P(service \le x) = 1 - e^{-\mu x},\ b(x) = \mu e^{-\mu x} \qquad (7\text{-}3)$$

The exponential distribution owes a great deal of its prestige to its "memoryless" property, which has already been indicated, and which may be stated as: “Knowing that the service has already lasted for 1 hour provides no information about how soon it will end”.

It will be noted indeed that, for the exponential law, the conditional event density (formula (7-2) above), is:

$$b(x/y)=\frac{\mu e^{\mu(x+y)}}{e^{-\mu y}}=\mu e^{-\mu x}=b(x)\text{, independently of } y.$$

The application of this “memoryless” property shows that the probability of an end of service in the coming instant (in the interval $[t,t+dt[$) is μdt, whatever the age of the service.

This result (mentioned in the literature under the term “taxi paradox” or “bus paradox”) has an apparently paradoxical characteristic. The distribution of the remaining time is identical to the initial distribution, for an exponential law, and thus also its mean value. It can also be shown that “age” (time Y in the diagram) has the same distribution and the same average. This means that the interval observed has duration double that of the mean value of the law!

The paradox can be dissipated when it is realised that the observation of the interval introduces a bias. Let us take the example of a colleague who wants to talk to you when you are taking part in a phone call. When he arrives in your office “by chance”, the risk is greater that he will arrive during a long conversation (as this interval is long). This colleague thus “samples”, without wishing it, samples of an exponential law which are biased: they have an average length that is greater; double in the case in hand. The fact remains that his/her “common sense” is somewhat disrupted.

The same exercise, reproduced with a service of constant length, would give a result which is more in line with naive intuition: remaining duration is half of total length. In this case the sampling process does not influence the selected times, which are all identical!

The exponential law is characterised by its moments:

Mean of variable: $m=1/\mu$,

Variance of variable: $\sigma^2=1/\mu^2$.

Its *coefficient of variation* i.e. the ratio of standard deviation of service time to its mean value is: $c = \sigma / m$. Here, the coefficient of variation is 1.

7.3.3. *Erlang laws*

Let us assume that the service process is composed of a cascade of k elementary exponential servers, which are identical (i.e. have the same μ parameter) and independent of each other. The service time is the sum of the times spent in each server. There is indeed only one server, with k stages (only one client at a time is authorised to enter the server).

The service time is distributed as the sum of k exponential variables, which are independent and with the same parameter. This case has already been encountered in Chapter 4: it is an Erlang-k law whose form may be again stated as follows:

$$B(x) = P(X_1 + X_2 + ... + X_k \leq x) = 1 - e^{-\mu x} \sum_{j=0}^{k-1} \frac{(\mu x)^j}{j!} \tag{7-4}$$

As this is a sum of independent random variables, the mean and variance are easily obtained, as the sum of the mean and the variance of each exponential variable:

Mean of variable: k / μ,

Variance of variable: k / μ^2,

The coefficient of variation is: $1/\sqrt{k}$.

We should recall the property of this law: its coefficient of variation $c = 1/\sqrt{k}$ is always less than 1, and it is used as a system type in which the service time is less variable than the exponential. The limit case will be, for a very large k, a constant service time.

7.3.4. *Hyperexponential law*

Another interesting configuration is that of the hyperexponential law. The "service" mechanism consists of two exponential servers with different rates, between which the client chooses (Figure 7.4).

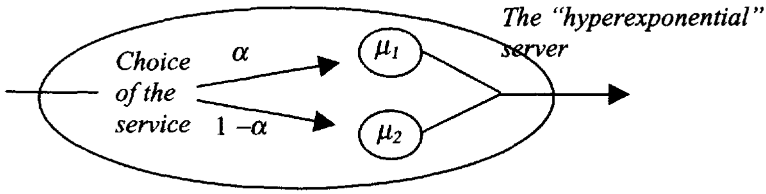


Figure 7.4. *The hyperexponential server*

On entering the server, the client "chooses" service 1 (average duration $1/\mu_1$) with probability α, and server 2 with probability $1-\alpha$. Only one client at a time resides in the server. For example, the client is a task in a machine, which according to a given criterion will run either one programme or another. The situation can be enriched on the same principle, by proposing a choice between n servers. The corresponding distribution has also been presented in Chapter 4. Let us recall the form for 2 servers:

$$P(Service \le x) = \alpha\left(1-e^{-\mu_1 x}\right) + (1-\alpha)\left(1-e^{-\mu_2 x}\right)$$

The calculation of the mean and variance gives the following result:

Mean: $E(s) = \dfrac{\alpha}{\mu_1} + \dfrac{1-\alpha}{\mu_2}$,

Coefficient of variation: $c^2 = \dfrac{2\left[\dfrac{\alpha}{\mu_1^2} + \dfrac{1-\alpha}{\mu_2^2}\right]}{\left[\dfrac{\alpha}{\mu_1} + \dfrac{1-\alpha}{\mu_2}\right]^2} - 1$

This factor is always greater than 1: a "supervariant" law such as standard deviation, which is stronger than the mean, may be represented by a hyperexponential law. The couple (Erlang-k, hyperexponential) provides a means of approximating any law whose mean and variance are known. We may recall the generalisation in the form of the "Cox laws" which enable the representation of any probability distribution of a service law by a combination of exponential servers.

The example with two components is easily generalised to the case of a choice between n exponential services (see also Chapter 4).

7.4. Birth and death process

7.4.1. *Notion of "state"*

The state concept has an intuitive basis: describing the state of a system means giving a list of the characteristics it possesses, and also providing elements which enable predictions about how it will change. In mechanics for example, a point particle is described by its coordinates (x, y, z) and by its velocity (v_x, v_y, v_z). Once these quantities are known, it is possible to describe the position of the point, and *via* the equations of mechanics and given the initial state, the trajectory it must follow.

The same is true of a stochastic system, except that the forecast of the future takes on a probabilistic character: it is not possible to predict the exact state within a few seconds, but only its probability distribution.

7.4.2. *Markov chains*

Let us consider a random variable *X*, with discrete state space (it only takes its values in a discrete space). *X* may represent the number of clients in a buffer: $X = 0,1,2...$ These states may be noted: E_1, E_2, ..., E_k, ... Let us note as $t_1, t_2, ..., t_n$ the successive instants of state changes of *X*, and as $x_1, ..., x_n$ the sequence of the states visited. The t_n could for example be the instants of arrival in or departure from the buffer. In general, the study of the evolution of *X* is very difficult. There is however one capital circumstance, which is the basis of all the following developments:

In the general case, the value of *X* at instant t_{n+1} depends on the whole of the past, that is on $X(t_1), X(t_2)$, etc. It may be said that *X* is obeying the *Markov property* if:

$$P[X(t_{n+1}) = x_{n+1} / X(t_n) = x_n, X(t_{n-1}) = x_{n-1}, ...] = P[X(t_{n+1}) = x_{n+1} / X(t_n) = x_n]$$

In short, *only* the current state at instant t_n influences the future evolution of the process, and the earlier states may be forgotten. This is another form of the *memoryless property.*

Let us also assume that the process is *homogeneous*, i.e. invariable by translation over time. This enables the introduction of the notation:

$$p_{j,k} = P[X(t_{n+1} = E_k / X(t_n)] = E_j$$

(without homogeneity, it would have been necessary to introduce a notation of the type $p_{j,k}^{(n)}$). The Markov property gives:

$$P[X(t_{n+1}) = E_k] = \sum_j p_{j,k} P[X(t_n) = E_j] \tag{7-5}$$

This is the elementary form of a fundamental result, known under the name of *Chapman-Kolmogorov equation.* The process studied is a *Markov chain.*

7.4.3. *Birth and death processes*

The study of the most general Markov processes is one of the main concerns of probability theory. Considering the very specific needs of this section we will limit our interest to the very special, and very useful, case in which the process can only make "jumps" to its direct neighbours. And for simplification and without loss of generality we will say that the state of the system can change from E_k to E_{k-1} or E_{k+1}. The rate of passage from E_k to E_{k+1} is conventionally noted λ_k : this is the *birth rate*. The rate of passage from E_k to E_{k-1} is the *death rate* μ_k. The birth and death process concept has been introduced into the study of population trends, where the state refers to the size of the population. In agreement with this origin, and to simplify the writing, the states will be named simply $1,2,3,...,k$, with no loss of generality.

The problem is simple: we are aiming to derive the probability distribution of the state, i.e. the function $P_k(t) = P[X(t) = k]$, with the initial state being supposed to be known.

Establishing the equation is relatively intuitive. In fact, the aim is simply to rediscover the dynamics of the Chapman-Kolmogorov equations. Let us examine the possible evolutions from instant *t*. At instant $t + \Delta t$, our process will be observed in state *k* if:

- At instant *t*, the state was *k* <u>and</u> nothing happened in the elementary interval $[t, t + \Delta t]$.
- At *t*, the state was *k–1* <u>and</u> a "birth" occurred during the interval $[t, t + \Delta t]$.
- At *t*, the state was *k+1* <u>and</u> a "death" occurred during the interval $[t, t + \Delta t]$.

The interval $t+\Delta t$ is assumed to be "small". Other more complex events (2 arrivals, one arrival and one departure) will take place with probabilities in $o(\Delta t)$. We will therefore write:

$$\begin{aligned}P_k(t+\Delta t) = P_k(t)[1-\lambda_k\Delta t - \mu_k\Delta t] \\ +P_{k-1}\lambda_{k-1}\Delta t \\ +P_{k+1}\mu_{k+1}\Delta t \\ +o(\Delta t)\end{aligned}$$

This equation accurately transcribes the description of the possibilities given above. Based on this form, we write a differential equation by classical analysis processing:

$$\frac{P_k(t+\Delta t)-P_k(t)}{\Delta t} = -(\lambda_k+\mu_k)P_k(t)+\lambda_{k-1}P_{k-1}(t)+\mu_{k+1}P_{k+1}(t)+\frac{o(\Delta t)}{\Delta t}$$

One special case: at 0 the "death" transition will not exist, as there is no state noted "-1". Finally, the passage to the limit leads to the fundamental differential system:

$$\begin{aligned}&\frac{d}{dt}P_k(t) = -(\lambda_k+\mu_k)P_k(t)+\lambda_{k-1}P_{k-1}(t)+\mu_{k+1}P_{k+1}(t),\ k>0 \\ &\frac{d}{dt}P_0(t) = -\lambda_0 P_0(t)+\mu_1 P_1(t)\end{aligned} \tag{7-6}$$

This equation takes us back to the analogy of the mechanics of the point particle. If the initial state is known, we can predict the distribution for any instant in the future. The necessity of knowing the initial state seems very difficult! Fortunately, there exists a class of systems, so that it is not necessary to know this initial state. The analysis of this process leads to a distinction being drawn between two cases:

– Case 1. When the time increases, the probabilities tend towards a limit that is independent of the initial state. We then speak of systems that have reached their statistical equilibrium – or their steady-state condition.

– Case 2. No equilibrium is achieved, the $P_k(t)$ have no asymptote. This will in particular be the case of overloaded systems (intuitively, for such a system, the number of clients present increases indefinitely, no stationary condition could exist). This configuration is of course entirely theoretical: in practice, congestion will result in the abandonment of requests, and in general by a modification of the behaviour of the whole system.

We will limit ourselves to the study of the systems in Case 1, and we will study the equilibrium state. For a dynamic system at equilibrium, a *steady-state condition* exists, such that the state probabilities are equal to the asymptotic value and thus do not depend on time. We will simply rewrite the above system, cancelling all the derivatives. The stationary distribution will be denoted P_k.

$$\begin{aligned}(\lambda_k + \mu_k)P_k &= \lambda_{k-1}P_{k-1} + \mu_{k+1}P_{k+1} \\ \lambda_0 P_0 &= \mu_1 P_1\end{aligned} \qquad (7\text{-}7)$$

This system is resolved by recurrence: P_1 as a function of P_0, and then P_2 as a function of P_0, P_1, and so on. The calculation is completed by the *normalisation condition*, which simply stipulates that the system must exist somewhere:

$$\sum_k P_k = 1$$

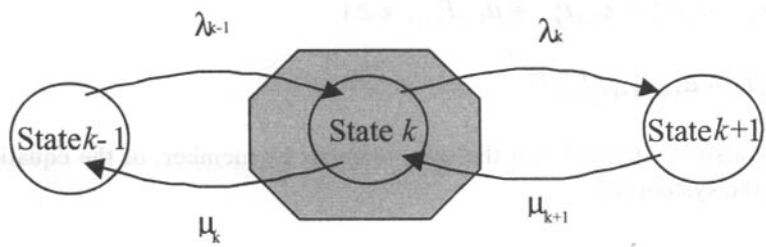


Figure 7.5. *Evolution of the state of the birth-and-death process*

The set of equations (7-7) is open to a very fruitful interpretation, illustrated in Figure 7.5. The left hand term $(\lambda_k + \mu_k)P_k$ can be interpreted as a probability flow that is leaving state k. A term such as $\lambda_k P_k$ measures the probability flow rising from k to $k+1$. This flow is the product of the probability of reaching the state by the rising departure rate (knowing that one is there). The same interpretation applies to the other terms. The equation says that, for the probabilities to retain constant values, the incoming flows and the outgoing flows of the various states must be equal.

This interpretation is used to write the equations very rapidly, from a diagram similar to that in the figure above. A particularly effective method consists of noticing that the "principle of conservation" applies for all groups of states. In particular, the outgoing flow of probability of the states 0, 1, 2, ... k, i.e. the outgoing flow of k, must be equal to the flow that enters coming from the state k+1: Figure 7.6.

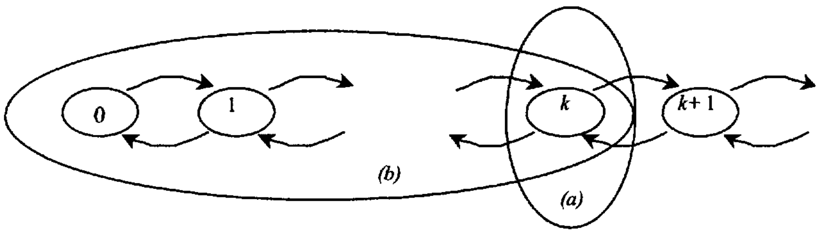


Figure 7.6. *Conservation flow around boundaries (a) or (b) allows writing state equations*

The application of approach (b) is extremely effective. The putting into equations are compared:

$$(a)\quad \begin{aligned} &\lambda_0 P_0 = \mu_1 P_1 \\ &(\lambda_k + \mu_k)P_k = \lambda_{k-1}P_{k-1} + \mu_{k+1}P_{k+1},\ k \geq 1 \end{aligned}$$

$$(b)\ \lambda_k P_k = \mu_{k+1}P_{k+1},\ k \geq 0 \qquad (7\text{-}8)$$

It may also be checked that the sum, member by member, of the equations (a) gives the system (b).

7.4.4. *PASTA property (Poisson arrivals see time averages)*

We come back here to the properties of the Poisson process. It has quite an attractive property known as the PASTA property, which is particularly useful in calculations relating to quality of service.

Imagine a system that evolves in such a way that it was possible to evaluate its state probabilities. Rigorously speaking, we will speak of *stationary state probabilities*, to indicate the fact that they do not depend on time (note the passage to the limit carried out to establish state equations, and note that the existence of stationarity is a property to be verified, as a given system will not necessarily have a stationary state).

The intuitive significance of these stationary probabilities is as follows:

The system is observed for a relatively long period Θ. The system runs through, in accordance with its logic, a series of states (noted 1, 2, 3, ...). We note the time of passage in each state, and we take the aggregate of the times spent in each of the states $\theta(n)$. Now, if the observation time is relatively long,

$$\frac{\theta(n)}{\Theta} \approx P_n \text{, and more precisely } P_n = \lim_{\Theta \to \infty} \frac{\theta(n)}{\Theta}$$

We often speak in the case of stationary probabilities of *time averages*, to recall the formula. Now, one can imagine another method of system observation. We observe the clients who arrive at the input to the system. Each client arriving will note an instantaneous state. We add up the number of clients observing the system in state 0, 1, 2, ...; thus $q(n)$ is the number of arrivals finding n clients, and N is the total number of clients taking part in the experiment. It is possible to calculate the ratio $Q_k = q(k)/N$, which is also a set of probabilities (because $0 \le Q_k \le 1, \sum Q_k = 1$) : probabilities at instants of arrival. In general, the two sets of probabilities are different. But:

if the arrivals take place in accordance with a Poisson process, then $P_k = Q_k$ *: Poisson arrivals see time averages i.e. the PASTA property.*

We will see later counter examples to this result, when the arrivals are not Poissonian, and which can be interpreted as a certain correlation between the system and the arrivals. The PASTA property signifies, or rather confirms, that the Poisson process is carrying out a "pure chance" process.

7.5. Classical queueing models

We would like to very briefly recall the main results: reference should be made to a more comprehensive queueing course for more results.

7.5.1. *Kendall notation*

To identify a queueing system, the following formal expression has been proposed and unanimously adopted:

A/B/n/K/N

The first letter identifies the arrival process distribution, the second the service process, with in both cases the following conventions:

– M: memoryless law (Poissonian arrivals or exponential service);

– D: deterministic law;

– Ek: "Erlang-k" law

– Hk: "hyperexponential" law of order k;

– GI: general law, with the successive variables being independent;

– G: general law, without any independence assumption.

The following figure gives the number of servers, the following letters (optional) identify the size of the queue, and the size of the source population (if these values are omitted, they are supposed to be infinite).

For example, M/M/1 refers to the basic model: Poissonian arrivals, exponential service, a single server, non-limited queue length. M/D/K/K indicates a constant duration service, K servers and K places in total (i.e. no waiting: Erlang model, see below).

In a queue, a parameter of prime importance is the *utilisation factor*, which is traditionally noted ρ. This is the product of the client arrival rate by the service time: $\rho = \lambda E(s)$.

It can be shown that $1-\rho$ gives the probability of finding the server inactive. It is necessary (for a queue of infinite capacity) that $\rho < 1$ (stability condition: intuitively, that the work must enter the system more slowly than the system can handle it).

7.5.2. *General results*

"Resolving" a queueing system consists, depending on the system parameters (arrival rate, service rate), of writing the distribution of the number of clients in the system and the waiting time – or, at least, of forecasting the mean values of these quantities. Some results of general significance, independent of the laws of the processes used, can be mentioned:

Parameter ρ is the utilisation factor i.e. the server load, that is also the traffic offered (in a stationary regime). The stationary probability that the server is inactive is $P_0 = 1 - \rho$ (see the discussion of the concepts of traffic offered and traffic carried on this matter).

The mean numbers and the waiting times are linked by the celebrated *Little formula*: $E(N_x) = \lambda_x E(W_x)$. Index x indicates that any possible interpretation can be given to these values: the average waiting time and the number of clients waiting, or the sojourn time and the number of clients in the system, etc.

This last formula is very important. An intuitive proof can be given of this, based on a very simple graphic reasoning.

Let us observe the occupation of the system (number of clients in the system – note that *no assumption* is made on the system, or on the state of the clients, waiting, in service, or otherwise). Each time a client enters or leaves, the number of individuals present is updated, and the trend in *N(t)*, the number of clients present, resembles the diagram in Figure 7.7:

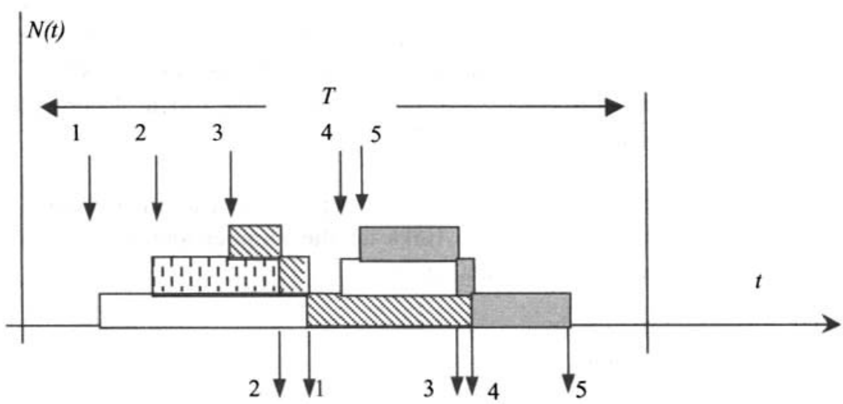


Figure 7.7. *Little formula*

To estimate the average number of clients in the system, the total surface area intercepted by *N(t)* is counted: this is the sum of the surface areas contributed by each client: $\sum D_k$, D_k being the time that client k will spend in our system. The average number will then be:

$$E(N) = \lim_{T\to\infty} \frac{\sum D_k}{T}$$

(in fact, the "mean" is traditionally defined as a limit).

Now the sum can be rewritten, by introducing the number n of clients concerned in time T:

$$\frac{\sum D_k}{T} = \frac{\sum D_k}{n(T)} \times \frac{n(T)}{T}$$

Under relatively general conditions, the limit of the product is equal to the product of the limits. The term $n(T)/T$ is simply the arrival rate of the process, it is noted λ. The first term is nothing other than the mean time that each client spends in the system. If the "system" in question is the queue in front of a server, we obtain the classical form of the *LITTLE formula*:

$$E(N) = \lambda E(W)$$

More generally, it may also be noted that, interpreting the "system" as a pool of resources, we simply find the concept of traffic in erlangs as presented at the start of the book. $E(N)$ is the mean number of clients (calls, sessions, etc.) simultaneously occupying the pool of resources, i.e. the traffic A generated by an arrival rate λ and a service time $E(T)$.

It is possible to arrive at a greater degree of particularisation of clients (in a system with priorities, the formula links up the number and waiting time of priority clients, non-priority clients, etc.).

7.5.3. *The queue M/M/1*

This is the best-known queueing model, and quite rightly so. It enables the illustration of the fundamental concepts linked to queueing for a server, as they are encountered in most of the network equipments, whether it is a matter of call processing tasks in a processor, or packets in a router.

7.5.3.1. *State probabilities*

The system is described by the arrival process, Poissonian of rate λ, the exponential service law (rate μ), and the queue of infinite capacity. Clients may be assumed to be served in the order of their arrival, if this assists the intuition, but this assumption is not in fact necessary. When $n > 0$ clients are in the system (one in service, $n-1$ queueing) at instant t, the end of service of the client in progress occurs in the interval $[t, t+\Delta t]$ with a probability of $\mu\Delta t + o(\Delta t)$: the fundamental property of the exponential service law. Similarly, an arrival will occur in the interval with a probability of $\lambda\Delta t + o(\Delta t)$. To this system is applied the formalism and results obtained earlier in the birth and death process. The state is described by n, which this time represents the total number of clients in the system. It evolves in accordance with a very simple birth and death process shown schematically in Figure 7.8.

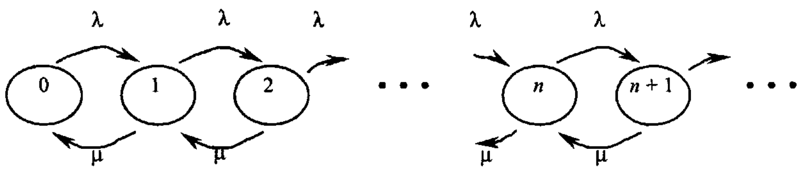


Figure 7.8. *Evolution of the state of the M/M/1 queue*

From the diagram, the following results can be deduced. This will take the form of a detailed demonstration. For the other systems presented later, the exercise will be left to the reader.

The analysis of the system, in accordance with the Chapman-Kolmogorov equations procedure, leads to the following equations:

$$\lambda P_0 = \mu P_1$$

$$\ldots$$

$$(\lambda + \mu)P_n = \lambda P_{n-1} + \mu P_{n+1}$$

The term to term summation of the equations of ranks 0 to n (approach (b) of equation (7-8)) leads to the particularly simple expression $\lambda P_n = \mu P_{n+1}$: thus $P_{n+1} = \rho P_n$, and by recurrence $P_n = \rho^n P_0$. The normalisation condition completes the processing, and finally:

$$P_n = \rho^n(1-\rho),\ n \ge 0,\ \rho = \lambda / \mu, \tag{7-9}$$

A client will wait if the server is active on his/her arrival, i.e. if the system contains one or more clients. In other words, P_W, the probability that a client will have to wait, is given by:

$$P_W = \sum_{n \geq 1} P_n = \rho$$

In fact, this expression gives the probability that an external observer will find the server busy: to be precise, the distribution of P_n is a stationary probability ("time average"), and we speak of probability of congestion over time. It is also, thanks to the PASTA property mentioned earlier, the probability observed by the client arriving, and thus the probability of having to wait.

The above expressions are such that ρ plays an important role. It will be noted that they are meaningless in the absence of the condition $\rho < 1$. This is what is called the *stability condition*, or the *ergodicity condition*. It is shown that the condition $\rho < 1$ is a necessary and sufficient condition for the existence of stationary probabilities, which is in line with intuition, as this parameter is nothing other here than traffic offered. Intuitively, $\rho > 1$ amounts to having an arrival rate λ *greater* than the service rate μ: the system could not achieve a stable state in this configuration.

The distribution of the number of clients is a geometric distribution of parameter ρ, from which are deduced the mean and the variance (see Chapter 4, we also recommend that the reader makes here the complete exercise):

$$E(n) = \frac{\rho}{1-\rho}, \ \sigma^2(n) = \frac{\rho}{(1-\rho)^2}$$

With regard to waiting time: a distinction is drawn between queueing proper, or waiting in short (time spent between arrival in the queue and the start of the service) and sojourn time (time elapsed up to departure from the system). These quantities are noted W_q (waiting in the queue) and W_s (sojourn time). In the same way, a distinction is drawn between clients in the system and clients waiting in the queue. There are 0 clients waiting if the server is inactive or if a single client is present; there are n waiting if there are n+1 in the system:

$$P(n\text{ waiting}) = \rho^{n+1}(1-\rho), \quad n \geq 1, \quad \rho = \lambda / \mu,$$
$$P(0\text{ waiting}) = (1-\rho) + \rho(1-\rho) = 1-\rho^2$$

And by operating as in the previous case, the mean number of clients waiting is obtained:

$$E(n_W) = \frac{\rho^2}{1-\rho}, \; \sigma^2(n_W) = \frac{\rho^2(1+\rho-\rho^2)}{(1-\rho)^2} \tag{7-10}$$

From this can be deduced, via the Little formula, the mean waiting and sojourn times:

$$\begin{aligned} E(Waiting) &= \overline{W_q} = \frac{E(n_W)}{\lambda} = \frac{\rho}{1-\rho} \times \frac{1}{\mu}, \\ E(Sojourn) &= \overline{W_s} = \frac{E(n)}{\lambda} = \frac{1/\mu}{1-\rho} \end{aligned} \tag{7-11}$$

In many sizing problems, the issue is the probability of exceeding a certain queueing threshold, or a certain system threshold. The probability sought is therefore $P(\geq n)$.

This is written simply as follows for the system:

$$P \geq n = \sum_{i=n}^{\infty} P_i = \sum_{i=n}^{\infty} \rho^n (1-\rho) = \rho^n \tag{7-12}$$

And for the queue (excluding service): $P(\geq n_W) = \rho^{n_W+1}$.

7.5.3.2. *Use of generating functions*

The direct resolution by substitution of the state equations is immediately comprehensible, but is sensitive as soon as the expressions become a little more complex. In this case it may be advisable to use generating functions, as presented in Chapter 3.

The generating function of the distribution $(P_k, k = 0, \ldots)$ is the function of the complex variable: $P(z) = \sum_k z^k P_k$.

It is necessary to place oneself in the domain $|z|<1$, so that the function $P(z)$ exists: this is an *analytical function* of the complex variable. Its value is linked to the use of its derivatives, which will directly give the moments of the variable (see Chapter 3).

How can these properties be used? Let us look again at the elementary case of the queue M/M/1: we have written the system of equations

$$\lambda P_0 = \mu P_1$$

$$\ldots$$

$$(\lambda+\mu)P_k = \lambda P_{k-1} + \mu P_{k+1}$$

So, from the set of equations, the form of the generating function is calculated by application of the definition: in this extremely simple example, each of the equations of rank k is multiplied by z^k, and the sum is then calculated member by member:

$$\lambda P_0 + \sum_{k\geq 1}(\lambda+\mu)P_k z^k = \mu P_1 + \sum_{k\geq 1}(\lambda P_{k-1} + \mu P_{k+1})z^k$$

$$\lambda P(z) + \mu\left(P(z) - P_0\right) = \lambda z P(z) + \frac{\mu}{z}\left(P(z) - P_0\right)$$

A simple algebraic manipulation leads to the result:

$$P(z) = \frac{\mu P_0}{\mu - \lambda z} = \frac{P_0}{1-\rho z} \tag{7-13}$$

The normalisation condition translates here as $P(1)=1$, thus $P_0 = 1-\rho$.

From this highly synthetic expression are deduced the first and second derivatives, and thus the first two moments, and the variance of the number of clients in the queue; the reader is left to find the results already given. It is also possible, by expanding in series the fraction, to find the state probabilities by identification.

In this highly simplistic case, the justification may appear to be fallacious. But we will see how, with this method, to obtain certain fundamental results in the important case of the M/G/1 queue. The references, such as [KLE 76]. should also be consulted. Finally, we should stress the fact that transforms are only being used here to resolve a previously established system of equations (as with the Laplace transforms in Chapter 6). We will see subsequently how to make use of the transforms, and more exactly the characteristic function, for the resolution of complex problems, which are impossible to describe using a system of state equations as before.

7.5.3.3. *Waiting distribution*

Even if the calculation of the mean or the variance of the waiting time is an important result, it is often essential to have the distribution of the waiting time, or at least its quantiles, particularly for sizing purposes (readers are invited to refer to the discussion of QoS parameters in Chapter 2). In the case of the M/M/1 system, it is possible to deduce the distribution of state probabilities.

Let us assume that a new client arrives while n clients are already in the system. He will have to await for n service times. As these n service times are independent and distributed in accordance with the same exponential law (even the service in progress, which according to the results of section 7.3.1 has the same exponential distribution), their total time is given by their convolution product (see Chapter 3). In point of fact we have already demonstrated that the resulting law is the Erlang-n law (special case of the Gamma law for the discrete variable).

The probability density for n service times is (cf. formula 7-7).

$$f(t) = \mu e^{-\mu t} \frac{(\mu t)^{n-1}}{(n-1)!}$$

If the waiting time in the queue is noted W , this gives:

$$P(W \le t / N = n) = \frac{\mu^n}{(n-1)!} \int_0^t x^{n-1} e^{-\mu x} dx$$

The application of the total probabilities theorem (see Chapter 3) enables us to write:

$$P(W \le t / n > 0) = \sum_{n=1}^{\infty} P(W \le t / N = n) P(n)$$

and thus:

$$P(W \le t / n > 0) = \sum_{n=1}^{\infty} \frac{\mu^n}{(n-1)!} \int_0^t x^{n-1} e^{-\mu x} dx P(n)$$

with, as seen previously, equation (7-9):

$$P(n) = (1-\rho)\rho^n = (1-\frac{\lambda}{\mu})(\frac{\lambda}{\mu})^n$$

thus:

$$P(W \le t / n > 0) = \sum_{n=1}^{\infty} \frac{\mu^n}{(n-1)!} (1-\frac{\lambda}{\mu})(\frac{\lambda}{\mu})^n \int_0^t x^{n-1} e^{-\mu x} dx$$

$$P(W \le t / n > 0) = \int_0^t \lambda e^{-\mu x} (1-\frac{\lambda}{\mu}) \sum_{n=1}^{\infty} \frac{(\lambda x)^{n-1}}{(n-1)!} dx$$

Now, we have:

$$\sum_{n=1}^{\infty} \frac{(\lambda x)^{n-1}}{(n-1)!} = e^{\lambda x}$$

Thus:

$$P(W \le t / n > 0) = \int_0^t \lambda e^{-(\mu-\lambda)x} (1-\frac{\lambda}{\mu}) dx$$

which after integration gives:

$$P(W \le t / n > 0) = \frac{\lambda}{\mu}(1-e^{-\mu(1-\frac{\lambda}{\mu})t})$$

$$= \rho(1-e^{-\mu(1-\rho)t}) = \rho(1-e^{-\frac{t}{\overline{\overline{W}}}}) \tag{7-14}$$

which is the probability of waiting for less than t, for clients who are waiting as n has been strictly taken to be greater than zero.

And thus also:

$$P(W \le t) = P(W = 0) + P(0 < W \le t) = 1 - \rho + \rho(1 - e^{-\mu(1-\rho)t}) = 1 - \rho e^{-\mu(1-\rho)t}$$

Noting that ρ represents the probability of waiting, one can also directly derive from this result the distribution of the total time spent in the system (sojourn time t_s).

At last, the following important expressions give the distributions of the waiting and sojourn times:

$$\begin{aligned} &P(W \le t) = 1 - \rho e^{-\mu(1-\rho)t} \\ &P(t_s \le t) = 1 - e^{-\mu(1-\rho)t} \end{aligned} \tag{7-15a}$$

From these expressions, the means and variances are easily obtained:

$$\begin{aligned} &E(W) = \frac{\rho}{1-\rho}\frac{1}{\mu} \quad \sigma^2(W) = \frac{2\rho - \rho^2}{(1-\rho)^2}\frac{1}{\mu^2} \\ &E(t_s) = \frac{1}{1-\rho}\frac{1}{\mu} \quad \sigma^2(t_s) = \frac{1}{(1-\rho)^2}\frac{1}{\mu^2} \end{aligned} \tag{7-15b}$$

(The reader remembers that, for the exponential distribution $F(x) = 1 - e^{-\lambda x}$, one has $E(x) = 1/\lambda$ and $E(x^2) = 1/\lambda^2$, see Chapters 3 and 4).

We find this result more directly by the Laplace transform method:

Let $F(W/n) = P(W \le t/n)$ denote the probability distribution function of the time t *spent in the system* by a new client if he finds n clients ahead of him.

If the client finds n other clients in the system, the time he resides in the system is the sum of the n service times of the n preceding clients plus his/her own service time.

The distribution of the total time spent is then given by the convolution product of the n+1 service time probability densities, as they are independent (note that a client is in the process of service, but that in view of the properties of the exponential the remaining service time is still exponential). In view of the properties already demonstrated, the Laplace transform of this convolution product is simply the product of their Laplace transform.

Each of the service times follows an exponential law. If this exponential law is of the parameter μ its Laplace transform is (see Chapter 3):

$$S^*(s) = \frac{\mu}{s+\mu}$$

The Laplace transform of $F(W/n)$ is then simply

$$T_S^*(s/n) = \left(\frac{\mu}{s+\mu}\right)^{n+1}$$

And thus for the set of possible n:

$$T_S^*(s) = \sum_{n=0}^{\infty} \left(\frac{\mu}{s+\mu}\right)^{n+1} (1-\rho)\rho^n = \frac{\mu(1-\rho)}{s+\mu(1-\rho)}$$

By inspection we find the original, for it is the classical form of an exponential function which is the probability density of the time spent in the system:

$$t_S(t) = \mu(1-\rho)e^{-\mu(1-\rho)t},$$

and thus for the distribution function:

$$T_S(t) = 1 - e^{-\mu(1-\rho)t}$$

To find the time spent waiting, it is just a matter of excluding the service time of the client himself, and thus limiting the convolution product to n and no longer to $n+1$.

We thus obtain:

$$W^*(s) = \sum_{n=0}^{\infty} \left(\frac{\mu}{s+\mu}\right)^n (1-\rho)\rho^n = 1-\rho+\rho\frac{\mu(1-\rho)}{s+\mu(1-\rho)}$$

Given that the original of $1-\rho$ is necessarily the Dirac pulse function at the origin $u_0(t)$, with the factor $1-\rho$ simply expressing the fact that the probability of not waiting is equal to the probability that the server is free and therefore that:

$$w(t) = (1-\rho)u_0(t) + \rho\mu(1-\rho)e^{-\mu(1-\rho)t}$$

and

$$W(t) = (1-\rho) + \rho(1-e^{-\mu(1-\rho)t}) = 1-\rho e^{-\mu(1-\rho)t}$$

We thus find the preceding result again.

7.5.4. *The M/M/R/R model (Erlang model)*

This is one of the most popular system model, and the first to have been studied. Let us consider a group of R servers, operated in pool mode (i.e. each of the servers can serve any of the clients presenting themselves.) The clients arrive in front of the group of servers in accordance with a Poisson process, of rate λ. On their arrival, the clients are served immediately as long as at least one server is free; if the servers are all busy, the client that arrives is rejected, and is assumed to disappear definitively. We speak of a "loss" model.

Let us imagine a system in the state $k < R$. This means that k clients are present, and occupy k servers (the others are inactive). In the coming interval Δt, each of the services in progress may end, with a probability of $\mu\Delta t + o(\Delta t)$ (the services are exponential). The probability of one and only one end of service is the sum of probabilities that each of the services ends, after deducting the probability of two ends of service or more: *event of probability at* $o(\Delta t)^2$, and thus negligible in the operation of passage to the limit.

The service rate is thus $k\mu$ in this state, while $k \le R$, and zero beyond (there can not be any state beyond the limit $k = R$). The traffic offered is noted A ($A = \lambda / \mu$) and one can define a traffic by server $\rho = A/R$. The stability condition remains $A < R$, that is $\rho < 1$.

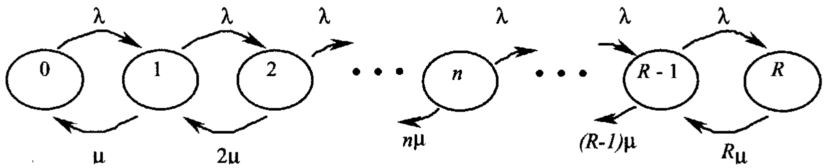


Figure 7.9. *State diagram of the Erlang model*

Based on the diagram, we find the distribution of state probabilities:

$$P_k = \frac{\frac{A^k}{k!}}{\sum_{j=0}^{R} \frac{A^j}{j!}}, \; 0 \le k \le R \tag{7-16}$$

$$E(R, A) = P_R = \frac{\frac{A^R}{R!}}{\sum_{j=0}^{R} \frac{A^j}{j!}} \tag{7-17}$$

The notation *E(R, A)* is relatively classical, and it gives the probability of finding the system full (all the servers active); in view of the nature of the arrivals, this is also the rejection probability. It is possible to prove that the preceding result is valid even for a service of any type (M/G/R/R system). (The Anglo-Saxons designate this fundamental result as the Erlang-B formula).

When the number of servers becomes large, the denominator of the formulas above tends towards the exponential of A, and in these conditions the state probabilities are given by $P_k \approx \frac{A^k}{k!} e^{-A}$. They are governed by a Poisson law (but this is an approximate limit, that is valid only for low loss probabilities, k close to R and large R).

The effective calculation of the loss formula raises a problem in the form given above: it would be pointless to evaluate a factorial by a direct method, for a relatively large R! It is best to use a recurrence method, based on the relationship (derived directly from the above formula):

$$E(R+1,A)=\frac{AE(R,A)}{R+1+AE(R,A)} \tag{7-18}$$

which leads to the recurrence:

```
X := 1
 for j going from 1 to R do
    X := 1 + X*j/A
 End for
 E(R,A) := 1/X
```

It is also often possible to use an approximation (to replace the sum in the denominator by the exponential, and use the Stirling formula for the factorials: $R! \approx R^R e^{-R}\sqrt{2\pi R}$), which gives quite spectacular precision:

$$E(R,A) \approx \left(\frac{A}{R}\right)^R e^{R-A} / \sqrt{2\pi R} \tag{7-19}$$

If we return to the definition of traffic offered and traffic carried, the calculation of traffic carried enables the checking of the matching up of the definitions to be carried out:

$$A_e = \sum jP_j = A\left[1-E(R,A)\right]$$

The diagram given at the end of the book indicates the probability of rejection of an Erlang system for several values of R. The Erlang formula can also be used to illustrate the phenomenon of economies of scale already encountered. For example, two groups of 5 servers each receiving a traffic of 3 erlangs will present a rejection rate in the order of 10%. Grouping together the two groups, 6 erlangs will be offered to 10 servers, bringing this rate down to about 4%.

7.5.5. *M/M/R queue*

Let us consider again a system with R servers, this time with a queue. We are in the situation of the previous case, but in this case when a client arrives and finds the R servers occupied, he joins the queue.

What happens to service rates? As long as $k < R$, nothing changes compared with the previous example: the service rate is $k\mu$. As soon as k reaches the value R, the servers are saturated and the service rate remains equal to $R\mu$.

The traffic offered is noted A and the traffic per server is $\rho = A/R$. The stability condition remains $A < R$, that is $\rho < 1$.

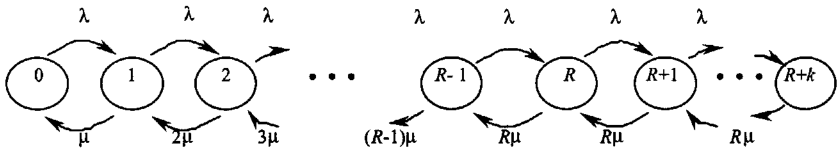


Figure 7.10. *State transition diagram of the M/M/R queue*

The exercises of writing the equations and their resolution are left to the reader. The following are obtained:

$$P_k = P_0 \frac{A^k}{k!} \quad k < R,$$

$$P_k = P_0 \frac{A^R}{R!} \rho^{k-R}, \; k \geq R$$

with:

$$P_0 = \left[\sum_{k<R} \frac{A^k}{k!} + \frac{A^R}{R!} \frac{1}{1-\rho} \right]^{-1}$$

The probability of having to wait is:

$$E_2(R,A) \equiv P_W = \frac{A^R}{R!} \frac{P_0}{1-\rho} = \frac{\dfrac{A^R}{R!} \dfrac{R}{R-A}}{\sum_{k<R} \dfrac{A^k}{k!} + \dfrac{A^R}{R!} \dfrac{R}{R-A}} \tag{7-20}$$

This is simply the stationary probability that R clients or more are in the system. This formula is known by the name of "Erlang formula with waiting" (or "Erlang-C ", *C(R,A)*). The mean waiting time is deduced from what goes before, after some calculations:

$$E(W) = \frac{P_W}{R-A} E(s)$$

This formula makes it possible to obtain the mean waiting time, as seen by the clients who are actually waiting. The calculation of the distribution of the waiting of clients who are actually waiting is carried out by using the theorem of conditional probabilities:

$$P(waiting \leq t \,/\, the\,client\,waits) = \frac{P(waiting \leq t)}{P(the\,client\,waits)} = \frac{P(waiting \leq t)}{P_W}$$

The transition to mean values gives the following directly:

$$E(W') = \frac{E(s)}{R-A},$$

where W' is the mean time measured on clients who *are actually waiting*. The comparison between the two expressions is instructive: waiting is not perceptible, except by clients who are actually waiting, and the mean waiting time calculated on all clients is only slightly representative of the perceived quality of service.

Furthermore, the use of these results offers an initial example of the general law of statistical phenomena, i.e. the gain in efficiency linked to the sharing of resources. Let us consider this principle using here a very simple exercise.

Example: sharing of resources

A projected communication system is organised around two so-called specialised processors, one of which processes the functions associated with a user group and the other those of another group, each having its own queue. This may be the case for example with call set-up or message or packet switching functions. The processing request arrival rates are the same for each processor (140/second). The arrivals are assumed to be in accordance with a Poisson law. The service times are the same for each processor (same power) and are assumed to conform to an exponential law with a mean of 6 ms.

a) What is the waiting probability, and the average waiting time?

b) The head of the system team decides to make the architecture evolve into a system of two general-purpose processors: the processing will be carried out by one or other of the processors, in fact the first one available, with both processors handling a common queue. The aim is thus to optimise the system's response time. What is the gain to be achieved by this modification?

Solution

a) As the servers work independently, we are dealing with two separate M/M/1 queues, with identical characteristics: $\rho_1 = \rho_2 = 140 \times 6.10^{-3} = 0.84$.

Hence immediately:

- the activity rate of each processor is 0.84, i.e. P(inactive)=0.16;
- the average waiting time is $W = \rho E(s)/(1-\rho)$ = 6×0.84/0.16 that is 31 ms (37.5 ms if this time is related to those who are waiting).

b) Now the servers are organised in a pool, which means the system is of the M/M/2 type.

The arrival rate is double (280 requests per second),

The total traffic offered is double: A = 280×6/1000 = 1.68, and $\rho = 0.84$ (the load of each of the processors remains identical).

The state probabilities of the system in this case can be written explicitly:

$$P_0 = \left[1 + A + \frac{A^2}{2(1-\rho)}\right]^{-1} = \frac{1-\rho}{1+\rho},\ P_W = \frac{A^2}{2(1-\rho)}P_0,\ W = \frac{E(s)}{2(1-\rho)}P_W$$

Here, we have: $P_0 \approx 0.088, P_W \approx 0.76, W \approx 14\,\text{ms}$. W' = 18 ms.

We have therefore almost divided in half the average waiting time, without an increase in the installed power.

This elementary example illustrates the argument for pooling resources: an M/M/*k* queue will be more efficient than *k* M/M/1 queues in parallel, all other things being equal.

Relationship with the Erlang formula

It is useful to link up the preceding formulae with those of the Erlang "loss" system. The Erlang formula is easily calculated by means of the recurrence given above, which can also supply $E_2(R,A)$. A simple manipulation gives:

$$E_2(R,A) = \frac{R.E(R,A)}{R - A + A.E(R,A)}$$

For low probability values, the following approximate form may be given:

$$E_2(R,A) \approx \frac{R}{R-A} E_1(R,A) \tag{7-21}$$

Waiting time distribution

When all the servers are busy, the interval until the next end of service is exponentially distributed, with a rate $R\mu$. The probability that no end of service will occur for a time t is $(e^{-\mu t})^R = e^{-\mu Rt}$ (this is the probability that no service will stop, and thus the product of probabilities for each of the servers). As soon as a server is released, and if there is at least one client waiting, the server is immediately busy.

On arrival of a client, if there are $i \geq 0$ clients already waiting, the client will wait for a time longer than t, if and only if there are fewer than i ends of service during t. This probability is given by the Poisson law in view of the preceding remark:

$$\sum_{x=0}^{i} \frac{(\mu Rt)^x}{x!} e^{-\mu Rt}$$

And thus by conditioning to the state of the queue on the arrival of the client:

$$P(\text{waiting} > t) = 1 - F(t) = \sum_{i=0}^{\infty} P_{R+i} \sum_{x=0}^{i} \frac{(\mu Rt)^x}{x!} e^{-\mu Rt} = \sum_{x=0}^{\infty} \frac{(\mu Rt)^x}{x!} e^{-\mu Rt} \sum_{i=x}^{\infty} P_{R+i}$$

(the latter relation is obtained by reversing the order of summations). By developing the state probabilities, expressed on the basis of the waiting probability, this gives:

$$1 - F(t) = E_2(R,A) e^{-\mu Rt} \left(1 - \frac{A}{R}\right) \sum_{x=0}^{\infty} \frac{(\mu Rt)^x}{x!} \sum_{i=x}^{\infty} \left(\frac{A}{R}\right)^i$$

The second sum is rewritten $\left(\frac{A}{R}\right)^x / (1 - A/R)$, and thus:

$$1 - F(t) = E_2(R,A) e^{-\mu Rt} \sum_{x=0}^{\infty} \frac{(\mu Rt)^x}{x!} \left(\frac{A}{R}\right)^x = E_2(R,A) e^{-\mu(R-A)t} \tag{7-22}$$

7.5.6. *Limited-capacity models*

The M/M/1/K model corresponds to the case of a capacity of K clients. Note that the customary use of K is to represent the *total* number of clients in the system, i.e. waiting or in service (one occasionally finds the notation M/M/1/K+1, indicating that K are on standby and one in service).

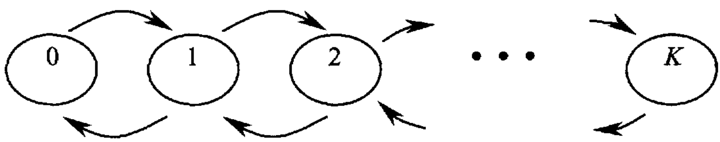


Figure 7.11. *State transition diagram for the limited-size queue*

The evolution diagram is obtained by truncating the M/M/1 diagram, and the equations easily give the result:

$$P_n = \frac{\rho^n(1-\rho)}{1-\rho^{K+1}}$$

For a limited capacity system, the performance criterion is of course the rejection probability. In this case, it is simply the stationary probability of observing K clients in the system, as then an arriving client will not be admitted:

$$\Pi = P_K = \frac{\rho^K(1-\rho)}{1-\rho^{K+1}} \tag{7-23}$$

Of course, in this case it is possible to remove the stability condition ($\rho < 1$), a higher loading value resulting in a higher level of rejection. It will be noted that the case $\rho = 1$ leads to a difficulty with the formula. This is only a mathematical problem, and it would be necessary to read for example:

$\Pi = \dfrac{\rho^K}{1+\rho+\rho^2+\ldots+\rho^K}$, which removes the difficulty.

7.5.7. *M/M/∞ queue*

There are situations in which arriving clients are always served immediately, without waiting, because the number of servers is always greater than the level of demand – the image used for this situation is that of an infinite number of servers.

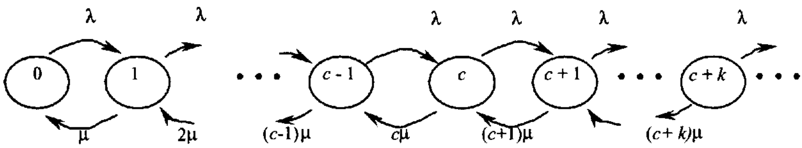


Figure 7.12. *The queue with infinitely many servers M/M/∞*

The modification of the M/M/R queue diagram is simple, and the state probabilities can be easily written:

$$P_k = P_0 \frac{A^k}{k!},\ k \geq 0 \text{, and thus } P_0 = \left[\sum_{k\geq 0} \frac{A^k}{k!}\right]^{-1} = e^{-A} \text{, i.e. } P_k = \frac{A^k}{k!} e^{-A}$$

This system is quite special, in the sense that no client will ever wait. The term sometimes used for this system is *pure delay*.

It will be observed that the state probabilities are governed by a Poisson Law with parameter A.

It has already been noted, in the Erlang problem, that the Poisson Law is the limit. This result is also found here, as the M/M/∞ system is the limit of the M/M/R/R system, if R is tending towards infinity. In fact, we will see below another configuration, the Engset problem, of which the Erlang and Poisson distributions can be seen as special cases, which all have the Poisson law as a limit. The importance of the previous result is related to the fact that when we have a system that is very generously sized, to which Poissonian traffic is offered, the system may be calculated using the Poisson law.

The reader will note that we have already found the application of this property in the sizing of spare part lots, in which for low probabilities of saturation of maintenance resources, the maintenance stock can be dimensioned by the Poisson formula.

7.5.8. *Limited-population system: the Engset problem*

The Poisson assumption, which enables the analysis of the preceding mechanisms, is based on the assumption of a constant client arrival flow. A particularly interesting case is that in which the flow stems from a population of sources of finite size, with each of the sources having a very simple behaviour, and governed by memoryless laws. This is the Engset problem.

It is assumed that the behaviour of each source is as follows: a source remains inactive for a random time with an exponential distribution of the parameter λ. When it becomes active again, it requests a service, which will last for a time whose distribution is exponential (parameter μ). In the Engset problem, the servers are in limited number, and if they are all busy when a request arrives, the request is rejected and the source begins another period of non-activity. The following diagram shows the behaviour of a source in schematic form.

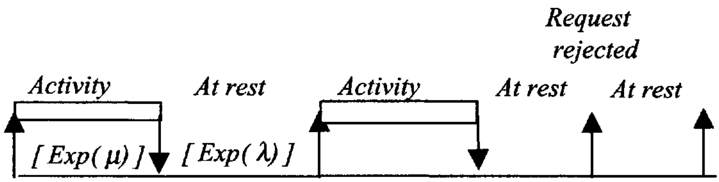


Figure 7.13. *The evolution pattern of a source*

Let us consider a population of N sources with identical behaviour, served by a group of R servers without waiting. To make the problem significant, we will assume that $R \leq N$. This configuration is noted "M(n)/M/R/R".

The key to dealing with this problem can be found in the following statement: *the arrival process depends on the instantaneous state of the system.* In fact, "an active source does not generate a new request". Assumptions on the behaviour of the sources make the same type of analysis possible: the system is characterised by the number n of active clients. On this occasion however, the elementary events are:

- The arrival of a new client, which causes a move from n to $n+1$ if $n < R$. In state n, only $N - n$ sources are inactive and likely to become active again: the corresponding birth rate is $(N-n)\lambda$.
- The end of a service, with the rate $n\mu$, which moves n on to $n - 1$, as in the system with several ordinary servers.

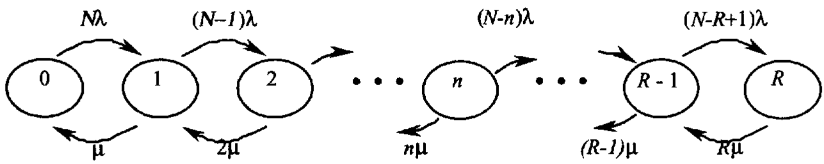


Figure 7.14. *State transition diagram for the M(n)/M/R/R system*

The set of equations giving the stationary probability of observing n busy servers is written (here form (b) of the equation (7-8) takes on its full importance):

$$(N-n)\lambda P_n = (n+1)\mu P_{n+1},\ 0 \le n < R$$

The solution is written:

$$P_n = \frac{\binom{N}{n}\alpha^n}{\sum\limits_{0\le k\le R}\binom{N}{k}\alpha^k}, \qquad \alpha = \frac{\lambda}{\mu}$$

In this equation, parameter α represents the behaviour of the elementary source. But it does not play the offered traffic role (total traffic offered is not $N\alpha$). The traffic offered by a source is $a = \dfrac{\lambda}{\lambda+\mu} = \dfrac{\alpha}{1+\alpha}$. This is also the occupancy factor of the source, which may be noted p (= a). Then we have the relationship: $\alpha = \dfrac{a}{1-a} = \dfrac{p}{q}$ whose usage will be seen later.

Rejection probability

It is essential to note that, unlike the case of the Erlang problem, P_R here is *not* the probability of rejection! If we reflect on the construction of the diagram and its significance, we realise that this quantity represents the proportion of time during which all the servers are busy. It is the occupation probability that would be measured by sampling by an external observer, in other words independent of the system (in accordance with the remarks made in section 7.4.4). But clients who arrive do not do so independently of the state of the system, as they arrive in fewer numbers when the system is filling up. This system is a good *a contrario* illustration of the PASTA property.

The special case $N = R$ is instructive in this respect. It is worth noting the simplified form that state probabilities then take:

$$P_n = \frac{\binom{N}{n} a^n}{(1+a)^N}$$

We often speak of Bernoulli traffic in this configuration. We then have $P_R = P_N = [a/(1+a)]^N \neq 0$, while the rejection probability is of course zero.

How then is it possible to calculate the rejection probability, in the general case in which $N > R$? Let us imagine observation of the system over a "long" period T. In accordance with the state probability properties, the system resides for a proportion P_n of this time in each state n. The number of clients arriving in duration T will therefore be, on the average, $\sum (N-n)\lambda P_n T$. The clients rejected are those who arrive in state R, in the number of $(N-R)\lambda P_R$, and the rejection probability will be the quotient of these two values. Thus,

$$\Pi = \frac{(N-R)\lambda P_R}{\sum (N-n)\lambda P_n}.$$

After a moderate amount of algebraic gymnastics, the formula will be expressed as a function of known quantities:

$$\Pi = \frac{\binom{N-1}{R} \alpha^R}{\sum_{n=0}^{R} \binom{N-1}{n} \alpha^n}, \text{ or } \Pi = \frac{\binom{N-1}{R} p^R q^{(N-1)-R}}{\sum_{n=0}^{R} \binom{N-1}{n} p^n q^{(N-1)-n}} \qquad (7\text{-}24)$$

This formula is known as the Engset formula.

As for the Erlang formula, the numerical evaluation of the loss probability is made easier using the following recurrence:

$$\Pi_j = \frac{\alpha(N-j)\Pi_{j-1}}{N + \alpha(N-j)\Pi_{j-1}}$$

Lastly, as the probability of success of the activity attempt is $1-\Pi$, the mean duration of the cycle is:

$$E(c)=\frac{1}{\lambda}+\frac{1-\Pi}{\mu}$$

The traffic carried by each source will be given by the activity rate of the source:

$$a_e=\frac{(1-\Pi)/\mu}{E(c)}=\frac{\alpha(1-\Pi)}{1+\alpha(1-\Pi)},$$

The total traffic carried will be N times this quantity.

The same holds for the traffic offered. The traffic offered per source is $a_o=\dfrac{\alpha}{1+\alpha(1-\Pi)}$. The total traffic offered will be N times this quantity (this value is generally the one given in tables).

Example: sizing of a subscriber concentrator

For example, let us consider the subscriber stage of a telephone switch, concentrating the traffic of N users on a restricted number R of lines to the switch. In view of the telephoning habits of the subscribers, it is necessary to calculate R to offer a given level of service without using circuits whose utilisation factor would be too low.

The following figure shows what the formula gives, depending on N and a values, for R = 10.

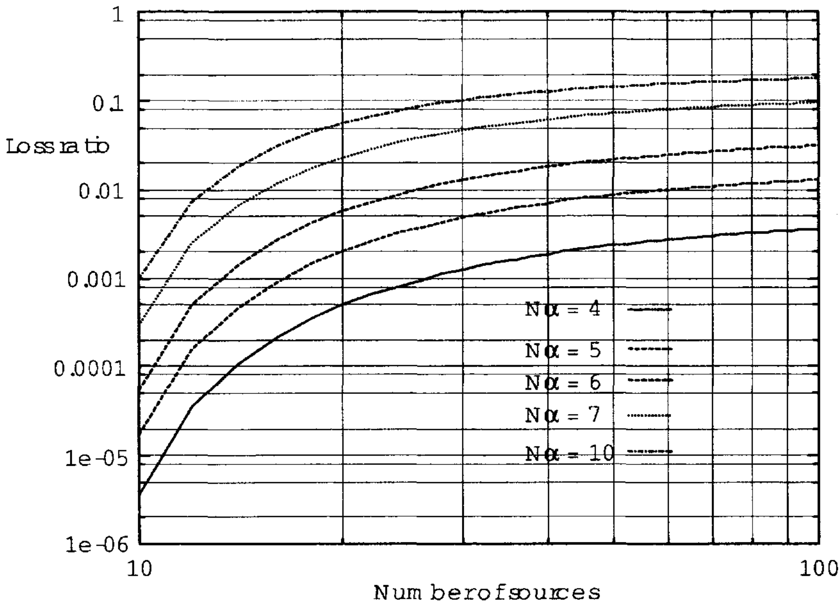


Figure 7.15. *Loss probability for the Engset model, R = 10*

Clearly when N increases (for a constant $N\alpha$ product), the loss probability increases. It could be checked that the limit (when $N \to \infty$) is that given by the Erlang formula (M/M/R/R queue). It can also be seen that the limited population has a beneficial effect on the curve.

Numerical application: Poissonian traffic of 4 erlangs offered to 10 circuits suffers a loss in the order of $5\ 10^{-3}$; if the traffic comes from a group of 20 sources, the rejection will be 5 times lower, in the order of 10^{-3}.

7.6. More complex queues

The assumptions underlying the previous models (Poissonian arrivals, exponentially distributed service times) make analysis simple. Abandoning these hypotheses prevents the use of the birth and death process approach. There are however systems for which adapted modelling enables precise resolution.

7.6.1. *The imbedded Markov chain method*

In the following, we will use this type of method to establish some important results on queueing with any service law. However, the arrival process will be such that the evolution of the system (clients and server) is what is known as *semi-Markov* process. (Then, we carry on in the following stage to resolve the general case, with any arrival law and any service law, using the Pollaczek method).

For such a process, one attempts to characterize transition probabilities at particular instants only (as opposed to the birth and death processes, where the system is studied at an arbitrary instant t). The key to modelling always consists of finding, in the evolution of the system, instants in which the future and the past are independent. It is then possible to extract from this evolution a Markov chain (*imbedded Markov chain*) , whose resolution is possible, possibly in numerical form.

This method is particularly well suited to calculating the number of clients in the system, if one is able to define instants such that knowing the instantaneous state allows deducing the following one. Especially, the instants of end of service have this property. The state of the system at these instants is characterised by the number of clients left by the departing client. We must be able to calculate the probability of having a certain number of arrivals, α, during the next service duration, which depends only on the service duration and is independent of the system state. This imposes several conditions on the arrival process, such that the independence between arrivals occuring in the successive service durations.

We now apply this method to a queue with general service distribution and establish a quite interesting result concerning the number of clients in the system. This result will be used in the following for the systems M/G/1, and Geo/D/1 for example.

7.6.2. *The number of clients in system*

Let us consider a queue, with Poissonian arrivals, and service times distributed in accordance with any law, whose distribution will be noted $B(t)$, and with an arrival process complying with the criterion given above (of the Poisson or Bernoulli kind, for example). The stationary analysis of the birth and death processes is not applicable, because the probability of an end of service in the elementary interval Δt will depend on the service age in progress (referring back to the result on the remaining service time, it will become obvious that the exponential law is the only one to demonstrate the “memoryless” property).

Now, let us denote as α_k the probability that k clients arrive during a service (the arrival process must be such that this probability may be calculated), and let us observe the system at the end of service instants. Let us note X_n the number of clients left behind after the departure of the n^{th} customer. If X_n is greater than 0, another service starts immediately. If not, the server remains inactive, until the arrival of a client, who will be served without having to wait. In all cases, at the end of the next service, the system will have accommodated other clients, in the number of A_n. These clients are those who arrive, during the time of the n^{th} service.

It is then possible to write a recurrence relation, whose resolution will give the probability distribution of the X_n's.

Let us assume that $X_n > 0$. The departure of client n results in the diminution of one unit of the number of clients. Any arrivals during the next service will make the number of clients increase, and this will give:

$$X_{n+1} = X_n - 1 + A_{n+1}$$

If on the contrary, $X_n = 0$, then the next client begins the service, and we then see that:

$$X_{n+1} = A_{n+1}$$

These two cases are summed up by the abridged notation:

$$X_{n+1} = [X_n - 1]^+ + A_{n+1},$$

an expression in which $[x]^+ = \max(x,0)$.

(This is the Lindley relation. We will see later, with Pollaczek's method, who introduced this kind of relation, all the benefit one can draw from it.)

It is also possible to illustrate the behaviour of the system (evolution of number of clients waiting or in service) by a diagram, similar to the preceding ones, but which can no longer be termed “birth and death”. Between the departure of clients n and n +1, 0, 1, 2, etc. clients may arrive, causing jumps corresponding to the relation (7-31). Note that it is possible to speak of a state: the number of clients *just after* the departure of the nth enables certain prediction of the probability distribution at the next departure. Indeed, the assumptions of the model allows calculation of α_k, the probability that k clients arrive during a service.

If the departure of a client leaves the system in state $j > 0$, then a service immediately begins. If now k clients arrive during this service, the number remaining after the next departure will be $j + k - 1$ (as the client served will leave the system). More generally, the system jumps from a state $j > 0$ to another state $m > 0$ with the arrival of $m + 1 - j$ clients, event having probability α_{m+1-j}. This is shown in the graph in Figure 7.16.

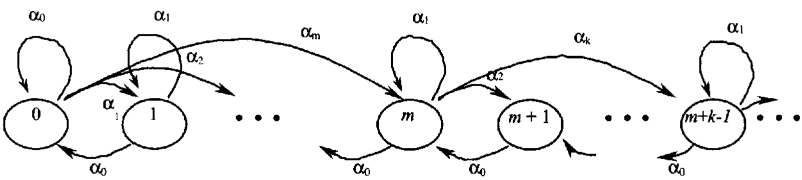


Figure 7.16. *State transition diagram of the imbedded Markov chain*

In state 0, the server is inactive, and an arrival triggers a start of service. If k clients arrive during this service, the departure of the client starting up the period of activity will leave these k clients behind him, which explains the particularity of the transitions observed since state 0, compared with the transitions from the other states.

The resolution of this system, i.e. the calculation of the state probabilities, will follow the same general method. A system of equations of the Chapman-Kolmogorov type will be written:

$$P_0(1-\alpha_0) = \alpha_0 P_1$$
$$P_1(1-\alpha_1) = \alpha_1 P_0 + \alpha_0 P_2$$
$$P_2(1-\alpha_1) = \alpha_2 P_0 + \alpha_2 P_1 + \alpha_0 P_3$$
$$\dots$$
$$P_k(1-\alpha_1) = \alpha_k P_0 + \sum_{j=1}^{k+1} P_j \alpha_{k+1-j} - \alpha_1 P_1$$

The resolution of this system will take place by means of the resolution of a truncated matrix system, or better by making use of generating functions: multiplying the equation of rank k by z^k, and summing the system, the generating functions are introduced:

$$P(z) = \sum P_k z^k, \; A(z) = \sum \alpha_k z^k$$

Setting up the equation raises no difficulties. This gives:

$$P(z) = P_0 \frac{(1-z)A(z)}{A(z)-z}$$

Moreover, one has evidently $P_0 = 1 - \rho$.

This gives finally the important relation giving the transform of the number of clients in the system:

$$P(z) = A(z)\frac{(1-\rho)(1-z)}{A(z)-z} \tag{7-25a}$$

For Poissonnian arrivals, one has:

$$A(z) = \sum z^k \int \frac{(\lambda x)^k}{k!} e^{-\lambda x} dB(x) = \int e^{\lambda zx} e^{-\lambda x} dB(x) = B*(\lambda z - \lambda)$$

(we have introduced the Laplace transform of the service law).

At last we have the important result for the M/G/1 system:

$$P(z) = (1-\rho)\frac{(1-z)B*(\lambda - \lambda z)}{B*(\lambda - \lambda z) - z} \tag{7-25b}$$

which is, as we will see later, one of the forms of the Pollaczek transform.

As usual, we derive the moments of occupation probabilities from the transforms. We will provide in Chapter 9 an example of application of these results (Geo/D/1 queue for the case of a ATM switching matrix).

7.6.3. *Waiting times: Pollaczek formulae*

7.6.3.1. *Preliminary: calculation of remaining service time*

Let us observe the server of an M/G/1 system at any instant. What is the time remaining until the end of the service in progress (zero if the server is inactive)? Note the difference with the calculation in Section 3, where it was a matter of calculating the remaining time bearing in mind that y seconds had already elapsed. Here the observation of the server takes place independently of the server. We are therefore searching for the temporal mean of the remaining service time – without knowing when it began.

We observe the system for a time T. Note as $X(t)$ the time remaining at any instant t. At each end of service, and if a client is present, a new service begins, triggering an instantaneous increase in $X(t)$ of a value equal to the service time requested (we will note as S_k the service time of client k). Finally, the remaining time evolves analogously to the chart in the figure below.

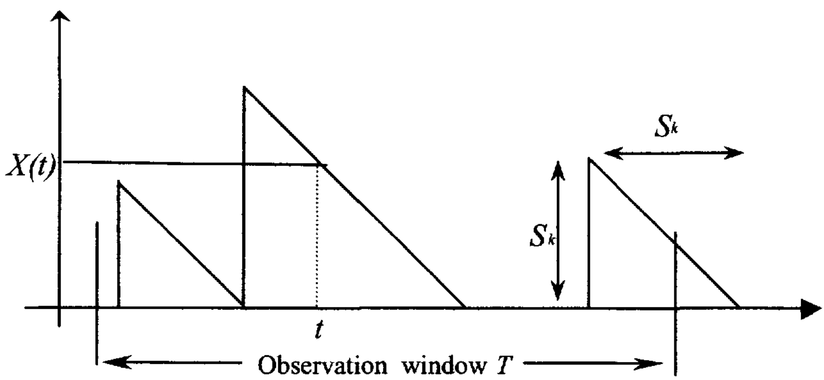


Figure 7.17. *Residual service time in the M/G/1queue*

The chart consists of initial jumps (at the start of each service, the remaining time is increased by the new service time), and then by segments of 45° slopes. The temporal mean can be calculated:

$$\frac{1}{T}\int_0^T X(t)dt = \frac{1}{T}\sum_0^{N(T)} \frac{S_k^2}{2},$$

where $N(T)$ represents the number of clients served in the interval, and S_k the sequence of service times.

All the clients are statistically identical, and the passage to the limit gives:

$$\begin{aligned}E(X) &= \lim_{T\to\infty} \frac{1}{T}\sum \frac{S_k^2}{2} \\ &= \frac{1}{2}\lim_{T\to\infty} \frac{N(T)E\left(S_1^2\right)}{T} \\ &= \frac{1}{2}\lambda E(S^2)\end{aligned} \tag{7-26}$$

This quantity is conventionally noted as W_0. It is worth pointing out that the calculation does not depend on the service discipline (i.e. on the choice that the server makes about the next client to be processed).

7.6.3.2. *The Pollaczek-Khintchine formula*

We put ourselves in the context of the M/G/1 queue. A client who arrives in accordance with the Poisson process, and thus independently of the state of the queue, observes the queue's stationary state. He may find a client in service (or the server inactive). He observes the corresponding service time remaining, x. He also observes n other clients in the queue, who will be served before him. His/her wait will therefore be:

$$w = x + \sum_1^n S_i$$

Let us take the mean values. That of x has been calculated above. The mean of the sum will be *E(n)* times the mean service time. But *E(n)*, the mean number of clients waiting, is linked to *E(W)*, by the Little formula. Hence:

$$\begin{aligned}E(W) &= W_0 + E\left[\sum_1^n S_i\right] \\ &= W_0 + E(N)E(S) \\ &= W_0 + \lambda E(W)E(S)\end{aligned}$$

and finally the formula sought:

$$E(W) = \frac{W_0}{1-\rho} = \frac{\rho}{1-\rho} \times \frac{1+c_s^2}{2} \times E(s), \tag{7-27}$$

into which we have introduced the *coefficient of variation* of the service time: $c_s^2 = \left(\frac{\sigma(s)}{E(s)}\right)^2 = \frac{E(s^2)-\left(E(s)\right)^2}{E(s)^2}$, which quantifies the "variability" of service times.

This is the Pollaczek-Kintchine formula (from the name of those who developed it), often designated PK. We will redemonstrate this result in the sequel, this time, as a special result of the original method of Pollaczek.

Its interpretation is extremely interesting. The first term quantifies the waiting phenomenon, which depends on the system utilisation factor. The second indicates the variability of the service, and the third is a "scale factor", the scale of the phenomenon being given by the mean service time.

7.6.3.3. *Example 1: the M/M/1 queue*

For an exponentially distributed service time, it can easily be shown that $E(s^2) = 2/\mu^2$, i.e. $c_s^2 = 1$. Hence:

$$E(W)_{M/M/1} = \frac{\rho E(s)}{1-\rho} \quad (\mathrm{Rq}: E(s) = \frac{1}{\mu})$$

7.6.3.4. *Example 2: the M/D/1 queue*

Clearly, the variance of the service time is zero. Thus, $c_s^2 = 0$, and:

$$E(W)_{M/D/1} = \frac{\rho E(s)}{2(1-\rho)} = \frac{1}{2} E(W)_{M/M/1}$$

7.6.3.5. *Generalisation: Takacs formulae*

The following result, attributed to Takacs, enables obtaining the successive moments of the waiting times for the M/G/1 queue. They involve the successive moments of the service time.

$$E(W^0) = 1$$

$$E(W^k) = \frac{\lambda}{1-\rho} \sum_{i=1}^{k} \binom{k}{i} \frac{E(s^{i+1})}{i+1} E(W^{k-i}) \tag{7-28}$$

We will leave it for the reader to verify by exercises that, taking $k=1$ in the preceding form, we find again the PK formula. The following relations can be deduced for *sojourn times*:

$$E(T^k)=\sum_{i=0}^{k}\binom{k}{i}E(s^i)\ \ E(W^{k-i}) \tag{7-29}$$

For example, for $k=1$: $E(T)=E(s)+E(W)$.

7.6.4. *The Benes method. Application to the M/D/1 system*

The Benes method focuses on "unfinished work" or "virtual waiting time". It will be illustrated here by deriving the distribution of waiting times in the M/D/1 queue.

Let us take an arbitrary instant as the origin, and note as $A(t)$ the quantity of work arriving in the interval $[-t,0)$. (We assume the server processes 1 unit of work per unit of time.) Let us term V_{-t} the working time, or virtual waiting time, remaining at the instant $-t$ (time to serve the waiting clients, plus the remaining time for any client in progress). We introduce $\xi(t)=A(t)-t$, the work in excess arriving in the interval. It can be seen that the unfinished work is given by:

$$V_{-t}=\sup_{u\geq t}\left(\xi(u)-\xi(t)\right)$$

In particular, $V_0=\sup_{t\geq 0}\xi(t)$ (this is the strongest surplus of work to have occurred in the past, the one whose effect remains at 0). The method is classically focused on the complementary distribution $Q(x)$:

$$Q(x)=P\left[V_0>x\right]$$

The event $\{V_0>x\}$ can be read: "there exists a value of t, such that $\xi(t)>x$". One can thus condition on the last instant T_x (the largest interval) where one observed $\xi(t)=x$: this is $T_x=\sup\{t\geq 0;\xi(t)=x\}$. For more remote instants, one must have $\xi(t)<x$, such that $V_{-T_x}=0$ (it is at this point that the difference is greatest). We can then write (this is the key point of the method):

$$Q(x) = \int_{u \geq 0} P\{V_{-u} = 0 \ \underline{\text{and}} \ \ \xi(u) \in (x, x+dx)\} \tag{7-30}$$

Application to the M/D/1 system

Let us term *n(t)* the number of clients arriving in $[-t, 0)$. The service time is taken as the time unit, such that $\xi(t) = n(t) - t$. The integral is then rewritten as a sum:

$$Q(x) = \sum_{j > x} P\{n(j-x) = j \quad \text{and} \quad V_{-(j-x)} = 0\}$$

The arrival process is independent of the state of the queue: the probability is written as the product of the two terms. As the arrivals are Poissonian, this gives:

$$Q(x) = \sum_{j > x} \frac{[\rho(j-x)]^j}{j!} e^{-\rho(j-x)}(1-\rho)$$

(λ and ρ are identical, as the service time is equal to 1). After manipulations involving combinatorics, the result is rewritten in the form of a finite (but alternate) sum:

$$Q(x) = 1 - (1-\rho) \sum_{j \leq \lfloor x \rfloor} \frac{[\rho(j-x)]^j}{j!} e^{-\rho(j-x)} \tag{7-31}$$

7.7. The G/G/1 queue

Many methods can be used to establish results relating to GI/G/1 queues. We will make use here of the *Pollaczek method*. Pollaczek was the first to establish fundamental results relating to this type of queue, and above all introduced a technique capable of solving the most complex problems.

7.7.1. *Pollaczek method*

The value of this method, in addition to its elegance, is that it can be used for complex queueing problems, involving random variables governed by extremely general distributions. We present the main elements of this method below. For a full presentation, see [LEG 62], [SYS 93], [COH 82] and obviously the articles of Pollaczek [POL 57, POL 61].

The principle is as follows:

- on the one hand, very often it is possible to establish simple relationships between independent random variables such as the waiting time of two consecutive clients, their service time and their inter-arrival time, whatever the arrival and service time distributions may be.
- on the other hand, in association with these relations we identify incompatible events, for example: as a function of the arrival instants of two consecutive clients, either waiting takes place or it does not. These events, when introduced into the previous relations, define for the variable studied (waiting, holding time, etc.) one or more fundamental relations that may be called *stochastic relations associated with the stochastic variable studied.*

As seen in Chapter 3, by associating with this expression of the variable its *indicator* we completely define its probability distribution. It is then "enough" to write characteristic function and from this deduce the moments by derivation, and any other property such as the distribution function, by use of the inverse transform and residue calculation.

Clearly a method of this type very rapidly leads to pure mathematical developments. In fact, from the second stage onwards it may almost be said that one has already left the pure domain of probabilities. This is its attractiveness for those who love mathematics. The downside is that the approach quickly seems to become abstract, and very often the developments are relatively complex. This tends to put off the physicist. But it will always be a good idea to refer to the physical phenomenon to gain better control of the calculations, and particularly problems at the limits for example. In fact simply practising the method more frequently opens up new perspectives, bearing in mind that the resolution of the complex problems raised by the new world of telecommunications would find it difficult to do without such tools. It is in this spirit that we present, below, the method and its application to the system with one server.

We will first introduce the method and show how the celebrated Pollaczek transform can be very simply obtained. Then we will give application examples.

Characteristic function of a random variable corresponding to two incompatible events

In the first stage, let us demonstrate a fundamental expression of the characteristic function of a random variable corresponding to two incompatible events.

Let Y be a random variable such that:

$$Y = X \text{, if } X > 0\text{; } Y = 0 \text{, if } X \le 0$$

This can also be noted as $Y = X^+ = Max(0,X)$, X being itself a random variable.

There are only two possible and incompatible events: E1 is the event $X > 0$ and E2 is the event $X \le 0$. The corresponding indicators are respectively $H(X)$ and $H(-X)$ (the Heaviside functions). It should be recalled (see the chapter on probabilities) that the indicator of the *event* $(x - X) > 0$ is $H(x - X)$.

As events E1 and E2 are mutually incompatible and represent all the possible cases, we can write

$$e^{zY} = e^{zX} H(X) + H(-X)$$

By moving on to expectations, we have:

$$E(e^{zY}) = E\{e^{zX} H(X)\} + E\{H(-X)\} \qquad (7\text{-}32)$$

This is, it will be recalled, the characteristic function of variable Y.

And given that, (see the Heaviside function in Chapter 4):

$$H(-X) = \frac{1}{2\pi i} \int_{C\zeta} e^{-\zeta X} \frac{d\zeta}{\zeta} \text{, where } R(\zeta) > 0$$

and also:

$$e^{zX} H(X) = \frac{1}{2\pi i} \int_{Cu} e^{(z+u)X} \frac{du}{u} \text{, with } R(u) > 0$$

and if we note $z+u=-\zeta$, therefore $z=-(u+\zeta)$ and $R(z)<0$, we then obtain:

$$e^{zX}H(X)=\frac{-1}{2\pi i}\int_{C\zeta} e^{-\zeta X}\frac{d\zeta}{\zeta+z}, \text{ with } 0<R(\zeta)<R(-z) \tag{7-33}$$

This gives us finally:

$$E(e^{zY})=\frac{1}{2\pi i}\int_{C\zeta}\left\{E(e^{-\zeta X})\right\}\left[\frac{1}{\zeta}-\frac{1}{\zeta+z}\right]d\zeta,$$

that is

$$E(e^{zY})=\frac{1}{2\pi i}\int_{C\zeta}\left\{E(e^{-\zeta X})\right\}\left[\frac{1}{\zeta}-\frac{1}{\zeta+z}\right]d\zeta, \text{ where } 0<R(\zeta)<R(-z) \tag{7-34}$$

Fundamental relation for the characteristic function of the variable $Y = X^+$ corresponding to two incompatible events of X.

7.7.2. Application to the stochastic relation of the queue to one server (GI/G/1 queue)

This result can be applied to the study of waiting in the simple case of an isolated queueing system (i.e. that can be considered in isolation, independently of any correlation with its environment).

The principle is to find an expression relating to waiting that translates two incompatible events of indicators with values 0 and 1.

The single server, with any service law, serves the clients, with any arrival law, in their order of arrival. The first client is noted as $n = 0$, the second $n = 1$, etc. Services and arrival instants are assumed to be independent.

Let the terms be as follows:

– X_n the random arrival instant of the nth client and
– T_n the random duration of its service.

Let:

$$Y_n = X_{n+1} - X_n .$$

The Y_n process is assumed to be regenerative (the Y_n are therefore mutually independent).

The waiting time of the nth client W_n is the time interval separating its arrival instant from its start of service. And so the nth client begins to be served at the random instant $X_n + W_n$ and has finished being served at $X_n + W_n + T_n$.

There are two possible events.

Either client $(n+1)$ arrives before the departure of the nth, in which case:

$$(X_n + W_n + T_n) - X_{n+1} > 0$$

and its random wait is:

$$W_{n+1} = (X_n + W_n + T_n) - X_{n+1} = W_n + T_n - Y_n$$

Or it arrives after the departure of the nth, in which case:

$$(X_n + W_n + T_n) - X_{n+1} < 0$$

and its wait is zero: $W_{n+1} = 0$,

which brings us finally to the following relation:

$$W_{n+1} = Max(W_n + T_n - Y_n ,0) = (W_n + T_n - Y_n ,0)^+ \tag{7-35}$$

which is a fundamental stochastic relation of the GI/G/1 and of which another example has recently been encountered (see Lindley's relation).

By direct application to the fundamental application of the characteristic function of the wait established previously, we have:

$$E(e^{zW_{n+1}}) = \frac{1}{2\pi i}\int_{C\zeta}\left\{E(e^{-\zeta(W_n+T_n-Y_n)})\right\}\frac{z}{\zeta(\zeta+z)}d\zeta \text{ , with}$$
$$0 < R(\zeta) < R(-z) \tag{7-36}$$

Let us study this integral.

As the variables W_n, T_n, Y_n are assumed to be independent, and as W_0 is fixed at a constant arbitrary value, we have:

$$E(e^{-\zeta(W_n+T_n-Y_n)}) = E(e^{-\zeta W_n})E(e^{-\zeta T_n})E(e^{+\zeta Y_n}) = E(e^{-\zeta W_n})E(e^{-\zeta(T_n-Y_n)})$$

Let us return to the characteristic functions. We introduce the notations:

$$E(e^{zW_{n+1}}) = \phi_{n+1}(z), \ R(z) < 0$$

$$E(e^{-\zeta W_n}) = \phi_n(-\zeta), \ R(\zeta) > 0$$

$$E(e^{-\zeta(T_n-Y_n)}) = \varphi(-\zeta) = E(e^{-\zeta T_n})E(e^{\zeta Y_n}) = \varphi_1(-\zeta)\varphi_2(\zeta),$$

where $\varphi_1(\zeta) = E(e^{\zeta T_n}), \varphi_2(\zeta) = E(e^{\zeta Y_n})$

And thus the integral studied is written:

$$\phi_{n+1}(z) = \frac{1}{2\pi i}\int_{C\zeta} \phi_n(-\zeta)\varphi(-\zeta)\frac{z}{\zeta(\zeta+z)}d\zeta, \tag{7-37}$$

with $0 < R(\zeta) < R(-z)$.

This relation characterises the GI/G/1 queue. Although very simple to establish, it is very important, and we will make use of it to demonstrate, in a very simple way, the celebrated Pollaczek transform, more particularly concerning the waiting time in an M/G/1 queue, following for this purpose the use of the Pollaczek method according to the approach set out in [LEG 62].

Pollaczek has demonstrated another more elaborate expression of the previous result, but which to be established necessitates more complex developments. We provide the main lines in an appendix to this chapter. The result, still for the GI/G/1 queue, is as follows:

$$\phi(z) = \exp\left\{-\frac{1}{2\pi i}\int_{-i\infty-0}^{+i\infty-0} \ln\left[1-\varphi(-\zeta)\right]\frac{z\,d\zeta}{\zeta(z+\zeta)}\right\} \tag{7-38}$$

where $\varphi(-\zeta)=\varphi_1(-\zeta)\varphi_2(\zeta)$, $R(z)\leq 0$. And bearing in mind that in order to exist this stationary limit process requires compliance with the condition $|\varphi(-\zeta)|\leq\varphi[-R(z)]<1$.

In the following we will also use this formula to establish the result of the G/M/1 queue.

7.7.3. *Resolution of the integral equation*

7.7.3.1. *Application to the M/G/1 queue*

Let us reconsider the general expression (7-37)

$$\phi_{n+1}(z)=\frac{1}{2\pi i}\int_{C\zeta}\phi_n(-\zeta)\varphi(-\zeta)\frac{z}{\zeta(\zeta+z)}d\zeta$$

where:

$$\varphi(-\zeta)=E(e^{-\zeta T_n})E(e^{\zeta Y_n})=\varphi_1(-\zeta)\varphi_2(\zeta)$$

As the arrivals are Poissonian this gives

$F_2(t)=1-e^{-\lambda t}$ and $\varphi_2(\zeta)=\dfrac{\lambda}{\lambda-\zeta}$, where λ is the arrival rate.

The characteristic function becomes:

$$\phi_{n+1}(z)=\frac{1}{2\pi i}\int_{C\zeta}\phi_n(-\zeta)\varphi_1(-\zeta)\frac{\lambda}{\lambda-\zeta}\frac{z}{\zeta(\zeta+z)}d\zeta$$

In the zone $R(\zeta)>0$ the function:

$f(\zeta)=\phi_n(-\zeta)\varphi_1(-\zeta)\dfrac{\lambda}{\lambda-\zeta}\dfrac{z}{\zeta(\zeta+z)}$ has two poles $\zeta=\lambda$ and $\zeta=-z$,

which gives by application of the residue theorem:

$$\frac{1}{2\pi i}\int_{-i\infty-0}^{+i\infty-0}f(\zeta)d\zeta=-(R_1+R_2),$$

(with a minus sign as we do not integrate in the positive direction, we close the contour on the right!).

At the pole $\zeta = -z$, one has:

$$R_1 = \left[(\zeta + z) f(\zeta)\right]_{\zeta=-z} = -\frac{\lambda \varphi_1(z)}{\lambda + z}\phi_n(z)$$

Similarly at the pole $\zeta = \lambda$ we have:

$$R_2 = \left[(\zeta - \lambda) f(\zeta)\right]_{\zeta=\lambda} = -\frac{z\varphi_1(-\lambda)}{\lambda + z}\phi_n(-\lambda)$$

and therefore:

$$\phi_{n+1}(z) = \frac{\lambda \varphi_1(z)}{\lambda + z}\phi_n(z) + \frac{z\varphi_1(-\lambda)}{\lambda + z}\phi_n(-\lambda)$$

In a stationary condition, the relation becomes independent of n and therefore:

$$\phi(z) = \frac{z\varphi_1(-\lambda)\phi(-\lambda)}{\lambda + z - \lambda\varphi_1(z)}$$

Recalling that the series expansion of $\varphi_1(z)$ gives:

$$\varphi_1(z) = 1 + m_1 z + m_2 \frac{z^2}{2!} \cdots, \text{ where } m_1 = \varphi'_1(0), \text{ and } \rho = \lambda m_1$$

the condition $\phi(0) = 1$ finally gives, now noting $\phi(z) = \phi_w(z)$:

$$\phi_w(z) = \frac{(1-\rho)z}{\lambda + z - \lambda\varphi_1(z)} \tag{7-39}$$

This is the fundamental relation giving the characteristic function of waiting for the M/G/1 server. This is the celebrated Pollaczek transform, of which we will see application examples in the following sections.

The simplicity of the demonstration is remarkable.

It is now possible to easily obtain moments by deriving and by returning to series expansion:

Let us recall that $\varphi_1(z) = 1 + z\varphi'_1(0) + \frac{z^2}{2!}\varphi''_1(z)..$, that is:

$$\varphi'_1(z) = \varphi'_1(0) + z\varphi''_1(0)..$$

In particular the average waiting time is obtained:

$W = \phi'(0) = \frac{\lambda}{2(1-\rho)}\varphi_1''(0)$, which is written:

$W = \frac{\rho(1+c^2)}{2(1-\rho)}\bar{x}$ where $\bar{x}$ is the mean of the service times and c^2 its coefficient of variation.

Or again by taking $\bar{x}$ as the time unit:

$$W = \frac{\rho(1+\sigma^2)}{2(1-\rho)} \tag{7-40}$$

This is the celebrated Pollaczek-Kintchine formula, established for the first time by Pollaczek in 1930 and since then redemonstrated many times by various methods (as we saw earlier).

We now need to calculate the *sojourn time* in the system:

By definition, we have, by calling S_n the sojourn time of the nth client:

$$S_n = W_n + T_n$$

and at equilibrium,

$$S = W + T$$

Let us note:

$$E(e^{zS}) = \phi_s(z)$$

As variables W and T are independent, this gives:

$$\phi_s(z) = \phi_w(z)\varphi_1(z)$$

Hence:

$$\phi_s(z) = \varphi_1(z)\frac{(1-\lambda)z}{\lambda + z - \lambda\varphi_1(z)}. \tag{7-41}$$

This is the characteristic function of the total time spent in the M/G/1 system.

By way of examples, let us apply these results to a set of typical cases which allow one to determine practical bounds (upper and lower) for most of real cases.

M/M/1 queue

Let us first apply the previous results to the M/M/1 case.

As the service law is exponential, this gives: $\varphi_1(z) = \dfrac{\mu}{\mu - z}$, and we immediately obtain:

$$\phi_w(z) = \frac{(\mu - z)(1-\rho)}{\mu(1-\rho) - z}$$

The function has a single pole:

$$z_1 = \mu(1-\rho)$$

Let us now apply our solution of the inversion formula:

$$F(x) = 1 - \frac{R_1}{(-z_1)}e^{-z_1 x}, \text{ and } P(>x) = \frac{R_1}{(-z_1)}e^{-z_1 x}$$

The residue R_1 in z_1 is:

$$R_1 = -\mu\rho(1-\rho).$$

Hence:

$$P(>x) = \rho e^{-\mu(1-\rho)x}.$$

We of course obtain the already established result.

M/D/1 queue

Let us now apply this to the M/D/1 queue. As the M/D/1 queue has already been processed, and as it gives rise to an expression which is complicated to calculate, we will establish here an asymptotic expression of the distribution of waiting. This is a very simple expression, which gives very precise results.

Let us start again from the Pollaczek transform. To simplify the writing we will take the service time as the unit. The Pollaczek transform is thus written:

$$\phi_{w}(z)=\frac{(1-\rho)z}{\rho+z-\rho\varphi_1(z)},$$

Let us now apply our asymptotic law as presented in Chapter 3:

$$P(>x)\approx\frac{R_1}{(-z_1)}e^{-z_1x}$$

Where $z_1 = \beta_0 > 0$ a singular point for $\phi_w(z)$, and such that:

$$\rho+\beta_0-\rho\varphi_1(\beta_0)=0$$

(This is the point closest to the origin, and on the real axis, see Levy's theorem in Appendix 1), and R_1 residue at z_1.

It will be recalled that (see Chapter 3) we also have: $R_1=\dfrac{P(z_1)}{Q'(z_1)}$

Hence:

$$P(>x)\approx\frac{(1-\rho)}{\rho\varphi'_1(\beta_0)-1}e^{-\beta_0x}$$

In the case of M/D/1, we have $\varphi_1(z)=e^z$,

$$\phi_w(z)=\frac{(1-\rho)z}{\rho+z-\rho e^z}.$$

We have β_0 such that $\rho+\beta_0-\rho e^{\beta_0}=0$ and $\rho\varphi_1'(\beta_0)-1=\rho e^{\beta_0}-1$.

And so,

$$P(>x) \approx \frac{(1-\rho)}{\rho e^{\beta_0} - 1} e^{-\beta_0 x} \tag{7-42}$$

This formula gives results that are sufficiently precise for most applications (see Chapter 9).

M/H2/1 queue

Now let us apply our results to the M/H2/1 system. This is to some extent the "opposite" of the M/D/1 case, from the viewpoint of the coefficient of variation, greater than 1, the M/M/1 case being "intermediate".

We start again from Pollaczek formula:

$$\phi_w(z) = \frac{(1-\rho)z}{\lambda + z - \lambda\varphi_1(z)}$$

$$P(>x) \approx \frac{(1-\rho)}{\lambda\varphi'_1(\beta_0) - 1} e^{-\beta_0 x}$$

with $z_1 = \beta_0 > 0$ singular point for $\phi_w(z)$, and such that:

$$\lambda + \beta_0 - \lambda\varphi_1(\beta_0) = 0$$

The service has a hyperexponential distribution H2, it becomes:

$$\varphi_1(z) = \frac{\alpha_1\mu_1}{\mu_1 - z} + \frac{\alpha_2\mu_2}{\mu_2 - z} \tag{7-43}$$

with μ being the service rate (larger than λ) given by $\frac{1}{\mu} = \frac{\alpha_1}{\mu_1} + \frac{\alpha_2}{\mu_2}$.

Hence:

$$\lambda + \beta_0 - \lambda\left(\frac{\alpha_1\mu_1}{\mu_1 - \beta_0} + \frac{\alpha_2\mu_2}{\mu_2 - \beta_0}\right) = 0$$

which gives, after a few developments, and using the relation $\frac{1}{\mu}=\frac{\alpha_1\mu_2+\alpha_2\mu_1}{\mu_1\mu_2}$:

$$\mu_1\mu_2(1-\rho)+\beta_0(\lambda-\mu_1-\mu_2)+\beta_0^{\,2}=0$$

the solution of which is:

$$\beta_0=\frac{-(\lambda-\mu_1-\mu_2)-\sqrt{(\lambda-\mu_1-\mu_2)^2-4\mu_1\mu_2(1-\rho)}}{2}. \tag{7-44}$$

One has too $\varphi_1'(\beta_0)=\frac{\alpha_1\mu_1}{(\mu_1-\beta_0)^2}+\frac{\alpha_2\mu_2}{(\mu_2-\beta_0)^2}$.

Hence the distribution function of the waiting time:

$$P(>x)\approx\frac{(1-\rho)}{\lambda\left(\frac{\alpha_1\mu_1}{(\mu_1-\beta_0)^2}+\frac{\alpha_2\mu_2}{(\mu_2-\beta_0)^2}\right)-1}e^{-\beta_0 x} \tag{7-45}$$

The reader may verify that, taking $\mu_1=\mu_2=\mu$, one finds as needed $\beta_0=(1-\rho)$, and:

$$P(>x)=\rho e^{-\mu(1-\rho)x}$$

which is the expression to be expected, for the exponential service.

7.7.3.2. *Application to G/M/1 queue*

Let us now study the "symmetrical" case of the M/G/1 queue, still beginning with the Pollaczek method.

This time let us start again from the general expression (7-38) of Pollaczek.

This gives:

$$\phi(z)=\exp\left\{-\frac{1}{2\pi i}\int_{-i\infty-0}^{+i\infty-0}\ln\left[1-\varphi(-\zeta)\right]\frac{zd\zeta}{\zeta(z+\zeta)}\right\}, \text{ with}$$

$$\varphi(-\zeta)=E(e^{-\zeta T_n})E(e^{\zeta Y_n})=\varphi_1(-\zeta)\varphi_2(\zeta)$$

or again:

$$\phi(z) = \exp\left\{-\frac{1}{2\pi i}\int_{-i\infty-0}^{+i\infty-0} \ln[1-\varphi_1(-\zeta)\varphi_2(\zeta)]\frac{zd\zeta}{\zeta(z+\zeta)}\right\}$$

As the service is governed by an exponential law with a rate μ, this gives:

$F_1(t) = 1 - e^{-\mu t}$ and $\varphi_1(\zeta) = \dfrac{\mu}{\mu - \zeta}$

and so:

$$\phi(z) = \exp\left\{-\frac{1}{2\pi i}\int_{-i\infty-0}^{+i\infty-0} \ln\left[1-\frac{\mu\varphi_2(\zeta)}{\mu+\zeta}\right]\frac{zd\zeta}{\zeta(z+\zeta)}\right\}$$

It will be noted that:

$$\frac{zd\zeta}{\zeta(z+\zeta)} = \left(\frac{1}{\zeta} - \frac{1}{z+\zeta}\right)d\zeta = d\ln\frac{\zeta}{z+\zeta}.$$

This gives:

$$\phi(z) = \exp\left\{-\frac{1}{2\pi i}\int_{-i\infty-0}^{+i\infty-0} \ln\left[1-\frac{\mu\varphi_2(\zeta)}{\mu+\zeta}\right]d\ln\frac{\zeta}{z+\zeta}\right\}$$

The integration is carried out by parts (the form is $u.dv$), and we thus obtain:

$$\phi(z) = \exp\left\{\frac{1}{2\pi i}\int_{-i\infty-0}^{+i\infty-0} \ln\frac{\zeta}{z+\zeta}\frac{d}{d\zeta}\ln\left[1-\frac{\mu\varphi_2(\zeta)}{\mu+\zeta}\right]d\zeta\right\},$$

or yet, $\phi(z) = \exp\left\{\dfrac{1}{2\pi i}\displaystyle\int_{-i\infty-0}^{+i\infty-0} \ln\frac{\zeta}{z+\zeta}\frac{d}{d\zeta}\ln\left[\frac{\mu+\zeta-\mu\varphi_2(\zeta)}{\mu+\zeta}\right]d\zeta\right\}$.

Let us consider the last expression under the integral: $(\mu+\zeta)-\mu\varphi_2(\zeta)$, and let us apply Rouché's theorem to $\mu+\zeta-\mu\varphi_2(\zeta)$, (sum of two functions).

Inside the circle with the centre $(-\mu)$, and of radius μ, thus $R(\zeta) < 0$, we have $|\zeta+\mu| < \mu$.

According to Rouché's theorem applied to the expression on the circle considered, it is deduced that it only has in the circle a single root ζ_1 and thus also in the whole plane $R(\zeta)< 0$.

We may therefore write:

$$\phi(z) = \exp\left\{\frac{1}{2\pi i}\int_{-i\infty-0}^{+i\infty-0} \ln\frac{\zeta}{z+\zeta}\frac{d}{d\zeta}\ln\left[\frac{\zeta-\zeta_1}{\mu+\zeta}\right]d\zeta\right\},$$

or $\phi(z) = \exp\left\{\frac{1}{2\pi i}\int_{-i\infty-0}^{+i\infty-0} \ln\frac{\zeta}{z+\zeta}\left[\frac{1}{\zeta-\zeta_1}-\frac{1}{\mu+\zeta}\right]d\zeta\right\}.$

Let us apply the residue theorem to both poles:

ζ_1 such that $\mu+\zeta_1-\mu\varphi_2(\zeta_1)=0$,

and: $\zeta_2=-\mu$

We obtain:

$$R_1 = \ln\frac{\zeta_1}{z+\zeta_1},\quad R_2 = -\ln\frac{\mu}{\mu-z}$$

And thus finally by now denoting $\phi_w(z)$ the characteristic function of the waiting time, we obtain:

$$\phi_w(z) = \frac{\mu-z}{\mu}\frac{\zeta_1}{\zeta_1+z}$$

A fundamental relation: the characteristic function of waiting for the G/M/1 server.

This expression is also written:

$$\phi_w(z) = \frac{(\mu-z)(1-\varphi_2(\zeta_1))}{\mu-z-\mu\varphi_2(\zeta_1)} \tag{7-46a}$$

which becomes, by calling: $\sigma=\varphi_2(\zeta_1)=\frac{\mu+\zeta_1}{\mu}$,

$$\phi_w(z) = \frac{(\mu - z)(1-\sigma)}{\mu - z - \mu\sigma} \tag{7-46b}$$

This is a remarkable result as we find an expression similar to that of the M/M/1 queue, this time σ playing the role of ρ.

We deduce from this immediately the expression for the waiting time distribution:

$$P(>x) = \sigma e^{-\mu(1-\sigma)x} \tag{7-46c}$$

Just as for the M/G/1 system, we can apply these results to several practical cases approximating most of realistic cases.

M/M/1 queue

Before considering the more general cases, let us again apply first the previous result to the M/M/1 case.

As the arrival instants are governed by an exponential law, we have:

$\varphi_2(-z) = \dfrac{\lambda}{\lambda + z}$,

the solution sought is such that: $\mu - \beta_0 - \mu\sigma = \mu - \beta_0 - \mu \dfrac{\lambda}{\lambda + \beta_0} = 0$,

hence $\beta_0 = \mu - \lambda$ and finally $\sigma = \dfrac{\lambda}{\mu} = \rho$,

and: $P(>x) = \rho e^{-\mu(1-\rho)x}$

We find again the result already established.

Application to D/M/1 case

This is "symmetrical" to the M/D/1 case. In this case we have:

$\varphi_2(z) = e^{z/\lambda}$, where λ is the arrival rate (lower than μ)

the solution sought is such that:

$$\mu - \beta_0 - \mu\sigma = \mu - \beta_0 - \mu e^{-\beta_0/\lambda} = 0, \tag{7-47}$$

We have: $\sigma = 1 - \beta_0 / \mu$ and $P(>x) = \sigma e^{-\mu(1-\sigma)x}$, or of course:

$$P(>x) = \sigma e^{-\mu(1-\sigma)x}$$

Application to H2/M/1 case

This is in a sense the opposite case to the D/M/1 case, in terms of the variation factor (greater than 1), with the M/M/1 case being an "intermediate" case.

In this case we have: $\varphi_2(z) = \dfrac{\alpha_1 \lambda_1}{\lambda_1 - z} + \dfrac{\alpha_2 \lambda_2}{\lambda_2 - z}$, where λ is the arrival rate (less than μ) such that: $\dfrac{1}{\lambda} = \dfrac{\alpha_1}{\lambda_1} + \dfrac{\alpha_2}{\lambda_2}$.

The solution sought is such that:

$$\mu - \beta_0 - \mu\sigma = \mu - \beta_0 - \mu\left(\frac{\alpha_1 \lambda_1}{\lambda_1 - \beta_0} + \frac{\alpha_2 \lambda_2}{\lambda_2 - \beta_0}\right) = 0. \tag{7-48}$$

We have: $\sigma = 1 - \beta_0 / \mu$ and $P(>x) = \sigma e^{-\mu(1-\sigma)x}$, or of course:

$$P(>x) = \frac{\mu - \beta_0}{\mu} e^{-\beta_0 x}.$$

7.8. Queues with priorities

If a class of clients has a particular level of urgency, it is possible in most cases to set up priority mechanisms. For example, a processor must be able to process very rapidly tasks relating to vital security functions, even if this means delaying less critical tasks. Alternatively, a router will give preference to a packet whose quality of service requirements (in terms of delay) involve greater constraints. This service mechanism will be represented by a model with *priorities*.

The normal case, where clients are served in the order of their arrival, is the FIFO (*First In – First Out*) discipline. Under the LIFO (*Last In – First Out*) discipline, the last to arrive is the first one to be served. In the case of the HoL discipline (*Head of Line*), each client belongs to a priority level, clients of a given level are served after the ones of higher level have been served.

A further distinction is drawn between two variants in the analysis of priority queues, depending on whether they incorporate a *pre-emption* mechanism. In the case of pre-emptive priority, a client with a higher priority than the client in service will interrupt the latter's service, and immediately take the latter's place. On completion of the handling of the priority client, the client with a lower priority will resume his/her place in service (and depending on the case, start from the beginning or continue his/her service, referred to as "pre-emptive resume").

7.8.1. *Work conserving system*

Two viewpoints are considered in the analysis of systems with priorities: that of the client and that of the server. Understandably, in some cases the choice of client is indifferent to the server, in the sense that this choice does not change the total amount of work it must handle. The consequence is that at each instant, the server is capable of accounting for the quantity of work it is offered (*unfinished work*) : the service time still to be provided to the current client, to which is added the sum of the services of clients waiting at the instant in question. This quantity can be evaluated at a fixed instant, but it is only meaningful if the remaining work does not depend on the next decision! So, let us denote as τ the remaining (unfinished) work at instant t. Let us stop the arrival process at this instant, and observe the random behaviour of the system: the server will work continuously until $t+\tau$, whatever the scheduling decisions taken after t.

A *work conserving* system is a system of this type in which the unfinished work does not depend on the choice discipline applied by the server.

Although this is a common situation, counter-examples are a multi-queue system with switchover time; a system with impatience; a system in which the service time depends on waiting (ageing), etc. For most of these systems, it is not even possible to calculate this work at a given instant.

M/G/1 conservation law

It is clearly the case that, whereas the work remaining does not depend on scheduling, the waiting time will depend on it. The conserving property operates "from the server's viewpoint". For customers, we will see what is involved.

A queue receives several different flows. We note as λ_j the flow intensity of clients of *j* type (we speak of *classes of clients*) and as W_j the mean waiting time they experience. We note as λ the total flow submitted to the server. It is possible, experimentally for example, to measure waiting time while being unaware of the existence of client classes. We note as $\overline{W}$ the corresponding quantity. This is a weighted sum, with the weighting causing the intervention of a proportion of each flow:

$$\overline{W} = \sum_j \frac{\lambda_j}{\lambda} W_j$$

The importance of this weighted mean time is that it may be impossible to measure any other quantity – let us assume for example a measurement process unable to differentiate between classes.

CONSERVATION THEOREM.– Let us imagine a queue receiving clients belonging to different classes and served in accordance with any priority mechanism, which is conserving and non-pre-emptive (which may or may not be based on classes). The conservation law is as follows:

$$\sum_i \rho_i W_i = \frac{\rho}{1-\rho} W_0 = Cste,$$

where $W_0 = \sum_i \lambda_i \frac{E(s_i^2)}{2} = \sum_i \rho_i \frac{E(s_i^2)}{2E(s_i)}$ (7-49)

The index *i* runs on client classes; $\rho_i = \lambda_i W_i$ represents the load in class *i* (W_i is the mean waiting time it experiences); $\rho = \sum \rho_i$ is the total load offered to the server.

The formula implies that the expression does not depend on the service discipline; note that the sum of the waiting times is weighted as ρ_i, and not by the λ_i as was the case for the mean wait $\overline{W}$.

Special cases:

Reminder:

– exponential law: $E(s^2)=2E(s)^2$, that is $W_0=\sum \rho_i E(s_i)$

– constant time: $W_0=\frac{1}{2}\sum \rho_i E(s_i)$

Let us assume two flows, with different service times, but without priority. Their waiting times will therefore be identical, and the application of the formula gives:

$$W_1=W_2=\frac{1}{1-\rho}\sum \lambda_i E(s_i^2)$$

we return in this case to the PK formula.

PROOF OF THE CONSERVATION LAW.– We observe the system at an instant t, and we count $n(i)$ clients of the class i (the classes are numbered: 1, 2, ..., P); the client whose service is in progress requests additional service with a time of x_0. The unfinished work, (the name explains the traditional notation U), is thus:

$$U(t)=x_0+\sum_{i=1}^{P}\sum_{k=1}^{n(i)} x_{k,i}$$

We have noted as $x_{k,i}$ the service time that will be requested by client k in class i. Taking the average,

$$E(U)=W_0+\sum_i E\left[\sum_k x_{k,i}\right]$$

Clients of the same class are assumed to be identical, i.e. to have the same service law, such that: $E\left[\sum_k x_{k,i}\right]=E(n_i)E(s_i)$. Bearing in mind that the use of the Little formula links the mean number of clients to their mean waiting time. Thus:

$$E(U)=W_0+\sum_i \rho_i W_i$$

Now, the unfinished work is independent of the scheduling mechanism – this is a direct consequence of the work conserving system hypothesis. Thus its mean value will be the same as with a service in the order of arrivals, a case for which all W_i are equal, and in which $E(U) = W_0/(1-\rho)$. Putting this expression into the previous one supplies the result sought.

Priority between identical clients

We now assume that all clients have identical characteristics: same service time distribution (no just merely the same average value). Whatever the choice mechanism, provided that the system is work conserving, the choice made by the server has no influence on the total occupation of the queue; in other words, the distribution of the total number of clients queueing does not depend on the discipline. In particular, the mean number of clients is independent of the discipline. Therefore, the Little formula asserts that the mean waiting time does not depend on the discipline (but the distribution of the waiting does depend on it). One can check that this result is in agreement with the work conserving relation (7-49). For example, FIFO, LIFO, and random choice give the same mean waiting time.

7.8.2. *The HOL discipline*

It is assumed that there are P classes of clients, each class groups together clients with identical statistical characteristics (but the characteristics differ from one class to another). The server processes first clients of the highest priority class (by convention, this is noted as 1), and then, if there are no more priority clients in the queue, clients of lower priority (noted 2), etc., up to clients of class P, of the lowest priority. The service mechanism is work-conserving.

We introduce the notation:

$$\sigma_k = \sum_{n=1}^{k} \rho_n$$

This is the sum of the partial loads contributed by classes with a priority greater than or equal to k. This is therefore the load "seen by" a client of class k, as it overtakes the clients of lower classes. Note however that a client of a given class will however be hindered by a client of a lower priority class in service on his/her arrival (as the discipline is not *pre-emptive*).

Then we have:

$$W_1 = \frac{W_0}{1-\sigma_1},$$
$$W_k = \frac{W_0}{(1-\sigma_k)(1-\sigma_{k-1})}, \quad k>1 \tag{7-50}$$

In fact, for a non-pre-emptive discipline,

$$W_j = W_0 + \sum\nolimits_{i\le j} E(s_i)\overline{N_{ij}} + \sum\nolimits_{i<j} \overline{M_{ij}}$$

W_j is the mean waiting time of clients in class j, and index i runs over classes $(i=1,\ldots N)$; $\overline{N_{ij}}$ represents the mean number of clients in class i already present on the arrival of the client of type j and who will be served before him/her (in particular, the client does not overtake those of his/her own class). $\overline{M_{ij}}$ represents the mean number of clients in class i arriving during the wait of the client of type j and who will be served before him.

For our discipline, the use of the Little formula gives:
$\overline{N_{ij}} = \lambda_i W_i$, $i=1,\ldots j$, and $\overline{N_{ij}}=0$, $i>j$ (our test client overtakes clients in lower priority classes);
$\overline{M_{ij}} = \lambda_i W_j$, $i=1,\ldots j-1$, and $\overline{M_{ij}}=0$, $i\ge j$ (only classes with strictly higher priority classes will overtake him/her). Finally:

$$W_j = W_0 + \sum\nolimits_{i=1}^{j} \rho_i W_i + \sum\nolimits_{i=1}^{j-1} \rho_i W_j$$

that is: $W_j(1-\sigma_{j-1}) = W_0 + \sum\nolimits_{i=1}^{j} \rho_i W_i$

From this formula, we calculate first W_1, then W_2, etc.

7.9. Use of approximate methods

The "exact" methods presented up to this point enable strict mathematical resolution of a limited number of problems. Real world systems rarely come into these models. Use must therefore be made of approximations. To this end, several approaches will be possible. The first attitude consists of looking for other mathematical models, enabling, at the price of numerical calculations, often difficult to resolve, exact models of very great complexity: for example, a general law will be represented by a superimposition of exponential laws (*Cox laws*), and we will use a Markovian analysis, giving numerical results.

The second consists of "forgetting" some details of actual functioning that models cannot represent, and latching on to a known model (M/G/1 queue, HOL priority, etc.). The principle of this approach is to represent in this model the "essence" of the behaviour, and the results will be orders of magnitude, which are more or less exact.

Finally, it will be possible to implement approximate results, without any solid theoretical justification, but which one knows give satisfactory results: this is the case of the Martin approximation, presented below, or results on cyclic servers. This is also the principle of diffusion approximations, or so-called "heavy traffic" approximations (see Chapter 9, M/D/1, N/D/D/1 queues).

7.9.1 *Some approximate formulae*

At the present time, one of the best and most universal formulae is the *Allen-Cunnen formula*. It enables the estimation of the mean waiting time, for a multiserver system with general arrival and service laws. Let us denote:

– c_s^2, the squared coefficient of variation of the service time;

– c_a^2, the squared coefficient of variation of the interarrival time;

– $A = \lambda / \mu$, the traffic offered, and $\rho = A / R$ the utilisation factor of each server;

– $C(A,R)$ is the waiting probability of M/M/R:

$$C(A,R) = \frac{A^R / R!}{\frac{A^R}{R!} + (1-\rho)\sum_{n<R} \frac{A^n}{n!}}$$

Thus,

$$\frac{E(W)}{E(S)} \cong \frac{C(A,R)}{R(1-\rho)} \times \frac{C_s^2 + C_a^2}{2} \tag{7-51}$$

This is in fact the formula for the M/M/R queue (an exact formula in this case), which is corrected by a factor that takes into account the arrival and service law coefficients of variation. It is worth noting that:

– the formula is *exact* for the M/M/R queue;
– it is *exact* for the M/G/1 queue (it reduces to the Pollaczek-Khinchine formula).

Quantiles calculation

"Quantiles" is the term given to an essential concept: the clients of a waiting system are sensitive not to a great extent to mean waits, but rather to "inadmissible" waits. In this respect, the reader will recall that the quality of service standards not only specify mean values to be complied with, but also values at x% (see Chapter 2). Quantile estimation is a delicate process, as the distribution is rarely known explicitly, with the exception of M/M/c systems. Use is quite often made of an empirical formula, known by the name *Martin formula*:

$$t_{90} = E(T) + 1.3\sigma_T$$
$$t_{95} = E(T) + 2\sigma_T$$

The formula is in fact very general: it is based on the resemblance that inevitably exists between any "good" distribution and a *Gamma law* (see Chapter 4). This resemblance excludes non-unimodal or non-continuous distributions. It will be preferably applied to sojourn time, which has no discontinuity at the origin.

As an example, the table below shows the case of the 90% quantile of the holding time of the M/D/1 queue (i.e. the time having 1 chance out of 10 of being reached or exceeded):

Load	**Exact value**	**Approximation (Martin's formula)**
0.3	1.85	1.73
0.5	2.5	2.35
0.7	4.1	3.6
0.8	6.0	5.1
0.9	11.8	9.6

7.10. Appendix: Pollaczek transform

We had:

$$\phi_{n+1}(z)=\frac{1}{2\pi i}\int_{C\zeta}\phi_n(-\zeta)\varphi(-\zeta)\frac{z}{\zeta(\zeta+z)}d\zeta$$

We can also write directly for the wait its characteristic function as follows:

$$\phi_n(-z)=\int_{0-}^{\infty}e^{-zt}dW_n(t/w_0)=E(e^{-zW_n}/w_0)$$

and in particular for the first client:

$$\phi_0(-z)=E(e^{-zW_0})=e^{-zW_0}$$

Let us introduce the generating function:

$$\phi(z,v)=\sum_{n=0}^{\infty}\phi_n(-z)v^n,$$

which, applied to the integral studied (after development and summation), gives the relation:

$$\phi(z,v)-\frac{v}{2\pi i}\int_{C\zeta}\phi_n(\zeta,v)\varphi(-\zeta)\frac{z}{\zeta(z-\zeta)}d\zeta=e^{-zW_0}.$$

In the case in which $W_0=0$, the solution of this equation is:

$$\phi_0(z,v)=\exp\left\{-\frac{1}{2\pi i}\int_{C\zeta}\ln[1-v\varphi(-\zeta)]\frac{z}{\zeta(z-\zeta)}d\zeta\right\}$$

With $0<R(\zeta)<R(z)$. The verification, which is relatively complex, is performed by inspection by expanding the logarithm in a Taylor expansion. Reference may be made to [LEG 62, SYS 93] for a detailed demonstration.

By application of the residue theorem at $\zeta=0$, where $\frac{\ln[1-v\varphi(-\zeta)]}{\zeta}\approx\frac{(1-v)}{\zeta}$, this gives

$$\phi_0(z,v) = \frac{1}{1-v}\exp\left\{-\frac{1}{2\pi i}\int_{-i\infty-0}^{+i\infty-0} \ln[1 - v\varphi(-\zeta)]\frac{z d\zeta}{\zeta(z+\zeta)}\right\}$$

The stationary limit process being independent of the initial condition W_0, we come to:

$$\phi(z) = \exp\left\{-\frac{1}{2\pi i}\int_{-i\infty-0}^{+i\infty-0} \ln[1 - \varphi(-\zeta)]\frac{z d\zeta}{\zeta(z+\zeta)}\right\} \quad ,$$

which is the characteristic function of the waiting that was sought, with $\varphi(-\zeta) = \varphi_1(-\zeta)\varphi_2(\zeta)$, $R(z) \le 0$. And bearing in mind that to exist the stationary limit process requires compliance with the condition $|\varphi(-\zeta)| \le \varphi[-R(z)] < 1$.

Recalling that the derivatives at z =0 of the characteristic function give us the moments, or that the expansion in Taylor series of $\ln\phi(z)$ gives us the cumulants, we obtain in particular the mean wait, and more generally the cumulants:

$$W = -\frac{1}{2\pi i}\int_{-i\infty-0}^{+i\infty-0} \ln[1 - \varphi(-\zeta)]\frac{d\zeta}{\zeta^2}$$

$$C_n = \frac{(-1)^n n!}{2\pi i}\int_{-i\infty-0}^{+i\infty-0} \ln[1 - \varphi(-\zeta)]\frac{d\zeta}{\zeta^{n+1}}$$

The non-waiting probability is also obtained directly from the characteristic function by growing R(z) to infinity by negative values. The waiting probability is its complement.

$$P(w=0) = \exp\left\{-\frac{1}{2\pi i}\int_{-i\infty-0}^{+i\infty-0} \ln[1 - \varphi(-\zeta)]\frac{d\zeta}{\zeta}\right\}$$

8

Simulation

The previous chapters have given prominence to the power and efficiency of the analytical approach to performance evaluation. However, many realizations make use of such specific mechanisms so that their analysis is beyond the scope of mathematical tools. In that case, the simulation method provides an efficient way to overcome the difficulty.

A simulation experiment aims at reproducing the dynamic behaviour of the system (customers, servers, etc) through a computer program, most often run on specific software tools. On this software model, observations are performed, which give the figures of interest, such as mean or variance of delays, loss probabilities, etc. Clearly, the goal is not to reproduce exactly the microscopic level of detail of the original system, but to take account of those peculiarities in the mechanisms responsible of the overall behaviour. It takes a careful analysis of the system, and a good understanding of the queueing phenomena to extract the strict set of details compatible with the required degree of precision and the budget set for the study, expressed both in terms of the time needed to elaborate and validate the model as well as the duration of each simulation run.

Actually, simulation happens to offer quite a powerful approach for the engineer: it not only allows a quantitative analysis of complex systems to be made, but it provides a *software prototype* of the system under development, whose role is the same as the experiment for the researcher. Indeed thanks to simulation one can observe and understand complex behaviours of the system, hypotheses can be formulated and verified, alternative technological solutions may be experimented with, eventually to give rise to an improved solution and its model.

However, simulation is generally not the definite answer to the performance problem. The large variety of external conditions (especially the traffic mix) and the evolution of the services offered are such that synthetic and tractable analytical tools, even approximate, are mandatory for the equipment provider or the operator. Actually, it would be quite expensive to run the simulation program for each new set of parameters or for each new customer demand. The real goal of a simulation experiment is the derivation of a set of rules or models, as simple and robust as possible.

Two main approaches have been developed, referred to as *roulette simulation* and *discrete-event simulation*. Both make use of random number generators as the basic tool to emulate the environment (customers, occurrence of failures, etc.). Another simulation method is sometimes described as *continuous simulation*. In the latter, the system state is re-evaluated at short intervals of time, just as in solving a differential equation by numerical integration.

8.1. Roulette simulation

Although use of this technique is somewhat restricted, it offers a powerful method, especially in terms of speed of execution.

The principle is quite simple:

Assume the system can be represented through a set of denumerable states enjoying the "memoryless property". For instance, the system could be a multistage subscriber concentrator, where the occupancy varies at each beginning and end of communication with exponentially distributed communication duration; or it could be a cluster of redundant processors, where failures occur according to a Poisson process, etc. The point is that one is able to calculate the transition probabilities from each state to the next one. Simulating such a system amounts to reproducing the sequence of transitions, by drawing at each step the next state to be reached. Simultaneously, all necessary counting is performed, allowing estimation of state probabilities. To the evidence, this technique is fairly well suited to Markovian systems, such as those studied in Chapters 6 and 7. Especially, if service durations or time intervals between breakdowns, etc., are exponentially distributed, the probabilities are the coefficients of the transition matrix already introduced (in fact, roulette simulation is just a kind of "empirical" resolution of the state equations summarised in the matrix).

Example

We consider the simulation of a subscriber concentrator (or a traffic sources multiplexer) offering R servers to N Poissonian sources, generating calls at a rate λ, with duration exponentially distributed with average $1/\mu$. This system has already been considered in Chapter 7: it is the "Engset problem". The state transition diagram is given in Figure 8.1:

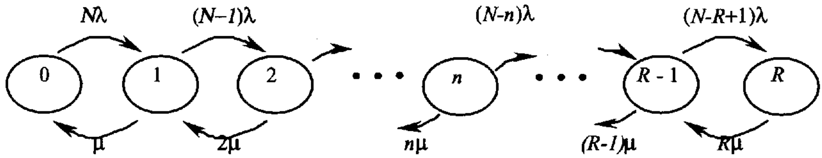


Figure 8.1. *State transition diagram of the concentrator*

The "state" stands for the number of calls in progress in the concentrator. Elementary events are:

- The arrival of a new request, which changes the state from n to $n + 1$ if $n < R$. In state n, $N - n$ sources only are idle and able to issue a request: the birth rate is $(N-n)\lambda$.
- The end of a call, with a rate $n\mu$, which changes the state from n to $n - 1$.

Given the state n the transition probabilities to adjacent states are respectively:

$$P(n \rightarrow n-1) = \frac{n\mu}{n\mu+(N-n)\lambda}, \quad P(n \rightarrow n+1) = \frac{(N-n)\lambda}{n\mu+(N-n)\lambda}. \tag{8-1}$$

Clearly the events are complementary, the probabilities sum up to 1. The simulation experiment runs as follows:

a) Draw a random number r, uniformly distributed in the interval (0,1).

b) Test if $r < \frac{n\mu}{n\mu+(N-n)\lambda}$. If yes (a departure occurs), move the system to the state $n - 1$. If no (an arrival occurs), move to $n+1$.

c) Repeat operations a) and b) until the number of events reaches the limit assigned to the experiment.

As it will be clear, as compared with the classical discrete-event approach presented in next section, the present method enjoys simplicity and efficiency, especially at run time, as the code to be processed reduces to execute simple calculations and to update a few counters. In the case of simulating a large network, for instance, it reduces to choosing the destination, looking for an available path, according to the routing algorithm and the architecture, and marking the corresponding resources as busy. Also, the occurrence of an end of service depends on the total number of ongoing connections. Powerful computers currently available make it possible to investigate fairly large structures this way.

However, this real simplicity raises several issues. The first two are common to all simulation techniques: the accuracy of the result, function of the simulation duration, and the measurement technique. The reader is directed to Chapter 5 for an in depth treatment of these issues, of which the main aspects are summarized below.

The main drawback with roulette simulation is that it does not incorporate any notion of "time". As already mentioned, the memoryless property of all the processes is central to the method, so that non exponential service or interarrival durations are impossible to take into account. However, it is always possible to approximate arbitrary processes, replacing them by a combination of exponential distributions (e.g. Erlang-k distribution, see Chapter 4).

More important, the principle of the method makes it unable to measure any delay, such as waiting times, busy periods, etc. Here too various tricks may be used, e.g. counting the number of events between significant changes of states (possibly adding "null" events occurring according to an exponential distribution), and deducing from it the corresponding duration.

In fact, as soon as time is the central parameter of concern, discrete-event simulation will be the preferred approach.

8.2. Discrete-event simulation

This technique is commonly used in all specialized software packages, e.g. Simula, Simscript, SES-Workbench, QNAP, OPNET).

As previously, the system under study can be described in terms of the states among which it progresses. Formally, a state is a set of *state variables* (number of busy servers, number of clients in a queue, date of the next arrival or departure, etc.) giving all the information necessary to predict the future evolution (on a probabilistic basis). Especially, the time to the next transition is contained in the state variables. Each transition is caused by an *event* (arrival, failure, etc.). The events occur at *discrete epochs*, and are described in the simulation program. So, the evolution of the system is fully reproduced, but only by jumping from one date of event to the following one, as nothing occurs between events from the simulation standpoint.

As opposed to roulette simulation, this method provides a full mastering of time and allows its precise measurement. Also, the gain as compared with continuous simulation is obvious, since in this last method much time is devoted to increase the clock without effective action (since "nothing occurs between events").

However, this method requires estimating the date of a future event, which is obtained by generating random intervals according to a given probability distribution. For instance, when a service of duration exponentially distributed begins, its duration is drawn using the properties of the exponential distribution, determining the date of occurrence of the event "end of service". This operation is even simpler when the service is of constant duration. These techniques are further discussed in section 8.4.

As events occur on a discrete-time basis (no simultaneous events) they are processed chronologically, in the following way: the *event notices* describing them (date of occurrence, actions to be executed) are stored in a timetable (*event list, sequencing set*) in order of increasing dates. The simulation logic scans the event in head of list, jumps immediately to the date of the first event and executes the corresponding actions (new request, end of session, failure of an element, etc.). Most often, the processing results in generating new future events (a beginning of service generates an end of service) which are inserted at the right location in the event list. Then the logic scans again the head of list, etc.

The efficiency of such a process depends first on the complexity of the actions to be executed, but also on the number of simultaneous events to be handled in the event list. Dedicated software tools incorporate sophisticated and powerful algorithms for managing the event list: inserting or extracting events (the so-called *synchronization kernel*).

The event list

Let us briefly describe a possible structure for the event list. It can be built using a circular twoway list, where each entry (event notice) is linked by forward and backward pointers to the next and previous event notice.

Each event notice is a data structure storing the reference of the event, its date of occurrence, a pointer to the code simulating it, pointers to the previous event and to the next one (see Figure 8.2). Adding an event at a given date is done by inserting a new notice and updating the pointers (sophisticated techniques are used to optimize the process).

As compared with continuous simulations, where the system keeps increasing time and testing possible actions, here the computer executes only the pieces of codes corresponding to the events. The gain in terms of run time is important, and such a method allows the building of powerful simulators, suited to traffic analysis of machines with more and more capacity (e.g., switches and call servers handling several millions of calls per hour, or routers handling packets).

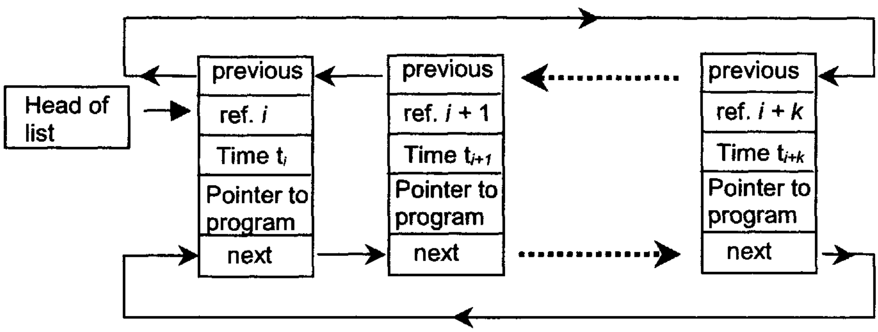


Figure 8.2. *An event list organized as a circular buffer*

The techniques to manage event lists are not restricted to classical simulations, and the performance engineer may have the need to create such a structure for his own machines, and especially for *load tests* (in labs or for field trials, see Chapter 10). Actually, consider the realization of a call generator, creating traffic for testing a system under real size load. At a given date a call is initialized by an *open* message sent to the system (request for call set up, e.g.). The generator then stands waiting for an answer from the system (connection set up) initializing the communication, communication which will be interrupted by the sending to the system of a *close* message, generated after a random duration. Another open message will then be generated at a date depending on the load level, etc.

As can be seen, an event list storing the scheduled dates of beginnings and ends of calls is perfectly suited to the design of this call generator.

Managing event lists is one of the examples emphasising the interest, for the specialist, of having a sound understanding of the techniques of simulation languages, as they may be re-used for the design and optimisation of all real-time systems.

8.3. Measurements and accuracy

Two difficult issues are related to simulation, whatever the technique in use (roulette or discrete-event approach): the precision of the measurement and confidence to put in the results, and the technique of observation allowing measurement (of sojourn time in various states, of number of events, etc.). The reader is referred to Chapter 5 for further developments.

Here are a few basic principles.

8.3.1. *Measurements*

The ideal measure should supply a *random* sampling of the states the system goes through during the simulation. One must always keep in mind this fundamental rule, as numerous traps may distort the observation process. For instance, think of how to measure queue lengths in the case of deterministic or bulk arrivals, or to measure the load of a cyclic server, etc.(see Chapter 7 and the Pasta property).

Fortunately, in most frequent cases it is possible to describe the phenomenon to be observed in precise terms of queueing theory. The simulation experiment then simply implements the measurement corresponding to the definition. Remember however that the observation cannot tell more than what has been defined, including all the *a priori* implicit in most of our reasoning. Experiments, using random observations are always fruitful in this perspective (they may make some unsuspected property appear).

These issues are less crucial in the case of roulette simulation, as the "memoryless property" allows one to simply increase the number of visits in a given state and the total number of transitions, the state probability being just the ratio of these quantities.

8.3.2. *Accuracy*

Regarding the accuracy of the measurement, it is simply reduced to a question of *estimation*, as we now explain.

Assume one wants to estimate some state probability (for instance the probability of having n busy servers, or the probability of more than n customers in a buffer). If the estimation is made on series of k time units (event driven simulation), or of k transitions (roulette simulation), the results can be displayed in the form of curves, similar to those in Figure 8.3. The value observed is the ratio: number of outcomes over k. The parameter k must be taken large enough so that two successive series may be seen as approximately independent.

These curves are called *trajectories* of the measurement process. Two different curves are represented, corresponding to two different initial states of the simulation.

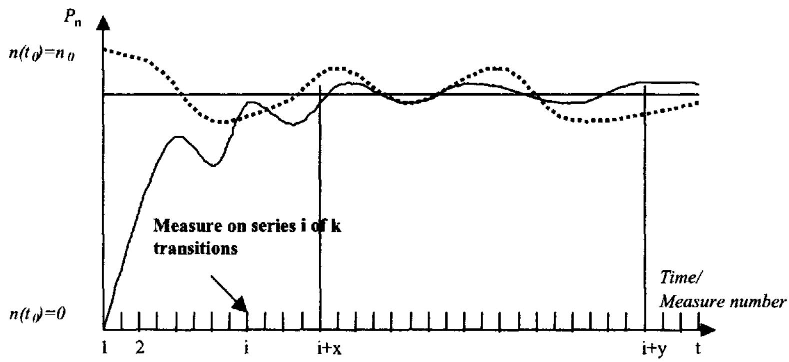


Figure 8.3. *Trajectories of the state observed*

After a transient period, the process goes to a stable limit as the number of measurements increases. This is because the system reaches a stationary regime so that the state probabilities have constant values and the measurement process converges toward these values. This represents the situation of almost all systems to be considered. Based on the evidence, if the initial state is closer to the stationary regime the results are more rapidly significant.

As a matter of fact, sophisticated techniques have been developed that lead to reduction in the simulation duration, for a given accuracy, especially in the case where *rare events* are to be observed. The idea is to "restart" the simulation when the trajectory of the system state reaches a zone of interest (i.e. where the critical states are likely to be found). See [VIL 91] for further developments.

Let us consider the trajectory once the stationary regime has been reached (after rank i, say). Then, the classical methods for estimating means and variances apply. The set of values observed from the measures of rank $(i+x)$ to $(i+y)$ is our *sample*. However, we clearly have no knowledge about the probability distribution of the measures, and this implies several assumptions. Most of the time, the process is considered as being stationary, and the classical approach of the estimation gives the answer (point estimators as well as confidence intervals). In most cases the normal assumption holds (as soon as $n > 30$): this is the consequence of the *central limit theorem*, see Chapter 5.

Especially, the classical results for the mean value give:

$$P\left(\hat{m}-\frac{\sigma}{\sqrt{n}}u_{1-\alpha_2} < m < \hat{m}-\frac{\sigma}{\sqrt{n}}u_{\alpha_1}\right)=1-\alpha, \tag{8-2}$$

n being the sample size, $\hat{m}$ the observed value, $1-\alpha$ the confidence level and σ the standard deviation of the population. An unbiased estimator for this quantity, usually unknown, is

$s=\sqrt{\frac{n}{n-1}}\hat{\sigma}$, $\hat{\sigma}$ being the values observed on the sample.

So finally, $\hat{m}-u_{1-\alpha_2}\frac{s}{\sqrt{n}} < m < \hat{m}+u_{\alpha_1}\frac{s}{\sqrt{n}}$,

or yet $\hat{m}-t_{1-\alpha/2}\frac{s}{\sqrt{n}} < m < t_{1-\alpha/2}\frac{s}{\sqrt{n}}$, (8-3)

for a risk symmetrically distributed. For instance, the 95% and 90% centred confidence intervals (values in common use) are:

$$\left[\hat{m}-1{,}96\frac{s}{\sqrt{n}};\hat{m}+1{,}96\frac{s}{\sqrt{n}}\right] \text{ and } \left[\hat{m}-1{,}65\frac{s}{\sqrt{n}};\hat{m}+1{,}65\frac{s}{\sqrt{n}}\right]. \tag{8-4}$$

Remember that the above result holds also for samples of small size n, provided the coefficients t are taken from the Student-Fischer table, in the column corresponding to the risk level α.

This method (sometimes referred to as *batch means*) is of great importance, as it visualizes both the transient period and the optimal measurement interval – when the stationary regime is reached. Taking only the cumulated result may lead to erroneous predictions, related for instance to a biased measurement during the transient period or also to the use of random number generators with too short periods, provoking non stationary behaviours. This last remark applies especially to "home made" generators, as modern simulation languages provide correct enough generators.

One can also point out the interest of the measure of several states, in order to obtain the whole probability distribution, and for different conditions, e.g. different load levels. This allows one to draw curves and thus to extrapolate the results for low probabilities. This can reinforce also the confidence towards the measures as the curve obtained should present a plausible shape.

This is exemplified in Figure 8.4, built from the simulation of the N/D/D/1, queue, and which exhibits the usual aspect of simulation results. The knee observed around 10^{-9} indicates the limit beyond which results are no longer valid, because of not enough observations. But the shape allows reasonable extrapolations for lower probability values, if needed.

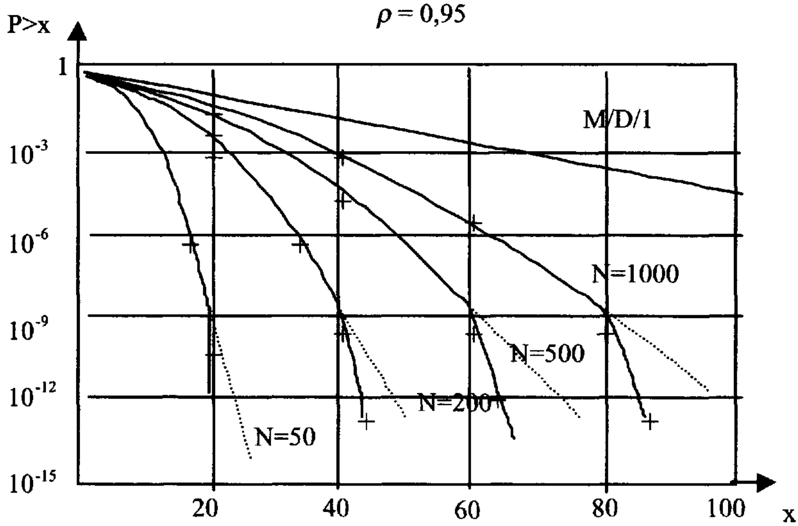


Figure 8.4. *Simulation of the N/D/D/1 queue*

The validity of the extrapolation is advantageously confirmed through another curve, such as in Figure 8.5.

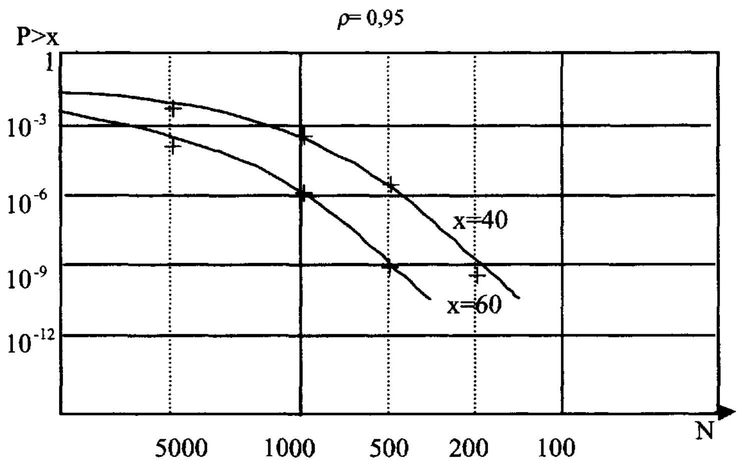


Figure 8.5. *Another view of the simulation results*

Anyway, one should preferably consider lengthening the simulation duration, or improving the simulation process (e.g. using such methods as Restart [VIL 91]), before trying any extrapolation.

Moreover, these examples show clearly the usefulness of the graphical representation, which greatly helps in clarifying the validity of the simulation process and in increasing the confidence towards numerical results.

8.4. Random numbers

Although random number generators are a facility offered by most computers, it is worth understanding their basic principles, as their usage goes far beyond the needs of simulation experiments. Here again in fact the specialist, in addition of his own knowledge of simulation languages, has to be able to make use, or even to design random number generators for specific needs (real traffic simulators, failure generation for reliability analysis, traffic measurements, etc.).

Generally speaking, the problem is to generate a sequence of numbers obeying as accurately as possible a given probability distribution, the latter being given by measurements, by the assumption on the model, etc.

8.4.1. *Generation according to a distribution*

The principle is as follows: one wants to generate a series of numbers X, (say, service durations) obeying a known distribution probability:

$F(x)=P[X \le x]$.

The method consists in drawing a random number u uniformly distributed between 0 and 1, and to estimate $X=F^{-1}(u)$ (as the probability is between 0 and 1). Figure 8.6 illustrates the principle.

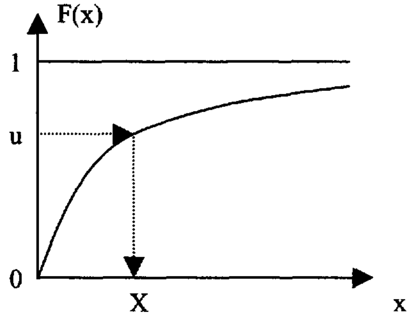


Figure 8.6. *Drawing a random number distributed according to a given probability law*

One can verify that the number is such that $P[X \le x]=F(x)$. In fact, $P\left[F^{-1}(u) \le x\right]=P[X \le x]=P[u \le F(x)]$, and $P[u \le F(x)]=F(x)$, as u is uniformly distributed in the interval (0,1).

The important example of the exponential distribution serves to illustrate the method. The distribution is $F(t)=1-e^{-\mu t}$.

The drawing operation yields a sequence of numbers $u=1-e^{-\mu t}$, from which the values of the variable t are given:

$$t=-\frac{\ln(1-u)}{\mu}. \tag{8-5}$$

It is immediately clear that the method can be generalized to the generation of complex laws, by combining several drawings. Of special importance is the case of distributions combining several exponential laws, such as the Erlang-k or hyperexponential distributions, see Chapter 4. For instance, the Erlang-k distribution is simply obtained by adding k successively drawn exponentially distributed variables.

Moreover, the method can be extended to the case of empirical distributions. This is of particular importance for performance tests where it may be necessary to reproduce a specific behaviour (experimental processing time distribution, complex packet arrival process observed at the output of a multiplexer, etc.). Assume the empirical distribution is given (actually a cumulated histogram). The target value X_k is obtained by looking for the interval (X_k, X_{k+1}) containing the random variable u:

i.e. $F(X_k) < u < F(X_{k+1})$.

Other methods have been developed (especially for the case where no closed form is available for F^{-1}. See e.g. [FIS 96].

All these developments assume the availability of a uniformly distributed random variable. Let us now indicate how to solve this problem.

8.4.2. *Generating pseudo-random variables*

We introduce here the concept of *pseudo randomness*. Actually, the methods in common use for generating "random" numbers rely on algorithms, rather sophisticated but perfectly deterministic and periodic. The method will be such that the series *looks like* a random sequence, presenting an autocorrelation as low as possible and a period of maximum length.

Numerous methods have been successively experimented with to generate such sequences. The first ones used quite empirical approaches (e.g. the *mid square method* introduced by Von Neumann), but they quickly have been discarded, in favour of more formal algorithms generating series that were open to theoretical analyses (based upon the results of algebraic number theory) guaranteeing optimal properties, with respect to periodicity and correlation.

One can mention the congruential method (multiplicative or additive), the use of shift registers, the use of specific series such as the successive decimals of π, and lastly the shuffling of several generators. The presentation here is restricted to congruential methods and shuffling, most often used in our applications.

Congruential methods

The method is based upon the following property: if I is an irrational number, then the sequence $u_n = nI, \bmod 1$ (modulo1), is uniformly distributed (theorem of Weyl Jacobi). The congruential method consists in taking

$u_n = ku_{n-1}, \mod m$, that is $u_n = k^n u_0, \mod m$. The value u_0 is the *seed*. The period of the generator is at best $m - 1$. The recommended values for m are $m = 2^p - 1$ with $p =$ 2,3,5,7,13,17,19, 31, 61, etc. (Fermat numbers). If $m = 2^q$, $q \geq 4$, which simplifies calculating the modulo, the maximum period is *at best* equal to $m/4$. Table 8.1 provides examples of parameters for good generators (see also [FIS 78, FIS 96]).

Table 8.1. *Parameters for random generators using the "multiplicative congruence" method*

k	m	period
16807 (= 7^5)	$2^{31}-1$	$2^{31}-2$
62089911		
4177924 (*)		
1226874159 (*)		
1099087573	2^{32}	2^{30}
2824527309		
69069		
1099087573		
2824527309		
3125	$2^{35} - 31$	$2^{35} - 32$
133955583 ($2^{27} - 2^{18} - 1$)		
68909602460261	2^{48}	2^{46}

(*) These values presented as "best choices" in [FIS 96]

The additive congruence makes use of the Fibonnacci series $u_n = u_{n-1} + u_{n-2}$, mod m. An example is given by $u_0 = u_1 = 1, \mod 2^p$, p being the machine data width in bits (e.g. 32). The following algorithm shows the practical use of the method:

Choose the seed u_0 (between 0 and 30), m (arbitrary, large), and u_1 (number < module m);

Take $\alpha = u_0$ and $\beta = u_1$;

1) Take $k = \alpha + \beta$;

Test if $k > m$; If yes: take $k = k - m$;

Take $\alpha = \beta$ and $\beta = k$;

Take $u = \dfrac{k}{m+1}$*; "u is the random number drawn, between 0 and 1"*

Repeat 1).

This method, fast and simple to implement usually provides generators worse than multiplicative congruence. However, combining generators of poor quality may overcome this weakness.

Shuffling generators

Shuffling two generators of low period is a way to generate another of improved quality. This procedure is appealing as it allows use of basic and fast generators, easy to be implemented. This is of special importance in field trials, where an artificial load is generated, where failures are simulated, etc. The principle is to use a first generator to fill a *shuffling table* which is accessed randomly using the second generator (the successive outcomes of this later are used as entry addresses in the table). Each number read in the table is then replaced by a new drawing of the first generator. A table with around 100 entries happens to give satisfactory results.

A single generator can even be shuffled with itself: the algorithm is as follows.

1) Initialize the table with the 100 first drawings of the generator, $t_i = u_i, i = 1...100$*;*

2) Draw the following random number u_k *and normalize it so as obtaining a value between 1 and 100. This yields i*

Entry i of the table gives the random number t_i*;*

Draw the following random number u_{k+1} *and update* $t_i = u_{k+1}$*;*

Repeat 1).

Another method, combining generators, results in sequences with increased periods [LEC 97]. Presented here is the case of a combination of two generators. Two generators are built, with $k_1 = 40014$, $m_1 = 2^{31} - 85$, and $k_2 = 40692$, $m_2 = 2^{31} - 249$:

– The step j-results in $X_{1,j}$ and $X_{2,j}$;

– Let $Y_j = X_{1,j} - X_{2,j} \bmod 2^{31} - 86$;

– The result is $r_j = Y_j / 2147483563$ if $Y_j > 0$, and $1 - 1/2147483563$ if not.

This algorithm generates a sequence with period $(m_1 - 1)(m_2 - 1)$, around 2×10^{18}.

This page intentionally left blank

9

Models

In this chapter our aim is to define a set of models used to evaluate the performances of systems and networks, based on the theoretical results and tools presented in the previous chapters. The essential objective is to provide as accurate a reflection as possible of the highly varied problems associated with modelling in an industrial environment (constructors, operators, laboratories, etc.).

The choice of these models is of course arbitrary and no attempt has been made to cover the whole of the subject. It is however based on two main criteria of essential importance:

– utility in our area of interest, that is the field of telecommunications;

– didactic interest from the viewpoint of modelling and the practical application of the theory. The level of demonstration will vary from one model to another, and it is the whole set of exercises which should provide a global view of modelling and resolution techniques. We will thus move from the simplest application of Markov graphs to the application of the Pollaczek method, while also covering simple rules of three and M/M/1 queues.

We will concentrate here solely on the field of traffic, as most of the important reliability models have already been covered in Chapter 6.

For the sake of clarity, we will draw a distinction between two major categories of models: those related to the control of systems and those to the field of information transport. Here also the distinction is arbitrary. It is primarily based on the most frequent usage in our field. It will be noted that some equipment, such as the Ethernet link, may be considered to belong to both fields: interconnection of control processors or user level data transport network, etc. The reader will easily overcome these distinctions wherever necessary.

9.1. Models for system control

9.1.1. *The simple feedback system*

This is a system in which each request, at its end of processing, has a probability p of being presented again in the system. This model for example corresponds to a transmission system with a failure transmission probability per message p, which results in its retransmission. It may also correspond to a processing system having to perform macrotasks which will only be capable of being run in several basic tasks. This may for example be the case of running processing whose code is not totally present in local memory: the processing must be interrupted with a probability p to go and retrieve the rest in the central memory (problem of virtual memory and page miss, for example). Figure 9.1 represents the system.

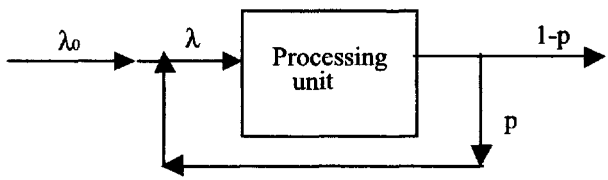


Figure 9.1. *The simple feedback system*

We have:

$$\lambda = \lambda_0 + p\lambda \text{, that is } \lambda = \frac{\lambda_0}{1-p}$$

In the hypothesis of Poissonian arrivals with rate λ, and with a service time per task obeying an exponential distribution with mean $1/\mu$, the results of the M/M/1 queue can be applied and we can thus write:

– mean time spent in the system

$$\overline{t_S} = \frac{1/\mu}{1-\rho} = \frac{1-p}{(1-p)\mu - \lambda_0}, \ (\rho = \lambda/\mu) \tag{9-1}$$

– mean number of tasks in the system:

$$\bar{n} = \frac{\rho}{1-\rho} = \frac{\lambda_0}{(1-p)\mu - \lambda_0} \tag{9-2}$$

We will find this type of model again in the case of a call repetition model (the two should not however be confused, as the variables and the use of the parameters are quite different).

9.1.2. *Central server*

The aim is to model a system such as a server (an IN server, a central processing unit, or a repair centre, hence its name *service centre*, *central server* or *repairman model*), handling requests from a finite number of identical clients (terminals, installed base equipments, etc.).

Each client submits a new request every $1/\lambda$ time units on average, an interval called *thinking time* or *operating time*. The processor of the central server processes a request in a mean time of $1/\mu$, referred to as the *service time*.

This model is very popular as it corresponds to a very large number of network cases. For example, it corresponds particularly well to the case of a server common to several terminals in a local network, each terminal waiting for a response after a request, or to that of a maintenance centre common to a limited set of equipments (terminals, computers etc) waiting for repair after a failure. Under similar conditions it may also correspond to the case of an IN (*intelligent network*) server that is common to several exchanges on the telephone network or on the NGN (*next generation network*).

Figure 9.2 shows the system and its N clients.

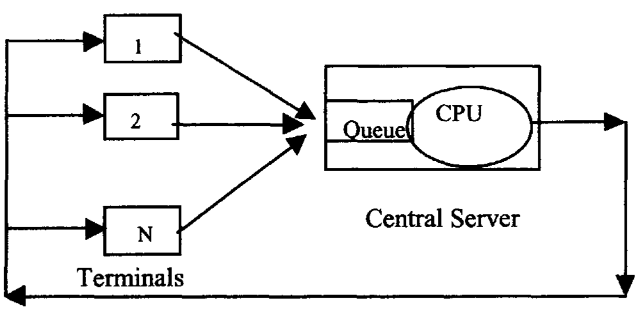


Figure 9.2. *The central server model*

If we assume an exponential law for the thinking time and processing time, with respective rates λ and μ, the transition graph representing the number of requests in the CPU is as indicated in Figure 9.3.

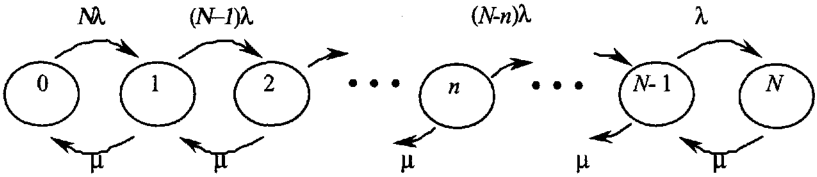


Figure 9.3. *Markov chain of the central server*

This is a special case of the Engset model (see Chapter 7) with a single server, termed M/M/1/N/N. This gives us the following quite simply:

$$P_n = \frac{\dfrac{N!}{(N-n)!}\alpha^n}{\sum\limits_{0\le k\le N} \dfrac{N!}{(N-k)!}\alpha^k}, \qquad \alpha = \frac{\lambda}{\mu}$$

and:

$$P_0 = \frac{1}{\sum\limits_{0\le k\le N} \dfrac{N!}{(N-k)!}a^k} = \frac{\dfrac{a^N}{N!}}{\sum\limits_{0\le n\le N} \dfrac{a^n}{n!}}, \text{ where } a = \frac{1}{\alpha} = \frac{\mu}{\lambda} \tag{9-3}$$

where readers will recognise as the Erlang loss formula, $E(N,a)$. This will give readers food for thought (what is the source, which are the servers?).

The utilisation factor of the central processing unit (CPU) of the server is thus:

$$\rho = 1 - P_0 = 1 - E(N,a) \tag{9-4}$$

Let us denote as c the mean cycle time for processing the requests from a terminal. This is the sum of its mean time between requests (thinking time), plus the mean wait w, plus the mean service time. Thus:

$$c=\frac{1}{\lambda}+w+\frac{1}{\mu}, \text{ or } w=c-\frac{1}{\lambda}-\frac{1}{\mu} \tag{9-5}$$

The mean cycle for the N terminal is N times smaller. This is the rate of arrival of requests at the CPU of the server. Hence:

$$\rho=\frac{N}{c}\frac{1}{\mu}$$

and so:

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

and finally:

$$w=\frac{N-1+E(N,a)}{1-E(N,a)}\frac{1}{\mu}-\frac{1}{\lambda} \tag{9-6}$$

The mean number of requests $\overline{L}$ in the central server queue is given by the Little formula:

$$\overline{L}=\frac{w}{c}$$

We also obtain the mean sojourn time $\overline{T_S}$, and the mean number of requests $\overline{L_S}$, in the central server:

$$\overline{T_S}=w+\frac{1}{\mu} \text{ and } \overline{L_S}=\overline{L}+\frac{\rho}{N}$$

It will be noted finally that these results apply to the case of a *server* functioning in *time sharing* mode (see below) in the case of a *finite number of sources* of processing. It will in fact be seen that such a system, from the viewpoint of the mean sojourn time (also referred to as *response time*), behaves in the same way as a simple M/M/1 server. We will therefore have as previously:

$$\overline{T_S}=w+\frac{1}{\mu}=\frac{N}{\rho\mu}-\frac{1}{\lambda}=\frac{N}{\mu(1-E(N,a))}-\frac{1}{\lambda} \tag{9-7}$$

We will now deal with the case of a time sharing processor with an infinite number of sources.

9.1.3. *Time sharing processor*

The aim here is to model a so-called *time sharing* system, as in such a system the processor shares its time equally between all the tasks (clients) in progress. There may be several call set-up contexts being simultaneously processed for example. There is no queue as such. Ideally, it is assumed that the processor spends an infinitely small time quantum on each task. This means that the processing time of a task is proportionally longer the more tasks there are in progress simultaneously, but also that the same processing time is smaller the intrinsically shorter the task. A system of this type favours tasks of short duration.

The model has been dealt with in particular by Coffman and Kleinrock [COF 70, KLE 75]. The results relate to characteristics (waiting time, sojourn time) of the processing of a given task (of given duration). We are interested here in the determination of "mean" characteristics of the processing time of all the tasks. It is in fact these parameters we will need to evaluate quality of service, e.g. in call processing processors, as the processing of a call is composed of several tasks of different durations. Essential characteristics are of course the mean and the variance of the total processing time of a task (its *sojourn* time in the system, i.e. its intrinsic processing time plus its elongation or *wait*).

The preceding references give the mean time spent waiting by a given task (or processing) of intrinsic service time s (i.e. its service time if it was alone in the system):

$$\overline{w_s} = \frac{\rho}{1-\rho} s \tag{9-8}$$

It should be noted that this result holds for a general service law.

In the case of an exponential service time distribution, its variance is:

$$\sigma_{w_s}{}^2 = \frac{2\rho s}{\mu(1-\rho)^3} - \frac{2\rho}{\mu(1-\rho)^4}(1 - e^{-\mu(1-\rho)s}) \tag{9-9}$$

$1/\mu$ being the mean service time of the tasks, exponentially distributed.

From this can be deduced the mean time spent in the system, i.e. the sojourn (or response time) for this task:

$$\bar{t}_s = \frac{1}{1-\rho} s \tag{9-10}$$

and:

$\sigma_{t_s}^{\ 2} = \sigma_{w_s}^{\ 2}$, for a task of fixed duration s.

Let us now concentrate on the characteristics of an "average" task, in the case of tasks with an intrinsic processing time s exponentially distributed with a mean value of $1/\mu$.

This gives:

$$\overline{w} = \int_0^\infty w_s \mu e^{-\mu s} ds = \frac{1}{1-\rho} \frac{1}{\mu} \tag{9-11}$$

For the variance:

$$\sigma_t^{\ 2} = E(t_s^{\ 2}) - [E(t_s)]^2$$

where:

$$E(t_s^{\ 2}) = E\left[(w_s + s)^2\right] = E(w_s^{\ 2}) + 2sE(w_s) + E(s^2)$$

or:

$$E(w_s^{\ 2}) = \sigma_w^{\ 2} + [E(w_s)]^2 = \sigma_w^{\ 2} + \frac{\rho^2}{(1-\rho)^2} s^2$$

and:

$$[E(t_s)]^2 = \left[\int_0^\infty \frac{1}{(1-\rho)^2} s \ \mu e^{-\mu s} ds\right]^2$$

and therefore:

$$\sigma_t^{\ 2} = \sigma_w^{\ 2} + \int_0^\infty \frac{\rho^2}{(1-\rho)^2} s^2 \mu e^{-\mu s} ds + \int_0^\infty \frac{2\rho}{(1-\rho)} s^2 \mu e^{-\mu s} ds + \int_0^\infty s^2 \mu e^{-\mu s} ds$$
$$-\left[\int_0^\infty \frac{1}{(1-\rho)^2} s\ \mu e^{-\mu s} ds\right]^2$$

$$\sigma_t^{\ 2} = \sigma_w^{\ 2} + \frac{\rho^2}{(1-\rho)^2}\frac{2}{\mu^2} + \frac{2\rho}{(1-\rho)}\frac{2}{\mu^2} + \frac{2}{\mu^2} - \frac{1}{(1-\rho)^2}\frac{1}{\mu^2}$$

$$\sigma_t^{\ 2} = \sigma_w^{\ 2} + \frac{1}{(1-\rho)^2}\frac{1}{\mu^2} = \sigma_w^{\ 2} + (\bar{t}_w)^2 \text{, a remarkable result} \tag{9-12}$$

We also have:

$$\sigma_w^{\ 2} = \int_0^\infty \sigma_{w_s}^{\ 2} \mu e^{-\mu s} ds = \int_0^\infty \left(\frac{2\rho s}{\mu(1-\rho)^3} - \frac{2\rho}{\mu(1-\rho)^4}(1-e^{-\mu(1-\rho)s})\right)\mu e^{-\mu s} ds$$

from which we deduce:

$$\sigma_w^{\ 2} = \frac{2\rho}{\mu(1-\rho)^3} - \frac{2\rho}{\mu(1-\rho)^4} + \frac{2\rho}{(2\mu-\rho)(1-\rho)^4}$$

From which, taking $1/\mu$ as the service time unit, we arrive finally at:

$$\sigma_w^{\ 2} = \frac{2\rho}{(2-\rho)(1-\rho)^2} \tag{9-13}$$

and:

$$\sigma_t^{\ 2} = \frac{2\rho}{(2-\rho)(1-\rho)^2} + \frac{1}{(1-\rho)^2} = \frac{2+\rho}{(2-\rho)(1-\rho)^2} \tag{9-14}$$

9.1.4. *Polling and token ring*

Let us first consider the general case of a cyclic service or *polling*. This type of service is used by many systems, both as a communication protocol between stations and as a service mode for real-time management between processes, or between message reception and transmission peripherals. We attempt to determine the waiting and processing time for a packet (or message, or process). Assuming Poissonian arrivals, we see that this time will be composed of the polling waiting time (for the first message) plus the waiting time in front of an M/G/1 server. Let us evaluate the mean cycle time for all the stations. This time is composed at each polled entity of an intrinsic polling time at each station τ_i (checking if it is free or busy, or the time to go from station to station), plus a service time s_i if there is a service to be performed.

Thus for N entities:

$$t_c = \sum_{i=1}^{N} \tau_i + \sum_{i=1}^{N} s_i \tag{9-15}$$

and so the mean is:

$$\overline{t_c} = \sum_{i=1}^{N} \overline{\tau_i} + \sum_{i=1}^{N} \overline{s_i} = c_0 + \sum_{i=1}^{N} \overline{s_i} \tag{9-16}$$

Now, for each entity i, on the assumption of arrivals with a rate of λ_i and tasks with times t_i, the mean number of tasks to be processed by cycle is $\lambda_i \overline{t_c}$ and their mean processing time is $\overline{s_i} = \lambda_i \overline{t_c t_i} = \rho_i \overline{t_c}$. We also have of course $\rho = \sum_{i=1}^{N} \rho_i$. From this the basic equation can be deduced:

$$\overline{t_c} = \frac{c_0}{1-\rho} \tag{9-17}$$

c_0 denoting what is generally called the "empty cycle".

In most cases in practice, $\overline{t_c}$ may be approximated to an almost constant value for, as soon as the number of entities polled is great, the variance of the scanning time becomes low, as in the case of service laws of the Gamma-n type with n large (for more complex models see [KUE 79]).

Let us now consider in more detail the case of a *token ring* type LAN based on the polling principle.

A token ring is a communication medium connecting in series a set of stations. The information is transmitted sequentially from one station to another in the ring. Each station retransmits the information to the next station bit by bit. The destination station retrieves the message intended for it by copying the information in transit. When the information returns to the transmitting station, this station erases the message from the ring. A station has the right to transmit when it detects the token on the ring. The token is a specific sequence of bits circulating in the medium and following each information transfer. The station that appropriates the token changes it into a start of frame signal SFS (*start of frame sequence*). When the transmission of its message has been correctly performed (without error, within a maximum delay etc.), the transmitter station reinitialises a new token which gives the following stations the possibility of transmitting in their turn.

The basic architecture is described in Figure 9.4.

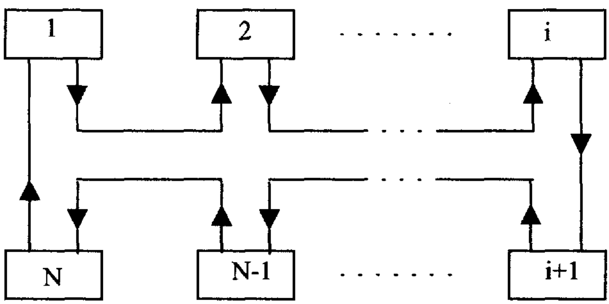


Figure 9.4. *Architecture of the token ring LAN*

A distinction is drawn between two major modes of token ring functioning: single token mode and early token mode.

In *single token mode*, a station that has transmitted does not transmit the "free" token to the next station until it has itself received in return the "busy" token that it had previously transmitted.

In *early token mode*, a station transmits the free token following the packet it transmits, thus enabling the next station to issue a message as soon as the last bit of the packet has left the preceding transmitter station. Thus, if the frames are short, several tokens may exist on the ring.

Single token mode

Let us first consider the single token mode, with an exhaustive service (all the messages waiting at each station are served).

The waiting time for a message or packet before transmission in the ring is the sum of the mean waiting time before being polled, i.e. approximately $\overline{t_c}/2$, plus the waiting time in front of an M/G/1 server. A message that finds no other message before it in a station will wait at least for the time to be polled. This instant will then be in a sense the start of the M/G/1 type service. If all the N stations are identical, $\lambda_i = \lambda/N$, $\rho_i = \rho/N$. This gives:

$$W = \frac{\overline{t_c}}{2}(1-\rho/N) + \frac{\lambda \overline{t^2}}{2(1-\rho)} \tag{9-18}$$

where:

$$\overline{t_c} = \frac{c_0}{1-\rho}$$

c_0, the empty cycle, is here the total propagation time on the ring. This is the propagation time τ, plus N times the latency time per station (from 1 to 16 bits).

t is the service time of a packet on the ring. It is the transmission time of the message, which depends on its length m (*payload* plus *header*, plus in some cases latency and inter-frame gap). If the frames are shorter than the propagation time, t is equal to c_0; if not, it is m.

Finally the global transfer time of a message is thus its waiting time W plus the transmission time t in the ring, plus the propagation time to reach a station, on average $c_0/2$:

$$t_f = \frac{c_0}{2(1-\rho)}(1-\rho/N) + \frac{\lambda \overline{t^2}}{2(1-\rho)} + \bar{t} + \frac{c_0}{2} \tag{9-19}$$

In the case of exponential frame lengths of a mean length $\overline{m}$, we obtain:

$$\bar{t} = c_0 \int_{x=0}^{c_0} \frac{1}{\overline{m}} e^{-x/\overline{m}} dx + \int_{x=c_0}^{\infty} x \frac{1}{\overline{m}} e^{-x/\overline{m}} dx$$

$$\bar{t} = c_0 + \overline{m} e^{-c_0/\overline{m}}$$

Similarly, we obtain:

$$\overline{t^2} = c_0^{\,2} + 2\overline{m}(\tau + \overline{m})e^{-c_0/\overline{m}} \tag{9-20}$$

In the case of constant frame lengths, we obtain:

$$\bar{t} = m\,,\ \overline{t^2} = m^2\,,\ \text{if } m >> c_0$$

$$\bar{t} = c_0\,,\ \overline{t^2} = c_0^{\,2}\,\text{if } m << c_0$$

We thus deduce the maximum admissible load of the LAN, or its efficiency E:

$$E = \frac{\overline{m}}{\bar{t}}$$

And for example in the case of exponential message lengths:

$$E = \frac{1}{(c_0/\overline{m} + e^{-c_0/\overline{m}})} \tag{9-21}$$

This relation clearly shows the impact of the empty cycle time (propagation and latency) compared with the length of the message to be transmitted. Efficiency will be lower, the closer the empty cycle value is to message length (or of course if it is greater than message length).

For example, in the case of a ring with a length of 1 km and a propagation time of 5 ns/m, with a bit rate of 5 Mbit/s, connecting 100 stations, and transporting messages of 1000 bits (200 µs), the impact of the latency is as follows:

– latency per station = 2 bits, total latency + propagation = 45 µs, c_0/m = 0.225:

$E = \rho_{\max}$=0.98

– latency per station = 8 bits, total latency + propagation = 165 µs, c_0/m = 0.825:

$E = \rho_{\max}$ = 0.79

– latency per station = 16 bits, total latency + propagation = 325 µs, c_0/m = 1.625

$E = \rho_{\max} = 0.55$

Early token mode

Let us now consider the early token mode. The calculations are the same as in the previous mode, but with the fundamental difference that the service time of a message is this time simply the transmission time m of the message. This gives:

$$W = \frac{\overline{t_c}}{2}(1-\rho/N) + \frac{\lambda\overline{t^2}}{2(1-\rho)} \tag{9-22}$$

and:

$$t_f = \frac{c_0}{2(1-\rho)}(1-\rho/N) + \frac{\lambda\overline{t^2}}{2(1-\rho)} + \bar{t} + \frac{c_0}{2} \tag{9-23}$$

where:

– in the case of exponential frame lengths with a mean length of $\overline{m}$:

$\bar{t} = \overline{m}$, $\overline{t^2} = 2\overline{m}^2$

– in the case of constant frame lengths:

$\bar{t} = m$, $\overline{t^2} = m^2$

In this mode, the efficiency of the LAN is no longer limited. In fact, more than efficiency, it is the response time that will be significant. Let us study the example of a communication LAN between processors of a distributed system, functioning in the early token mode.

In this case the length is limited: about 100 m. The bit rate of the LAN is 16 Mbit/s. We assume that the messages have a length exponentially distributed with a mean of 236 bytes (200 bytes of payload plus 32 bytes of header plus 4 bytes of latency buffer). That means, with 1 byte of interframe gap, a mean transmission time of 118.5 µs at 16 Mbit/s. The propagation rate is 5 ns/m, or 0.5 µs for 100 m. We assume a number of stations of 30, and latency per station is 2.5 bits, that is 0.156 µs.

We thus have:

$c_0 = 0.5 + 30 \times 0.156 = 5.2\ \mu s$

The share of the cycle time in the transfer time, at a load of 0.8 is therefore:

$t_1 = 5.2\ (1-0.8/30)/2(1-0.8) = 12.65\ \mu s$

The share of the wait in front of the M/M/1 server is:

$t_2 = 118.5 \times 0.8/\ (1-0.8) = 474\ \mu s$

The total transfer time is therefore:

$t_f = 12.65 + 474 + 118.5 + 2.6 = 607.7\ \mu s$

We note the preponderant influence of the M/M/1 service (474 + 118.5 = 592.5 μs). We will therefore retain in practice that the LAN in this application case may be modelled simply by an M/G/1 type server.

9.1.5. *Ethernet link*

Ethernet type LAN based on CSMA CD communication type protocols are characteristic support elements for communication between machines (processors or "stations" of a telecommunication system, servers, nodes of a network). Modelling of their performances is therefore essential, and above all allows the determination of the most significant performance parameters. As we shall see, the performances of this type of communication medium may differ greatly depending on applications.

These protocols have been developed as a result of problems relating to access to a radio channel by independent stations. As each one is unaware if the other one is starting to transmit, a collision may occur. Similarly, with an Ethernet LAN, independent stations are connected in star mode to an Ethernet hub or switch, and several may decide to transmit at the same time, if the link seems to them to be free, and a collision then occurs.

To resolve these conflicts, the protocols CSMA CD (CSMA, *carrier sense multiple access*, CD, *collision detection*) have been defined. There are three different types: persistent CSMA CD, non-persistent CSMA CD and p-persistent CSMA CD.

Persistent CSMA CD type. A ready-to-transmit terminal tests to see if the LAN is free, i.e. if there is no transmission in progress. If the LAN is free, it transmits the packet with a probability of 1. If the LAN is busy, it waits until the LAN is freed, and then transmits the packet with a probability of 1 (hence the name *persistence 1*).

Non-persistent CSMA CD type. In the non-persistent case, the terminal attempts transmission after a random time determined in accordance with a certain distribution function (e.g. exponential).

In both cases, the time can be cut up (*slotted*) into basic periods with duration T_s (linked to the propagation time). All the terminals are synchronised and test the channel at the start of the period.

In the persistent case, we can thus see that when two terminals or more are ready to transmit for the same period T_s, they will all transmit and will enter into conflict with a probability of 1. This has given rise to the idea of transmitting after a random time to minimise conflict problems.

P-persistent CSMA CD type. This is an evolution of the non-persistent case. After a collision, the transmitter retransmits immediately with a probability of p, and if not (with probability $1-p$) waits for a time T_s, and then restarts the same procedure.

Efficiency

Let us study the behaviour and more particularly the efficiency of these protocols, based on the theoretical configuration of Figure 9.5.

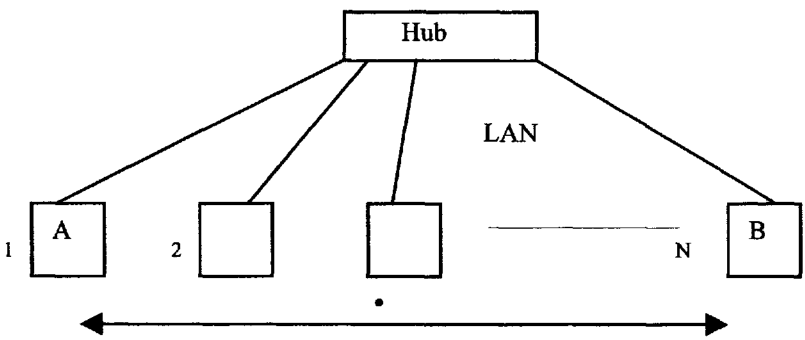


Figure 9.5. *Ethernet configuration*

Let m be the holding time of the bus by a message; N the number of stations connected to the bus; τ the link test time (equal to the propagation time between stations A and B, which are physically the most distant from each other); T_s: the time is divided into intervals with a duration of T_s, equal to twice the propagation time τ. The physical specification of Rec. IEEE 802.2 gives $T_s = 512$ "bit times" at 10 Mbit/s for 5 km, which corresponds to 51.2 µs (this is the worst case; in practice it will be faster as the LAN will be shorter).

We first calculate the virtual transmission time of a message, i.e. the time necessary to successfully transmit a message between two stations A and B. It first takes A m time units to transmit the message. Then B must wait for time τ to detect the end of transmission in view of the propagation time. The minimum transmission time is therefore $m + \tau$.

Let us now consider the case of a collision: A has begun to transmit a message and B also decides to transmit just before the message from A reaches it. B has tested message absence for τ, and it will take another τ before A detects the collision. There is therefore a time T_s equal to 2τ to detect a collision. If we suppose that it takes k periods of T_s to resolve the conflict, we then have an expression of the virtual transmission time:

$$t = m + \tau + k2\tau \tag{9-24}$$

Let us calculate k, where k depends of course on the retransmission strategy. For ethernet with persistent CSMA CD the following simplified model accurately reflects the reality:

– let π be the probability of success at each interval $T_s = 2\tau$. So:

$$\bar{k} = \sum_{1}^{\infty} x\pi(1-\pi)^{x-1} = \frac{1}{\pi} \tag{9-25}$$

– let p be the probability for a station of trying the transmission for T_s, and let N be the number of stations. The success probability for a station is then:

$$\pi = Np(1-p)^{N-1}$$

Bearing in mind that this probability is optimum for $p = 1/N$, we choose:

$$\pi = (1-\frac{1}{N})^{N-1}, \text{ that is } \pi = e^{-1}, N \to \infty \tag{9-26}$$

And finally the mean value of the virtual transmission time is:

$$\bar{t} = \bar{m} + \tau + 2e\tau \tag{9-27}$$

We then define the efficiency of the protocol by the relation:

$$E = \frac{\bar{m}}{\bar{m} + \tau + 2e\tau} \tag{9-28}$$

which is the maximum utilisation factor of the LAN.

One generally notes $a = \frac{\tau}{\bar{m}}$ and thus $E = \frac{1}{1 + (1 + 2e)a}$.

And from this it is deduced that to obtain efficiency greater than 0.6, it is essential that $a < 0.1$.

The interest of using such a LAN is therefore generally conditioned by this relation, which calls for the adjustment of the length of messages to the propagation and collision detection time.

Example of calculation of collision detection time T_s

We give an example of simplified calculation of this time in an actual application case, for its adjustment is fundamental for the efficiency of the protocol, as we have just seen. Above all we want to show the important impact of the time taken to pass through the physical layers of the hub and of stations.

We should first state that the time period during which the collision can normally take place corresponds to the transmission time of a preamble of 7 bytes (typical value), plus the transmission time of the SFD byte (*start frame delimiter*), plus the transmission time of the first 64 bytes of the ethernet frame.

Let t_0 be the instant at which the transmission ends without collision of a message by a station i. At this instant, the station i starts its *Interframe Gap Time* of duration D_g. The time t_0 is taken as the origin.

Let t_1 be the instant at which the last bit of the message from i is seen by all the other stations (at physical level). This time is equal to the propagation time D_p between station i and the *hub*, plus the time of passing through the *hub* D_h, plus the propagation time between the hub and the other stations, D_p, plus a time for passing through the physical layer in stations D_s.

$t_1 = D_p + D_h + D_p + D_s$

Let t_2 be the time at which station i transmits its first byte of a preamble of a message with the destination j:

$t_2 = t_0 + D_g$

Let t_3 be the time at which station j transmits its first preamble byte. It does it at t_1 plus, in the worst case, D_g:

$t_3 = t_1 + D_g$

Let t_4 be the time at which the first preamble byte of i arrives for transmission on the j port of the hub. It is t_2 plus the time for passing through the physical layer of station i, plus the propagation time, plus the time for passing through the hub:

$t_4 = t_2 + D_s + D_p + D_h$

Let t_5 be the time at which the first preamble byte of j arrives at the hub. It is t_3 plus a time for passing through the physical layer of the station, plus the propagation time D_p from j to the hub:

$t_5 = t_3 + D_s + D_p$

Let t_6 be the time at which the message presence signal for reception is seen by j. It is t_4 plus the propagation delay from the hub to j, plus the station detection time (which is assimilated to the time for passing through the physical layer):

$t_6 = t_4 + D_p + D_s$.

We therefore have:

$t_5 = D_p + D_h + D_p + D_s + D_g + D_s + D_p$

$t_6 = D_g + D_s + D_p + D_h + D_p + D_s$

We see therefore that the physical layer of station j detects at t_6 the arrival of a message when it is preparing to transmit. At t_5 the hub detects the collision on its port j. And finally station i will receive the collision signal at T_s. The station will then switch to idle and the access attempt process will restart for all stations:

$T_s = t_5 + D_h + D_p + D_s$

$T_s = D_p + D_h + D_p + D_s + D_g + D_s + D_p + D_h + D_p + D_s$

A good approximation of the time sought is therefore:

$T_s = 4\ D_p + 2\ D_h + 3\ D_s + D_g$

Orders of magnitude for the times are:

$D_p = 5.7\ ns$ per meter

$D_h = 46$ bit times (example for a 10 Mbit/s hub, a bit time equal to 0.1 μs)

$D_s = 24$ bit times

$D_g = 96$ bit times

For an ethernet at k × 10 Mbit/s and hub to station lengths of L metres:

$T_s = 4L \times 5{,}7ns + 260 \times (100/\text{k})\ ns$

We obtain typical values shown in Table 9.1.

Table 9.1. *Collision detection time*

LAN length	Bit rate	T_s value
$L = 1000$ m	10 Mbit/s	22.8 + 26 = 48.8 μs
	100 Mbit/s	22.8 + 2.6 = 25.4 μs
	1000 Mbit/s	22.8 + 0.26 = 23.06 μs
$L = 100$ m	10 Mbit/s	2.28 + 26 = 28.28 μs
	100 Mbit/s	2.28 + 2.6 = 4.88 μs
	1 000 Mbit/s	2.28 + 0.26 = 2.54 μs
$L = 10$ m	10 Mbit/s	0.228 + 26 = 26.22 μs
	100 Mbit/s	0.228 + 2.6 = 2.82 μs
	1 000 Mbit/s	0.228 + 0.26 = 0.48 μs

This clearly shows the important influence of the physical layer transfer delays and transmission times.

In the following, in our applications and in a dimensioning example in Chapter 10, we will adopt the typical values of 50 μs and 10 μs.

Let us consider the case of the inter-node signalling of a network of stations, or the case of VoIP packet exchanges, with an average message length of 200 bytes. To achieve an efficiency of 0.6, maximum distances between stations and hub of about 100 m at 10 Mbit/s and 10 m at 100 Mbit/s are required.

Let us now consider the case of data exchanges consisting of files in IP packets of 1500 bytes: the maximum distances required are 1000 m at 100 Mbit/s, and 100 m at 1000 Mbit/s.

This shows the fundamental impact of message length.

Transfer time

We are developing here an approximate expression for transfer time. We have established that the mean value for virtual transmission time is:

$$\bar{t} = \bar{m} + \tau + 2e\tau$$

We can now consider this time as the service time of the LAN, and thus evaluate the response time as that of an M/G/1 server. If λ is the rate of arrival of the messages to be transmitted on the LAN, then its utilisation factor is:

$$\rho = \lambda \bar{t} = \lambda(\bar{m} + \tau + 2e\tau)$$

The waiting time before transmission is:

$$w = \frac{\lambda(\overline{t^2})}{2\{1 - \lambda(\bar{m} + \tau + 2e\tau)\}}$$

and the transfer time is approximately:

$$t_f \approx \frac{\lambda(\overline{t^2})}{2\{1 - \lambda(\bar{m} + \tau + 2e\tau)\}} + \bar{m} + \tau + 2e\tau$$

Furthermore, in view of the above, it is clear that the system, in order to be efficient, must comply with certain strict constraints, and in particular that corresponding to $a = \frac{\tau}{\overline{m}} < 0.1$. In view of the orders of magnitude, we will therefore write:

$$\overline{t^2} \approx \overline{m^2} + (4e+2)\tau\overline{m} + (2e+1)^2\tau^2$$

and thus finally:

$$t_f \approx \frac{\lambda(\overline{m^2} + (4e+2)\tau\overline{m} + (2e+1)^2\tau^2)}{2\left\{1 - \lambda(\overline{m} + \tau + 2e\tau)\right\}} + \overline{m} + \frac{\tau}{2} + 2e\tau$$

This expression gives sufficiently precise results in actual application conditions. A more detailed expression is given in the literature [BUX 81].

And the limit value for very small values of a will be:

$$t_f \approx \frac{\lambda(\overline{m^2})}{2(1 - \lambda\overline{m})} + \overline{m}$$

Note also that in the case of frame lengths complying with an exponential distribution we have: $\overline{m^2} = 2\overline{m}^2$, and in the case of constant frame lengths, $\overline{m^2} = m^2$.

9.1.6. *Benchmarks, processor modelling*

The aim here is to evaluate the processing capacity of a new system, or a new version of the system by the use, for example, of more rapid processors or platforms.

In the case of a simple technological evolution, the use of a *benchmark*, i.e. a performance test software, is the most appropriate. As soon as possible, a significant extract of the code to be run will be tested on the new machine to measure the change in execution time. In telecommunications applications, it will be relatively easy to define the measurement of the most frequent communication primitives (*send*, *receive*) and the measurement of typical processing at application level. In many cases however, at this point in the life cycle, it is not yet really possible to execute significant code on the machines, and the evolutions are often complex (processor, but also memory, bus, etc.).

Information about the possible change in performances may then be obtained from benchmark results published in the literature, or on the Web, by users or constructors. In particular we find results known under the term *SPEC integer,* from the organisation *System Performance Evaluation Corporation*, an organisation of computer manufacturers which develops benchmarks and periodically publishes results. We then have Specs XYZ, where XYZ is the *Spec* type and the year. For example, we have *Spec CPU* 20xx, etc. which characterise the performances of computer centre type systems. But we also find *Spec Server*, or *Web*, which are better adapted to the performances of server or web type applications. The published results primarily constitute points of comparison between machines (CPU, memory, compiler, etc.), in very specific machine use conditions, as we shall see hereafter (type of software run, cache rate, bus speed, etc.). These results should therefore be handled with great care, but can provide some indications about the potential performances of products.

A distinction may be drawn between simply replacing a processor by a new and more rapid version of the processor, and using a new processor from another family.

In the first case, the gain can be estimated on the basis of *Spec integer* ratios.

In the second case, the *Spec integers* are rarely significant, and the performances must be estimated by a more detailed analysis of the behaviour of the processor, taking into account the characteristics of the software to be run. A typical behaviour model is given below.

The organisation of a processor board is shown in schematic form in Figure 9.6.

A processor board is characterised by its internal frequency, its cache capacity, the speed of its internal bus, and its private memory capacity. Its communication with other processors is characterised by its exchange memory capacity and its external bus capacity.

Its internal clock frequency is f_i: it expresses the value of the basic cycle $c_i = 1/f_i$ and thus the minimum value of the execution time of a processor instruction. We have the relation: 1 instruction = p cycles + memory access time, where p is generally equal to 1, or less (on an average per instruction), for existing processors (internal architecture with multiprocessors and *pipe*).

The memory access times are to be added as soon as the operation code or operand is not present in the primary cache. The processor also must run a certain number of cycles whose value depends on the access speed (local bus or external bus).

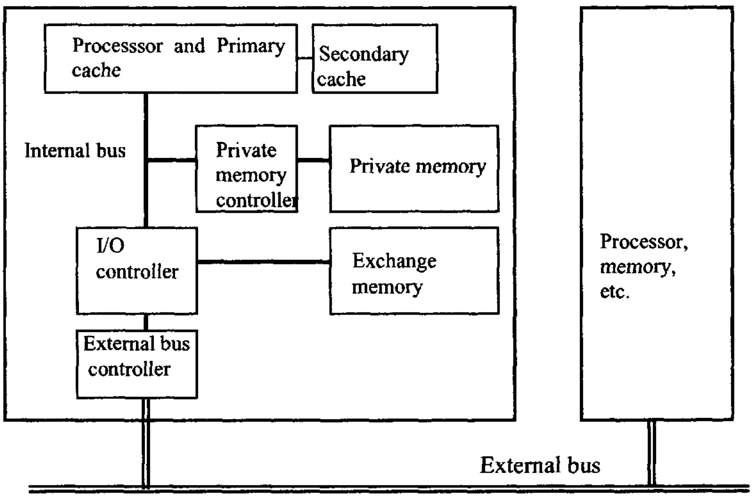


Figure 9.6. *General structure of a processor board*

We will consider the case of a processor equipped with a primary cache (direct execution in p cycles of duration c_p), a secondary cache (access in s_i cycles of duration c_s), and a private memory (memory access through the bus, access in b_i cycles of duration c_b).

The term *primary and secondary instruction hit rate* refers to the probabilities of finding the code to execute respectively in the primary and secondary caches: h_{ip}, *his/her.*

If the code corresponding to the *instruction* is not in the primary cache, it is necessary to go and retrieve it (line by line) from the secondary cache, or from the private memory, and refresh the primary cache.

Similarly for the data associated with the instructions: if the *data* is not in the primary cache, supplementary time is necessary to enable it to be found in the secondary cache (s_d cycles) or in the private memory (b_d cycles). This also leads to the definition of a primary and secondary hit rate: h_{dp}, h_{ds}. We call d_i

the reading rate of the data associated with the instructions (percentage of access to data by instruction).

Finally, as we are dealing with systems involving large numbers of exchanges between processors, we also have the times for access to external data via the exchange memories (e.g. where transmit and receive waiting queues may be located). We denote as d_e the exchange data reading rate, e the number of access cycles with the duration c_e and t_e the exchange memory reading time.

This gives the following relation for the mean execution time of an instruction:

$$I = h_{ip}.pc_p + (1-h_{ip})\left(h_{is}.s_i.c_s + (1-h_{is}).b_i.c_b\right) + d_i(1-h_{dp})$$
$$\left(h_{ds}.s_d.c_s + (1-h_{ds})b_d.c_b\right) + d_e.(e.c_e + t_e) \qquad (9\text{-}29a)$$

The evaluation of the values associated with these parameters is of two kinds.

On the one hand, information is available in the constructors' manuals enabling the quantification of the number of cycles necessary for the various access cases (cache or other accesses) and thus the determination of access times depending on bus speeds. For example, if the processor frequency is 266 MHz, we will deduce from this that c_p = 3.76 ns, and if the external bus frequency is 66 MHz, this will give c_b= 15.15 ns, and we will also find that p = 1 and b_i = 12 (see examples below).

Furthermore, it is necessary to determine from the code to be executed the cache rates and the access rates to private memories, exchange memories, etc. This can only be obtained by analysis of the various types of applications, by experience and analogy with previous measurements, or of course by new measurements if some of the code is available. It is worth noting the relative importance of the various parameters. For example, the access time to data in private memory, or to the exchange memory, may be counted in hundreds of ns (e.g. 150 ns) and be even larger if the information has to be retrieved from an external memory via the external bus (e.g. 500 ns). The instruction time will therefore be totally different, depending on whether the hit rate is high or not, and on whether the application has a large number of external exchanges or not.

This is why great caution must be shown in relation to the use of the *Spec integer*. The actual gain value may be extremely variable depending on the type of application in question.

Let us consider three very simple cases.

Example 1. In the first case we have only repetitive code, which is mainly located in the primary cache, or in private memory, and the same for the data. We assume also an extremely low external exchange rate per instruction (less than 10^{-3}). This case may arise when running calculation programmes. Furthermore, in the case of such linear programmes, any instruction that is not present in the cache means in practice retrieval in groups of 8 instructions, and thus the presence of the following 7 instructions in the cache. The h_{ip} rate is therefore at least 0.875.

It is assumed that $h_{ip} = 0.9$ and $c_p = 3.76$ ns (266 MHz processor).

The private memory is read in 12 cycles through the 66 MHz bus, in lines of 32 bytes, i.e. 8 instructions at a time. This therefore gives $b_i = 12$ and $c_{bi} = 1/8 \times 66$ MHz $= 1.9$ ns.

We assume a data access rate per instruction of $d_i = 0.6$. We also assume that $h_{dp} = 0.9$. If not, the data is in private memory. The reading of the data in private memory is carried out by line of 32 bytes by instruction, and is performed in 10 cycles. (Note that each time only the data for one instruction is retrieved). This gives therefore $b_d = 10$ and $c_{bd} = 15.1$ ns.

The instruction value is therefore in this case:

$$I = 0.9 \times 3.76 + 0.1 \times 12 \times 1.9 + 0.6 \times 0.1 \times 10 \times 15.1 = 14.7\ ns$$

Example 2. Let us now consider the example of software that again carries out very few external exchanges, but which runs the code with many context changes. This case may correspond to that of a central call processing processor.

As in the previous case, the instructions are either in the primary cache or in the private memory. The same applies to the data. But this time we assume that $h_{ip} = h_{dp} = 0.6$.

The value of the instruction in this case is therefore:

$$I = 0.6 \times 3.76 + 0.4 \times 12 \times 1.9 + 0.6 \times 0.4 \times 151 = 47.6\ ns$$

This represents a mean time over three times larger than in the previous case. Note the very substantial impact of the cache rate.

Example 3. Finally let us consider the case of software dedicated to external exchanges. This case corresponds to that of a pre-processing processor, specialising in exchanges, as part of a distributed system.

We assume that the machines are more specialised with limited software. We assume that $h_{ip} = h_{dp} = 0.8$, but this time we assume an exchange memory access rate *per instruction* of $d_e = 0.5\%$.

Access to the local exchange memory is via the 66 MHz bus in 10 cycles. About 10 accesses per message are required for file management (e.g. for updating pointers). This gives $e = 100$, $c_e = 15.1$ ns. The message transfer itself requires 5 cycles per block of 8 bytes. Thus the following is required for a message of 100 bytes: $t_e = 62.5 \times 15.1 = 944$ ns.

The value of the instruction in this case is therefore:

$I = 0.8 \times 3.76 + 0.2 \times 12 \times 1.9 + 0.6 \times 0.2 \times 151 + 0.005 \times (1\ 510 + 944) =$
$37.9\ ns$

That is a mean time which is intermediate between those of the two previous cases.

We could also have considered the case of a processor dedicated not only to exchanges but also to processing. In this case it would have been necessary to adopt slightly lower cache efficiency rates.

From the previous study, we could retain the following simplified model (in which in particular the secondary cache is not considered):

$$I = h_{ip}.t_{ic} + (1-h_{ip})(t_{im}) + d_i(1-h_{dp})(t_{dm}) + d_e.(a_e + t_e) \qquad (9.29b)$$

with the following numerical values for a 266 MHz processor:

$$I = h_{ip}.3{,}76ns + (1-h_{ip})(23ns) + d_i(1-h_{dp})(151ns) + d_e.(1510ns + 944ns) \qquad (9.29c)$$

We will provide an example of the application of this model for system capacity calculation in Chapter 10.

9.1.7. *Disc system*

The disc is an essential element in telecommunications systems, because of its enormous data storage capacity, which generally amounts to tens of gigabytes (9, 18, 36 or 72 Gbytes). It has many uses, in servers (on the web, in fixed and mobile networks), and in telecommunication network operating systems (gathering of observations, storage of charging). The evaluation of its service time is therefore an important consideration in evaluating the performances of the systems which use it (response time of a server, for example). This is particularly true in that mechanical elements intervene at this level which lead to execution times which are of a quite different order of magnitude from those of processors.

A subsystem with a disc is shown in schematic form in Figure 9.7.

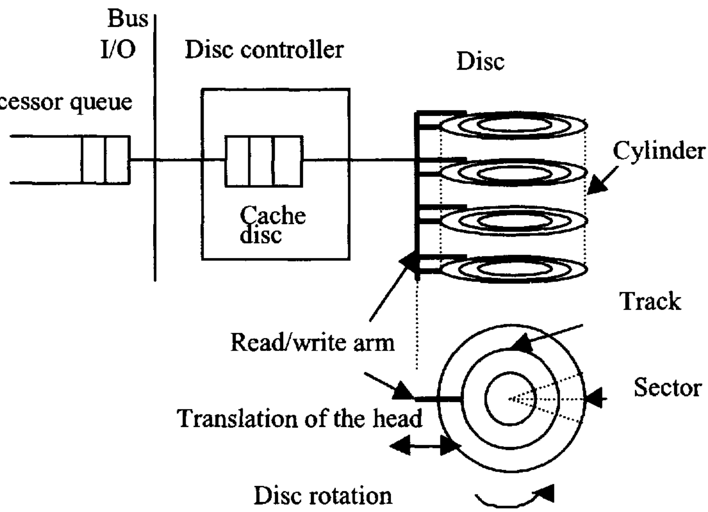


Figure 9.7. *Architecture of a subsystem with disc*

Let us describe the actions necessary for reading a disc (parallels for writing are easily established). A request is queued in the input/output system controller (there may be a cache effect previously to avoid an unnecessary input/output, but this forms part of the processor modelling, as in the case of memory accesses).

The disc controller provides the interface with the system (processor) via the input/output bus, and intelligently manages disk access by minimising reading and writing operations. For this purpose it has a buffer to carry out two separate major functions: *prefetch* (or *read look ahead*), and the *disc cache* function that we will describe a little later. 2 Mbytes is a typical buffer size. There are two main types of controllers: IDE (*integrated drive electronic interfaces*) and SCSI (*small computer system interface*). SCSI enables the management of a larger number of peripherals and is well adapted for multi-task applications on server type machines.

A disc is composed of several double-sided plates (e.g. 4 plates and 8 read/write heads). One face may contain 10, 15 or 20 gigabytes. On each plate, the information is organised in tracks (20,000 to 30,000 tracks, with for example 200 kbytes per track), and each track is divided into sectors of the same size (512 to 4096 bytes). All the tracks of the same rank on the plates form a cylinder. To read a block there will therefore be three operations: track selection, disc rotation and reading (transfer) of the information. There is therefore a horizontal displacement of the read arm to position itself on the right track/cylinder (*search* time t_S). It will be noted that the shorter the time, the closer the information is to the edge of the disc. This may be favoured by certain operating systems, and this is facilitated if the number of plates is greater for the same quantity of information. To simplify we will not consider this optimisation, just as in reality there is more information at the edges. The disc is then rotated to position itself on the good sector (*latency* time t_L). This time depends on the disc rotation speed and the distance of the block sought in relation to the arm position. This is followed by the transfer of the block from the disc to the disc controller (*transfer* time t_T). The transfer time depends of course on the quantity of information to be read, the rotation speed of the disc, but above all the possible density of information on the track.

Finally, the processing time at *controller* level is termed (t_C) (reception of the read request, or write request, and information transfer to the processor).

Modelling

Disc read service time

An information block read request ("READ") specifies the address of the first sector of the block to be read and its length expressed in number of sectors.

Reading of sectors

For mechanical reasons associated with disc rotation, there is a gap between the sectors of a track. When the buffers were small, reading could only take place sector by sector, and the buffer had to have transferred its information to the processor during the gap, before being able to receive the next sector. If not, it was necessary to await a complete rotation of the disc to read the next sector. This problem was particularly acute if the size of the cache was small. The *interleave* technique overcame this phenomenon by extending the gap by inserting physical sectors between two consecutive logic sectors (the logic addresses do not necessarily correspond to the physical addresses). As systems now have large caches (2 Mbytes), we can neglect this phenomenon. Furthermore, even when reading several consecutive tracks, it is also possible to use a similar method, the *offset* (track skew), to avoid the need for a complete additional rotation.

Prefetch

In prefetch mode, the buffer is filled by the logic sectors of the disk immediately behind the sector requested. Today, because of the size of the buffers, it is possible to retrieve a whole track. Thus, if the following requests relate to consecutive blocks, or blocks from the same track, the information is already in the buffer. This avoids a certain number of disk accesses (arm translation, rotation, etc.). If this happens, we speak about a *hit prefetch*, and if not it is a *miss*. This mode is particularly efficient when there is a *run* of requests relating to consecutive blocks. This is the case of a server-type environment in which large quantities of information are sought in sequential mode.

Cache

In cache mode, as for the memory cache, the controller keeps in its memory the sector blocks that have already been accessed and stored in the buffer. When a read request is made, if the address requested is already in the cache, the controller transmits all the subsequent blocks to the processor. If not, there is a disc access, transfer to the buffer, and then transfer to the processor. As in the case of memory, efficiency depends on the hit cache rate. Cache mode is particularly appropriate in the case of repetitive accesses to small segments of data, which are often the same. This mode is for example very well suited to desktop type environments.

Model

In view of the foregoing analysis, we evaluate the average service time of the disc server t_D, per request, by:

– *prefetch k sectors* mode:

$$t_D = t_C + (1 - h_{pf})\left(t_S + \frac{c_d}{2} + t_{Tk}\right) \tag{9-30}$$

In point of fact, if the blocks sought are already in the buffer, there is only a controller time t_C to take into account. If not, with a probability of $(1 - h_{pf})$, an arm movement is required with a mean value t_S, plus a half-rotation of the disc on average to reach the first sector with a value of $t_L = c_d/2$, and then the transfer of k consecutive sectors (t_{Tk}), and of course the processing time of the controller as above;

– *prefetch track* mode: in this case it is possible to retrieve everything (all the track in t_{Tt}) without even positioning the head on the first sector. The controller will put the information back into order:

$$t_D = t_C + (1 - h_{pf})(t_S + t_{Tt}) \tag{9-31}$$

– *cache* mode: this is the same type of formula, with h_c the cache hit rate:

$$t_D = t_C + (1 - h_c)\left(t_S + \frac{c_d}{2} + t_{Tk}\right) \tag{9-32}$$

A case study

Let us consider the case of request sequences with a length of n (there are n consecutive requests concerning the reading of a total of k consecutive sectors; this is also called a run of length n). The prefetch miss rate is then $1/n$. The service time of the disc per request is then:

$$t_D = t_C + \frac{1}{n}\left(t_S + \frac{c_d}{2} + t_{Tk}\right)$$

t_C is of the order of magnitude of a few hundred μs. We will assume a value of 200 μs; t_S is given by the constructor and has a mean value in the order of magnitude of a few ms. For a type 18 GB-SCSI disc for example, the value is 6 ms. c_d depends on the rotational speed of the disc, which is usually expressed in revolutions per minute. For a type 18 GB-SCSI disc, for example at 7200 rpm, this gives c_d = 8.3ms. t_{Tk} depends on the number k of sectors transferred and the transfer rate (given by the constructor). If we assume a prefetch of 60 sectors of 2048 bytes at a transfer rate of 20 Mbytes/s (which is a typical value). We thus obtain t_{Tk} = 6ms. Finally, for sequences of average length, n = 5, with each reading 12 sectors on average, we obtain:

$t_D = 0.2ms + 0.2(6ms + 4.15ms + 6ms) = 3.4\ ms$

Let us now assume that the whole track is being transferred:

$$t_D = t_C + (1 - h_{pf})(t_S + t_{Tt})$$

Let us take a track of 200 kbytes, retaining a transfer rate of 20 Mbytes/s (the reader will verify the coherence with the disc rotation speed of 7200 rpm). We thus obtain $t_{Tt} = 10$ ms. So:

$t_D = 0.2ms + 0.2(6ms + 10ms) = 3.4\ ms$

We thus obtain the same service time. In practice, it will be a matter of obtaining the best estimate of the characteristics of the requests (run value, number of sectors, etc.), to choose the best strategy depending on the characteristics of the disc. The question of the advantages of one solution as opposed to another will depend very heavily on all these characteristics (requests size and disc characteristics).

Service time in write mode

The use of the disc cache in write mode is very different. In fact it is only used for the temporary storage of the blocks to be written and to send back a write acknowledgement ("GOOD") to the processor, before any actual writing on the disc. The problem is that with this operating mode information is lost if the writing does not take place correctly, or if a failure occurs. In practice, the method of *grouping* will tend to be used at application level. The processor groups together several write requests into a single request, and only clears its transmission buffer once it has received an acknowledgment of actual writing on the disc. The performance gain is obvious (a single arm movement, a single rotation), and reliability is ensured by redundancies at application level. The processor on the other hand has a slightly more complex processing task. This method of grouping is particularly well adapted to real time data safeguard, for example for charging accounts or call details records, a key function for an operator.

If n is the average number of write requests that can be grouped into a single write ("WRITE") of a total of k sectors, the service time is:

$$t_D = t_C + \frac{1}{n}\left(t_S + \frac{c_d}{2} + t_{Tk}\right) \qquad (9\text{-}33)$$

Application example

Let us consider a *billing agent*. It receives and stores the charges and call details from telecommunications exchanges (before periodically transmitting them in files to a billing collector). The call detail reports (used for detailed billing for example) are made in real time on completion of each call. A call detail report corresponds to a message of about 250 bytes. It would clearly not be efficient to transmit a message each time a call is made, or to write on the disc each time there is a message. The exchanges thus group together call details in blocks of 8 (for example) to form messages of 2 kbytes. This packeting delay in transmission does not greatly disadvantage the real-time aspect of the safeguard, bearing in mind that traffic per exchange is in the order of hundreds of calls per second (anyway there is always a time out). The billing agent then puts them together into groups of 16 kbytes (8 blocks) to carry out the safeguards to disc and to send a global acknowledgment to the exchange (which keeps the groups in its memory until they are acknowledged). For a traffic of 2000 records per second (the traffic of a large exchange generating several details per call), the billing agent thus has to process 250 messages/second and 62.5 disc writes/second (two accesses to the disk are necessary, the second for pointers management with approximately the same time).

If we assume a processor processing time of 2 ms per message (including grouping and acknowledgment operations), the processor load is thus in the order of 50%.

If we assume that the disk has the same characteristics as before (7200 rpm, etc.), the write time from 16 kbytes at 20 Mbytes per second is 0.8 ms and the service time per group is:

$$t_D = 0.2ms + (6ms + 4.15ms + 0.8ms) = 11.15\ ms$$

(Note the preponderance of the access time in the sector, and that this time can be reduced further if sequential writes are possible.)

This represents a disc load of about 70%. This clearly shows how the disc might be the bottleneck and how essential it is to optimise writes.

Depending on the disk and information characteristics, an optimal strategy may be defined.

9.1.8. *Managing a pool by blocks*

We are attempting here to determine the size of a pool of resources organised in blocks.

We thus consider a pool of resources organised in blocks of N resources, to which several traffic flows are offered. How many blocks must be provided for a given probability of lack of resources, bearing in mind that there is no rearrangement of free resources between several blocks? This problem is a matter of the allocation of memory space by blocks (context management, data management for various types of transactions), or the allocation of ATM cells in the AAL2 technique for voice on ATM for example (VoATM), or finally the basic problem of circuit seizing on several trunk groups with overflow. And still without rearrangement, either because the service does not support interruption, or to simplify realisation.

This model, like the ethernet link model, covers both the control field and the transport field. Figure 9.8 illustrates the basic problem.

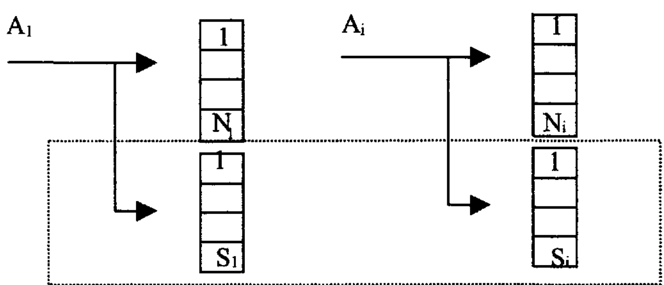


Figure 9.8. *Management of a pool of resources by blocks*

Traffics of intensity A_i are offered to blocks of N_i resources (servers). Another block of S_i resources must be taken if the first N_i resources are not sufficient. For this study only the single initially allocated block is considered. The problem is easily generalised to k_i initial blocks, to which are added a further block if necessary.

Let us first consider a single traffic current $A = \lambda/\mu$, where λ is the demand arrival rate and μ the service rate.

The calculation of the demand probability of another block is performed on the basis of state equations, in accordance with a Markov chain as shown below. We are following here Brockmeyer's demonstration [BRO 54]. We note respectively n the number of resources taken in the first block and s that of the second block. The behaviour of the first block of N places follows a birth and death process with a number of states limited to N. That of the overflow block is a pure death process as long as $n < N$. The second block only receives new connections once the first block has reached state N. Its occupation process then also becomes a birth and death process.

Let us call P_s^n the probability of the system state (n, s).

The following equations of the future can easily be written:

$$P_0^0(t+\Delta t) = P_1^0(t).\mu 1\Delta t + P_0^1(t)\mu 1\Delta t + P_0^0(t)(1-\lambda\Delta t)$$
$$P_s^n(t+\Delta t) = P_s^{n-1}(t).\lambda\Delta t + P_{s+1}^n(t)\mu\Delta t + P_s^{n+1}(t)\mu_{n+1}\Delta t + P_s^n(t)$$
$$\left[1-(\lambda+\mu n+\mu s)\Delta t\right]$$
$$P_s^N(t+\Delta t) = P_s^{N-1}(t).\lambda\Delta t + P_{s-1}^N(t)\lambda\Delta t + P_{s+1}^N(t)\mu_{s+1}\Delta t + P_s^N(t)$$
$$\left[1-(\lambda+\mu N+\mu s)\Delta t\right]$$
$$P_S^N(t+\Delta t) = P_{S-1}^N(t).\lambda\Delta t + P_S^{N-1}(t)\lambda\Delta t + P_S^N(t)\left[1-(\mu N+\mu S)\Delta t\right]$$

and at equilibrium we obtain:

$$\begin{aligned}
&P_0^0 A - P_1^0 - P_0^1 = 0\\
&P_s^n(A+n+s) - P_s^{n-1}A - P_s^{n+1}(n+1) - P_{s+1}^n(s+1) = 0\\
&P_s^N(A+N+s) - P_s^{N-1}A - P_{s-1}^N A - P_{s+1}^N(s+1) = 0\\
&P_S^N(N+S) - P_S^{N-1}A - P_{S-1}^N A = 0
\end{aligned} \tag{9-34}$$

To resolve this system we introduce the following expression S_r^m :

$$S_r^m(A) = \sum_{v=0}^{m} \frac{A^{m-v}}{(m-v)!}\binom{r-1+v}{v} \text{ where } S_r^m = 0 \text{ if } m < 0 \text{ or } r < 0 \tag{9-35}$$

The solution is then written:

$$P_i^j = \sum_{x=0}^{S-i} (-1)^x K_{i+x}\binom{i+x}{i} S_{i+x}^{j-x} \tag{9-36}$$

with $$K_k = \sum_{r=k}^{S} (-1)^{r-k} \binom{r-1}{k-1} a_r \ , \qquad K_0 = \frac{1}{S_1^{N+S}}$$

and $$a_r = \frac{1}{S_1^{N+S} S_r^N} \sum_{v=r}^{S} \binom{v-1}{r-1} S_0^{N+v}$$

The probabilities of the i states of the overflow block (states of second block) are given by:

$$Q(i) = \sum_{j=0}^{N} P_i^j$$

And so: $$Q(i) = \sum_{x=0}^{S-i} (-1)^x K_{i+x} \binom{i+x}{i} S_{i+1+x}^{N-x} \tag{9-37}$$

And finally the probability of utilisation of the second block (engaged) is:

$$P = \sum_{i=1}^{N} Q(i)$$
$$P = \sum_{i=1}^{N} \left[\sum_{x=0}^{S-i} (-1)^x K_{i+x} \binom{i+x}{i} S_{i+1+x}^{N-x} \right] \tag{9-38}$$

In practice, we will take action to ensure that this single additional block will suffice for traffic needs with a very high probability. In other words, the probability of needing a third block is negligible. This is first roughly checked with the simple Erlang formula, then *a posteriori* verified, and will be very generally the case taking account of the actual size of the blocks.

We can now calculate the total number of blocks necessary for D traffic flows.

We are here simplifying the calculation by considering that all the A_i are identical ($A_i = A, \forall i$), and that the blocks are of identical size.

As the traffic flows are independent, the binomial law is applied.

The probability of having $X = 2k+(D-k)$ blocks "engaged" (not necessarily totally occupied) is:

$$P(X) = \binom{D}{k} P^k (1-P)^{D-k} \tag{9-39}$$

where P is the previously calculated probability.

The average number of blocks engaged is:

$$\overline{X} = [(1-P) + P.2]D = (P+1)D \tag{9-40}$$

If we assume a pool of Y blocks, we can from these formulae calculate the probability of pool saturation, $P(Y)$, and thus determine the size of the pool.

We also have the pool utilisation factor: $\rho = \dfrac{\overline{X}}{Y}$. (9-41)

The previous results are generalised to the case in which x_0 blocks are systematically taken initially, with a single additional block for overflow. This models the actual case, as in general for each traffic flow a basic resource will be required, with a single additional one whose size is small compared to that of the basic resource.

We then have:

$$P(X) = \binom{D}{k} P^k (1-P)^{D-k}$$

where this time $P(X)$ is the probability of having $X = k(x_0+1)+(D-k)\, x_0$ blocks engaged, and the mean number of blocks taken is:

$$\overline{X} = [(1-P)x_0 + P(x_0+1)]D = (P + x_0)D \tag{9-42}$$

These results are also generalised to the case of different traffic flows, by replacing the binomial law by the multinomial law.

For example, let us study the case of a multiplex at 622 Mbit/s transporting voice traffic in ATM technology (AAL2).

The principle is as follows: the voice samples of calls going in the same direction are grouped together in one or more ATM cell(s) every 125 μs. On a multiplex of this type, 183 cells are available every 125 μs. An ATM cell can transport 48 voice samples (48 bytes of payload, plus 5 bytes of header). Let us now assume that we wish to transport D = 128 identical traffic streams, each with an intensity $A = 28.8$ E. We therefore initially allocate $x_0 = 1$ cell ($N = 48$ places) to each stream, and we will allocate to it a second cell when this is necessary, which will be released once it is empty. It is necessary to evaluate the probability of a lack of cells, bearing in mind that it is not possible to permanently allocate two cells to each traffic stream (256 would be needed), except by reducing the total traffic handled.

Figure 9.9 shows the probability of having a certain number of cells engaged.

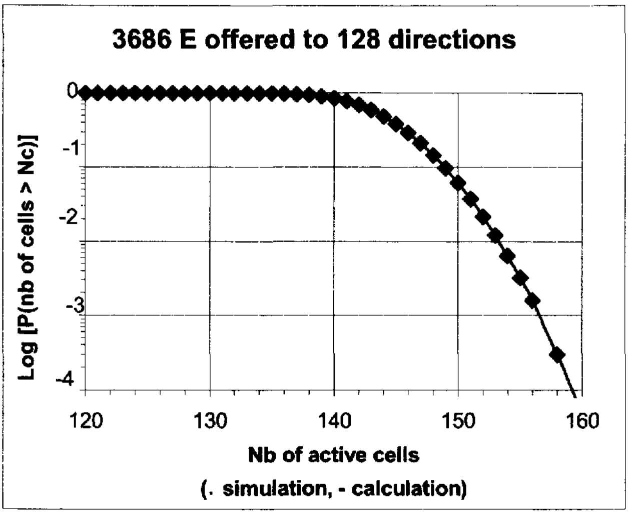


Figure 9.9. *Distribution of the number of active cells in pool*

The match between the simulation (dots) and the calculation (continuous curve) is excellent.

9.1.9. *Call repetition*

The model describes the macroscopic behaviour of users in the event of congestion.

In such circumstances (call rejection, extremely long wait, etc.) a phenomenon of impatience arises on the part of users, leading to the abandonment of calls and then insistent call renewal. These consecutive call attempts, most of which are doomed to failure, represent an additional load on network resources, causing more congestion and a snowball effect which is extremely detrimental to quality of service.

This phenomenon is extremely important in reality, particularly in the event of overloads, failures, traffic imbalance, etc. It has been shown that it leads, even in the event of initially slight overloads, to extremely severe overload situations. It will be the role of regulation, as seen in Chapter 1 and as analysed in the next paragraph, to protect systems and networks against these phenomena. It should also be noted that in these circumstances traffic observations should be carried out with great care. This is because it is difficult to distinguish between "fresh calls" and "reattempts" and the "blocking" rate observed will have a quite different significance.

The model below is a general model which links the final traffic offered to a server (handling system, transport system, network, etc.) to the traffic handled as a function of the probability of rejection by the server, and the perseverance rate of the source.

The phenomenon is characterised by the following phenomena:

– *traffic offered*: we will note as λ_0 the flow of 1st requests ("fresh" attempts) and as λ the total observed flow. Any client presenting him/her/herself in front of the service constitutes an attempt, which is a 1st attempt or a renewal;

– *repetition factor*: this is the ratio $\beta = \lambda / \lambda_0$, which measures the amplification caused by the congestion; it enables the linking of actual demand to the observation;

– *perseverance rate*: in a very simple model, allowance is made for perseverance of demand using a probabilistic method: let H be the probability that a failed attempt will be presented again (one possible refinement would be to take H as a function of the rank of the attempt, of the nature of the failure, etc.);

– *loss rate*: we note as p the probability of rejection of each attempt; this is assumed to be constant; reality is more complex (the failure rate differs depending on the rank of the attempt and the time interval between attempts);

– *efficiency rate*: we note as $r = \lambda_e / \lambda$ the efficiency rate. This is an apparent efficiency rate, as it does not correspond to fresh traffic λ_0. But this is the only one that is generally observable as it is difficult and very expensive to separate fresh attempts from renewed attempts in observations of systems and measurement campaigns.

Figure 9.10 shows the rejection and repetition mechanism in schematic form.

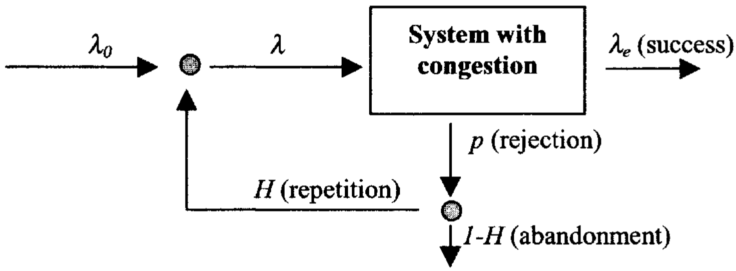


Figure 9.10. *The repeated attempts model*

We note as λ_e the intensity of the flow successfully handled as λ_{rnv} the flow from failures which give rise to a new attempt. This schematic representation enables the estimation of the flows in each branch:

$$\lambda_e = (1-p)\lambda$$

$$\lambda = \lambda_0 + \lambda Hp$$

hence:

$$\lambda = \frac{\lambda_0}{1-Hp}, \ \lambda_e = \frac{\lambda_0(1-p)}{1-Hp} \tag{9-43}$$

thus with $r = 1-p$:

$$\beta = \frac{1}{1-H(1-r)} \tag{9-44}$$

$$\lambda_0 = \frac{\lambda_e}{\beta r} \tag{9-45}$$

This model must be understood for what it is: it explains the behaviour observed, and in particular the difference between fresh, actual demand and observed traffic. Thus by measuring apparent offered traffic and handled traffic, we can work out the fresh traffic for a given source perseverance rate hypothesis.

An approximate analysis of the behaviour of a system of this type, and an evaluation of the various flows can be performed, assuming that the flow resulting from the renewed attempts retains its Poissonian nature.

By way of example, let us consider the case of a system modelled by an M/M/1/N queue.

We inject an unknown load λ_0. The measurement procedure supplies λ. It is assumed that the total resulting flow remains Poissonian, which would strictly require that the wait before renewal would tend towards infinity; we have made $N = 4$, and assumed a perseverance rate $H = 0.9$; the service time is taken to be 1. A sample of results is given below:

– for $_0 = 0.7$, we measure $= 0.78$, $p = 0.115$: calculation based on M/M/1/4 would give $p = 0.086$;

– for $_0 = 0.9$, we measure $= 1.2$, $p = 0.28$: calculation based on M/M/1/4 would give $p = 0.16$;

– for $_0 = 1.2$, we measure $= 3.07$, $p = 0.67$: calculation based on M/M/1/4 would give $p = 0.28$.

One can easily imagine the wrong interpretations that might result: either a traffic measurement underestimates rejection, or an actual rejection measurement results in an overestimate of ρ_0.

An approximate analytical model of this system can be given by assuming that the resulting flow of renewed attempts retains its Poissonian nature. We then estimate p by the rejection rate of the M/M/1/N queue:

$$p = \frac{\rho^N(1-\rho)}{1-\rho^{N+1}} \tag{9-46}$$

The previous relations gave us: $\rho = \dfrac{\rho_0}{1-Hp}$ (9-47)

The resolution is iterative: for a given value ρ_0, an initial value is chosen $\rho = \rho_0$ from which we deduce a value of p by the second formula, thus giving a new estimate of ρ, by the first formula, then a second estimate for p, etc., and so on up until convergence (precision of 1/1000 in a few iterations). The results are presented on the following charts:

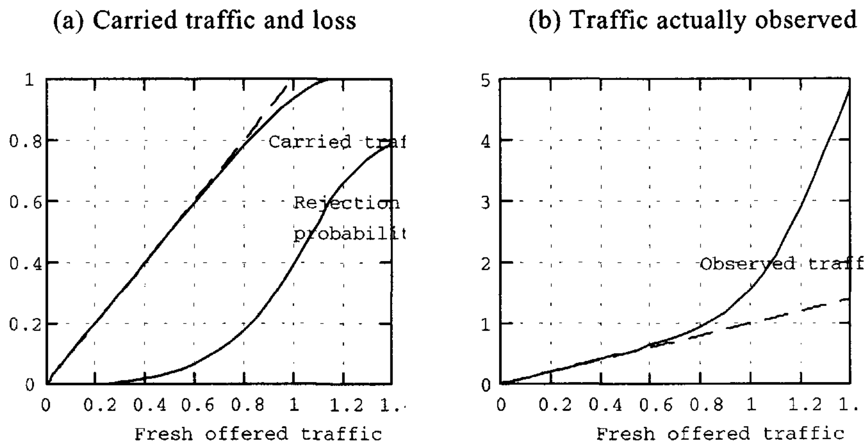


Figure 9.11. *Behaviour of an M/M/1/N system with repetition*

This model is applied not only to a control system but also to the case of a network of independent nodes (transport or signalling network) in the event of transmission errors resulting in repetitions.

Behaviour of a network with transmission error correction

Let us consider an end-to-end link of a data transmission network, whose nodes may be considered to be independent (see later the case of series queues).

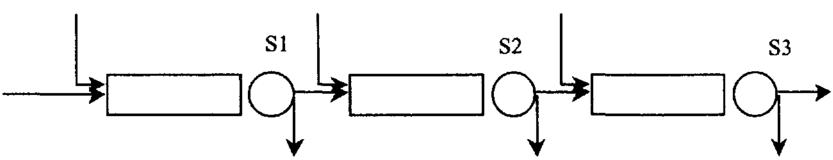


Figure 9.12. *End-to-end link configuration in a data network*

When a node is saturated (S3 for example), server S2 which precedes it will nevertheless handle and transmit a new packet that arrives; when it arrives at the full buffer of S3, the packet is rejected, even if it is correctly received. Node S2, which does not receive the expected acknowledgment, must retransmit the same packet once, twice, three times.: it is as though the packet was waiting by occupying server S2 and the transmission link until it is finally transmitted.

If we note as B the probability of rejection, the same at each stage (it is assumed that the link is homogeneous), and the same at each attempt (these assumptions are highly simplifying), then there is:

– one transmission, that is 1 service time, with a probability of $1 - B$,

– two transmissions, that is 2 service times, with a probability of $B(1 - B)$,

– three transmissions, that is 3 service times, with a probability of $B^2(1-B)$, etc.

Thus on average $1/(1-B)$ transmissions. This corresponds to taking an equivalent mean service time $E(s)/(1-B)$.

The service time increases with rejection, but also the load of transmission equipments is considerably increased.

At the input, we note as λ the arrival rate. The fresh traffic offered at each stage is λ: it is identical at each stage as it is assumed here that each rejection is corrected, with each packet occupying the node until transmission is successful. Each node is represented by an M/M/1/N model. Hence:

$$B = \frac{a^N(1-a)}{1-a^{N+1}}, \; a = \frac{\lambda E(s)}{1-B} \tag{9-48}$$

The equations are quite analogous to those of the previous model in which we would make $H = 1$, although their interpretation is different. The numerical resolution and the result would be comparable to the curve of the model in the previous section.

9.1.10. *Regulation*

The use of regulation mechanisms is essential to protect the systems and networks against overloads. It is not our concern here to consider a specific mechanism in detail, but to throw light on the fundamental relations which exist between calls (or transactions) that are accepted, rejected and prioritised.

The problem in fact is as follows: the system or network, in the event of an overload, i.e. in the event of an offered traffic which is greater than the handling capacity, must reject requests to preserve the correct processing of the calls accepted, but also handle some types of requests as a priority.

This is paradoxical in a sense:

On the one hand the system must refuse new requests as a saturated server has an infinite response time, and above all in practice in actual complex systems, queues will become saturated, information will be lost and there is a risk that no transactions will be completed.

On the other hand, even in an overload situation, the system must devote some of its time to identifying whether or not a new demand has priority, in order to accept or reject it. This may be more or less expensive, as in the case of calls to emergency numbers (fire service, police, etc.) for which the address requested must be analysed. It is thus possible to cause system congestion unnecessarily, as the proportion of calls which should really be prioritised is in fact low. Clearly therefore, compromise strategies must be found.

The model below enables an evaluation of the efficiency of the various strategies.

Let us consider a system to which the offered traffic has the following characteristics: *n* types of calls with the respective arrival rates: $\lambda_1,\dots \lambda_i,\dots \lambda_n$; τ_{iA} is the processing time for an *accepted call* of type *i;* τ_{iR} is the processing time for a *rejected* call (rejected for regulation reasons) of type *i*; and for a given system regulation state: a_i proportion of calls of type *i accepted;* r_i proportion of calls of type *i rejected*; where $a_i + r_i = 1$.

Let us take as a unit the mean processing time of an accepted call:

$$\overline{\tau_{iA}} = \frac{\sum_i \lambda_i \tau_{iA}}{\sum_i \lambda_i} \tag{9-49}$$

Let us term c_{iA} the relative "cost" of processing an accepted call:

$$c_{iA} = \frac{\tau_{iA}}{\overline{\tau_{iA}}} \tag{9-50}$$

and similarly for a rejected call:

$$c_{iR} = \frac{\tau_{iR}}{\overline{\tau_{iA}}} \tag{9-51}$$

and also the proportion for each call type:

$$p_i = \frac{\lambda_i}{\sum_i \lambda_i} \tag{9-52}$$

We thus have:

Load offered to the system:

$$\rho_{OFF} = \sum_i \lambda_i \tag{9-53}$$

System load:

$$\rho_{SYS} = \rho_{OFF} \sum_i p_i (a_i c_{iA} + r_i c_{iR}) \tag{9-54}$$

Processed load:

$$\rho_A = \rho_{OFF} \sum_i p_i a_i c_{iA} \tag{9-55}$$

Load rejected:

$$\rho_R = \rho_{OFF} \sum_i p_i r_i c_{iR} \tag{9-56}$$

Regulation in our model is governed by the following principle. Given the maximum load of the system $\rho_{SYS} = \rho_{MAX}$, the calls are rejected in accordance with a predetermined order as soon as the load reaches the value ρ_{MAX} and accepted as soon as the load falls below this value again. The system therefore oscillates between call rejection and call acceptance states. For the modelling with which we are concerned (very strong and very long overloads), we will neglect the oscillation phenomena which must be controlled by appropriate load measurement mechanisms (processor load measurement), or load evaluation mechanisms (e.g. call counting), by means of good adjustments of measurement periods and smoothing. The system remains in a given regulation state as long as the type(s) of calls rejected are sufficient to maintain its load below the value ρ_{MAX}. If however the rejection of the calls of type $x \le i$ is not sufficient, the system will begin to reject calls of type i+1, etc., bearing in mind that:

a) the types of calls with the highest priority are rejected last;

b) the higher the rejection level rises, the more drastic is the rejection, until it costs virtually nothing for the system ("blind" rejection, without analysis of the nature of the call).

Based on the previous equations, it is possible to describe the behaviour of the system being regulated depending on the strategies adopted.

Let us consider the example of the behaviour of a system with reference to two strategies.

Strategy 1 is the basic strategy, and is as follows:

As lower priority (first rejected) the non-urgent outgoing calls from users (type 1), and then the additional rejection of non-urgent incoming calls (type 2), and then the additional rejection of all outgoing calls (including urgent calls, type 3), and then the additional rejection of all incoming calls (including urgent calls, type 4).

Strategy 2 is more drastic:

From the second threshold onwards (once all type 1 calls have been rejected), all outgoing calls (type 1 and type 3) are rejected, and then this continues with the incoming calls.

Let us assume the following costs: $c_{1A} = c_{2A} = c_{3A} = c_{4A} = 1$, $c_{1R}= 0.25$, $c_{2R}= 0.1$, $c_{3R}= 0.1$, $c_{4R}= 0.1$.

(Note the low rejection cost of an incoming call compared to that of an outgoing call: the destination number has been analysed at the departure, then giving rise to a simple indication easier to analyse).

Furthermore, we assume for the sake of simplicity that the proportions of traffic remain the same in the various phases. This does not really correspond to reality for the transient regime, but does not make the model any less useful, for under actual conditions systems are very quickly subjected to a quasi-stationary regime of overload traffic. In addition, changes in the proportions may be assumed, and consequences evaluated.

The following thresholds are fixed: The first level of load rejection is 0.8, the second level of load rejection is 0.85, and the third level of load rejection is 0.9.

It is assumed that 60% of calls are outgoing, with 10% urgent calls, and that 40% of calls are incoming, also with 10% of urgent calls.

We thus obtain the curves shown in Figure 9.13, describing the fundamental behaviour of the system being regulated.

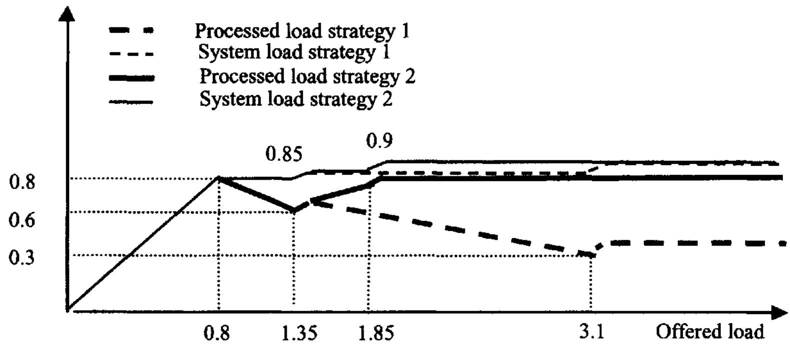


Figure 9.13. *Behaviour of a system under load regulation*

Strategy 1

The curve shown in the dotted line describes the system's behaviour for this strategy (strategy 1):

a) One can verify from the curve that the first regulation threshold is reached at an offered traffic of 0.8:

$0.8 = 0.8(0.6 \times 1 + 0.4 \times 1)$

b) Up to now all calls were accepted. Now non-urgent outgoing calls (type 1) must be rejected. Up to a traffic of about 1.35, the load of 0.8 is maintained, but it then rises, and the first rejection level no longer suffices.

$0.8 = 1.35 \times (0.6(0.9 \times 0.25 + 0.1 \times 1) + 0.4 \times 1)$

The load processed for an offered traffic of 1.35 is:

$1.35 \times (0.6(0.1 \times 1) + 0.4 \times 1)= 0.62$

We verify that the second threshold is reached at an offered traffic of 1.43:

$0.85 = 1.43 \times (0.6(0.9 \times 0.25 + 0.1 \times 1) + 0.4 \times 1)$

Note also that the handled load with traffic of 1.2 (1.5 times the reference traffic) is greater than 0.7. This represents 90% of the reference traffic (0.8), which corresponds to the objectives of the standards (see Chapter 2).

c) Beyond an offered traffic of 1.43, we must begin to also reject incoming non-urgent calls (type 2).

We verify that the load is maintained at 0.85 up to an offered traffic of 3.14, and then that the third threshold (0.9) is reached at an offered traffic of 3.32:

$$0.85 = 3.14 \times (0.6(0.9 \times 0.25 + 0.1 \times 1) + 0.4(0.9 \times 0.1 + 0.1 \times 1)$$

$$0.9 = 3.32 \times (0.6(0.9 \times 0.25 + 0.1 \times 1) + 0.4(0.9 \times 0.1 + 0.1 \times 1)$$

The traffic processed for an offered traffic of 3.14 is:

$$3.14 \times (0.6(0.1 \times 1) + 0.4 \times 0.1 \times 1) = 0.314$$

d) At this moment we begin to reject all outgoing calls without distinction (including type 3). We then verify that the system maintains its load at 0.9 up to a traffic of 6.6, and that the fourth threshold is attained at an offered traffic of 7, i.e. for an offered traffic almost 10 times greater than reference traffic:

$$0.9 = 6.6 \times (0.6 \times 0.1 + 0.4(0.9 \times 0.1 + 0.1 \times 1)$$

$$0.95 = 7 \times (0.6 \times 0.1 + 0.4(0.9 \times 0.1 + 0.1 \times 1)$$

The handled traffic for an offered traffic of 6.6 is:

$$6.6 \times (0.4 \times 0.1 \times 1) = 0.26$$

The final threshold is not considered, as with this policy it is unlikely that it would ever come into play.

This model demonstrates the usefulness of adopting quite rapidly a drastic rejection policy to reduce the cost of rejected calls and thus process more traffic, as we shall now see with strategy 2.

Strategy 2

The curve shown in an unbroken line describes the behaviour obtained if from the second threshold onwards (0.85), all outgoing calls, type 1 and type 3, are rejected (strategy 2).

Up to traffic levels of around 1.85, the load is maintained at 0.85 and then begins to rise. One can verify that the third threshold is reached at an offered traffic of 1.95:

$$0.85 = 1.85\ (0.6 \times 0.1) + 0.4 \times 1)$$

$$0.9 = 1.95\ (0.6 \times 0.1) + 0.4 \times 1)$$

The load processed with an offered traffic level of 1.85 is:

$$1.85 \times 0.4 \times 1 = 0.74$$

It is then possible to decide to reject all calls, without discrimination. As in the previous case, the system load is maintained at practically the same level, as the cost of rejected calls is negligible. We verify that this load level is maintained up to a traffic overload of 900%.

Experience proves that this strategy is the most appropriate one for dealing with real-life situations as in any case (particularly because of renewals), overloads very rapidly become very large and the most severe rejection thresholds are very quickly attained. Furthermore, note that with this strategy the system handles more traffic during very high overload, and thus even with blind rejection it gives more chance for an urgent call to be accepted on a reattempt.

9.2. Transport plane models

9.2.1. *Multi-bit rate traffic concentrator*

9.2.1.1. *Multi-bit rate Erlang model*

The aim here is to evaluate the handling capacity of a system that concentrates traffic from sources with different bit rates. This is typically the case of ISDN user concentrators, and synchronous network links, which have to handle calls with different bit rates (at n × 64 kbit/s for example). But above all, as we shall see later, analysis of this case forms the basis of the modelling of statistical multiplexing for ATM and IP traffic.

The aim is to evaluate the probability of blocking B (call rejection), for offered traffic A.

Let c_1 be the bit rate (or capacity) required by the source with the lowest bit rate. It is assumed that $c_1 = 1$, (c_1 is taken as the bit rate unit). The bit rates c_i

of the other sources will be approximated to multiples of c_1. C is the total capacity of the concentrator in bit rate units c_1.

The blocking probability is then simply given by the generalised Erlang formula applied to a system with $N = C$ servers.

The calls are grouped into bit rate classes: x is the number of bit rate classes, and a_i is the activity in erlangs of a bit rate class c_i.

The probability $P(n_1, ..., n_x)$ of having $(n_1, ..., n_x)$ calls in each class in progress is then:

$$P(n_1,...,n_x) = \frac{\prod_{i=1}^{x} \frac{a_i^{n_i}}{n_i!}}{\sum_{0 \le c_1 n_1 + \cdots + c_x n_x \le C} \left(\prod_{i=1}^{x} \frac{a_i^{n_i}}{n_i!} \right)} \tag{9-57}$$

and the blocking probability of a call with a bit rate c_i is:

$$B_i = \sum_{d_1 n_1 + \cdots + d_x n_x \ge C - d_i + 1} \mathrm{P}(n_1,\ldots,n_x) \tag{9-58}$$

and the mean blocking, for all types of call combined, is:

$$B = \frac{\sum_{i=1}^{x} B_i * a_i}{\sum_{i=1}^{x} a_i} \tag{9-59}$$

These fundamental formulae can give rise to relatively long calculations, and above all do not enable the rapid assessment of the impact of the mix of services. It is for this reason that approximations have been developed, all of which are based on the adjustment of moments of the exact distribution function with the moments of a simpler distribution. We present below the *peakedness factor* method, which is very simple to use and which is sufficiently precise for most practical purposes.

9.2.1.2. *Peakedness factor method*

It is simply a matter of adjusting the generalised Erlang law according to a simple Erlang law. We thus define a single equivalent bit rate, and also an equivalent number of servers. Note that in Chapter 3 we dealt with the case of the sum of independent random variables. We are now exactly in this case, as we are looking for the probability $P(n_1, ..., n_x)$ of having $(n_1, ..., n_x)$ calls in each class in progress with x independent and Poissonian classes of traffic. By applying the results of Chapter 3, we can replace x classes of traffic of bit rate d_i and activity a_i by a single equivalent traffic class with an activity of A/z and a single bit rate z, such that:

$$A = \sum_{i=1}^{x} a_i \, d_i \text{ and } z = \frac{\sum_{i=1}^{x} a_i \, d_i^2}{\sum_{i=1}^{x} a_i \, d_i} \tag{9-60}$$

z is called the peakedness factor because of its initial introduction by Hayward in overflow traffic problems. z reflects the burstiness of traffic. It is, it will be recalled, simply the ratio of the variance of the traffic to mean traffic $(V = \sum_{i=1}^{x} a_i \, d_i^2)$.

As a result, the model equivalent to the initial system is simply a concentrator of reduced capacity C/z, to which is offered a reduced traffic intensity A/z. And so the average blocking is:

$$B \approx E(\frac{C}{z}, \frac{A}{z}) \tag{9-61}$$

We then also deduce an estimate of blocking for the various traffic classes by interpolation between two successive values of z.

We write:

$$E(\frac{C}{z}-1, \frac{A}{z}) = E(\frac{C}{z}, \frac{A}{z})\alpha_1^{\,k}, E(\frac{C}{z}-2, \frac{A}{z}) = E(\frac{C}{z}-1, \frac{A}{z})\alpha_2^{\,k} \text{ etc.}$$

$$\text{where } \alpha = \left(\frac{E(\frac{C}{z}-1, \frac{A}{z})}{E(\frac{C}{z}, \frac{A}{z})} \right)^{1/z} = \frac{C}{A} \tag{9-62}$$

We also have:

$$B_1 = \frac{1}{z} E(\frac{C}{z}, \frac{A}{z}) \tag{9-63}$$

and thus for a bit rate c_2 which is for example equal to $2\ z + n$, where $0 < n < z$, we have:

$$B_2 = \frac{1}{z} E(\frac{C}{z}, \frac{A}{z}) \alpha_1^{\kappa} \alpha_2^{\kappa} \alpha_3^{n}$$

When d_i is not too large, it is possible to simplify, by keeping α constant:

$$B_i \approx E(\frac{C}{z}, \frac{A}{z}) \left(\frac{1}{z} \frac{\alpha^{d_i} - 1}{\alpha - 1} \right) \tag{9-64}$$

Table 9.2 provides an example of the precision of the approximation in the case of a mixture of two classes of traffic. It will be noted in particular that for many practical purposes the linear approximation is sufficient:

$$B_i = \frac{1}{z} E(\frac{C}{z}, \frac{A}{z}) d_i \tag{9-65}$$

Table 9.2. *Accuracy of the "peakedness factor" method*

$z = 2$

						Erlang Loss Generalised Erlang			Erlang Loss Peakedness method		
C	a_1	d_1	a_2	d_2	A	B_1	B_2	B	A/z	C/z	E(C/z,A/z)
200	105	1	7	5	140	$9.71\ 10^{-5}$	$6.84\ 10^{-4}$	$1.34\ 10^{-4}$	70	100	$1.38\ 10^{-4}$

$z = 10$

C	a_1	d_1	a_2	d_2	A	B_1	B_2	B	A/z	C/z	E(C/z,A/z)
200	38.5	1	1.5	21	70	$9.18\ 10^{-6}$	$6.83\ 10^{-5}$	$3.45\ 10^{-5}$	7	20	$2.99\ 10^{-5}$

The *peakedness factor* method of course remains an approximate method, but it can easily be verified that the precision is excellent as long as the C/d_x ratio remains large, with d_x being the highest required bit rate. This condition can be verified in most practical multiplexing cases, as we shall see later, in view of the high bit rates of the links.

We provide below some examples of results in order to validate the domain of validity of the model. We consider a mixture of two types of traffic with a total intensity of A, offered to a system of C servers:

a) $C = 200$, $a_1 = 105$, $d_1 = 1$, $a_2 = 7$, $d_2 = 5$ => $A = 140$, $z = 2$, $A/z = 70$, $C/z = 100$

Generalised Erlang: $B = 1.34\ 10^{-4}$, Peakedness method: $B = 1.38\ 10^{-4}$

b) $C = 100$, $a_1 = 45$, $d_1 = 1$, $a_2 = 3$, $d_2 = 5$ => $A = 60$, z = 2, A/z = 30, $C/z = 50$

Generalised Erlang: $B = 2.5\ 10^{-4}$, Peakedness method: $B = 2.2\ 10^{-4}$

c) $C = 50$, $a_1 = 18$, $d_1 = 1$, $a_2 = 1{,}2$, $d_2 = 5$ => A = 24, z = 2, A/z = 12, $C/z = 25$

Generalised Erlang: $B = 5.35\ 10^{-4}$, Peakedness method: $B = 3.78\ 10^{-4}$

d) $C = 20$, $a_1 = 3.75$, $d_1 = 1$, $a_2 = 0.25$, $d_2 = 5$ => $A = 5$, $z = 2$, $A/z = 2.5$, $C/z = 10$

Generalised Erlang: $B = 7{,}52\ 10^{-4}$, Peakedness method: $B = 2.16\ 10^{-4}$

It is immediately verified that precision depends on the ratio between the total capacity of the link and the highest bit rate (d_2). To obtain sufficient precision, a ratio greater than about 10:1 is required.

This type of result can be verified even for large values of z and large ratios between bit rates (for example a ratio of 21:1, as below, may corresponds to one bit rate of 64 kbit/s and another of 1 Mbit/s):

e) $C = 200$, $a_1 = 38.5$, $d_1 = 1$, $a_2 = 1{,}5$, $d_2 = 21$ => $A = 70$, $z = 10$, $A/z = 7$, $C/z = 20$

Generalised Erlang: B= $3.45\ 10^{-5}$, Peakedness method: $B = 2.99\ 10^{-5}$

f) $C = 100$, $a_1 = 8$, $d_1 = 1$, $a_2 = 0.75$, $d_2 = 16$ => $A = 20$, $z = 2$, $A/z = 2$, $C/z = 10$

Generalised Erlang: $B = 7{,}94\ 10^{-5}$, Peakedness method: $B = 3.82\ 10^{-5}$

9.2.2. *Multiplexing*

9.2.2.1. *Multiplexing of identical periodical sources*

Let us consider first the important case of identical periodic sources, such as for example the case of CBR (constant bit rate) ATM sources. This indicates the properties that are useful to simplify the processing of sources of different bit rates and different packet lengths.

Let us consider a multiplexing system whose output link capacity is one cell per time unit, to which is offered the traffic of N sources, $N < D$, each of which transmits one cell every D time units. This system is called N/D/D/1.

The system is shown in schematic form in Figure 9.14.

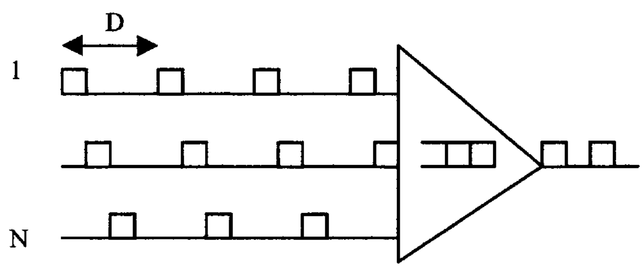


Figure 9.14. *Multiplexing of identical deterministic sources*

Let us call V_D the unfinished work at an arbitrary instant D. If $N < D$, there is at least one instant during the interval $(0,D)$ in which the system is empty. Therefore V_D depends only on arrivals after this instant. If we consider that the N arrivals are uniformly distributed between 0 and D, this gives:

$$Q(x) = \Pr\{V_D > x\} = \sum_{x<n\leq N} \Pr\{v(D',D=n\}.\Pr\{V_{D'} = 0/v(D',D)=n\}$$

where $v(D',D)=n$ the number of arrivals in the interval (D', D), and $D'=D-n+x$ (as x arrivals have still to be handled).

This situation is illustrated in Figure 9.15.

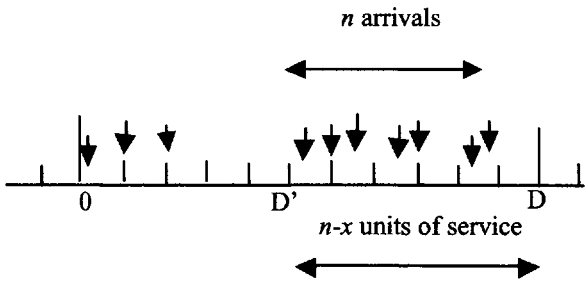


Figure 9.15. *Unfinished work calculation*

Because the arrivals are uniform $(n-x)/D = (D-D')/D$ expresses the probability, for an arrival, of being situated in the interval $(D-D')$.

Therefore:

$$\Pr\{\nu(D',D)=n\}=\binom{N}{n}\left(\frac{n-x}{D}\right)^n\left(1-\frac{n-x}{D}\right)^{N-n} \tag{9-66}$$

Similarly, because $N<D$ and $\nu(D',D)=n$, this gives $N-n<D'$, and the probability of having $V_{D'}=0$ is that of having $D'-(N-n)$ service units free in $(0, D')$, that is:

$$\Pr\{V_{D'}=0/\nu(D',D)=n\}=\frac{D-N+x}{D-n+x} \tag{9-67}$$

We thus obtain:

$$P(>x)=Q_{\mathrm{N,D}}(x)=\sum_{n=x+1}^{N}\binom{N}{n}\left(\frac{n-x}{D}\right)^n\left(1-\frac{n-x}{D}\right)^{N-n}\left(\frac{D-N+x}{D-n+x}\right) \tag{9-68}$$

It may be verified from Figure 9.16 that this formula gives (for high loads) results that are substantially different from those obtained by the formula M/D/1, even for relatively large values of N. The periodic nature of the arrivals is fundamental.

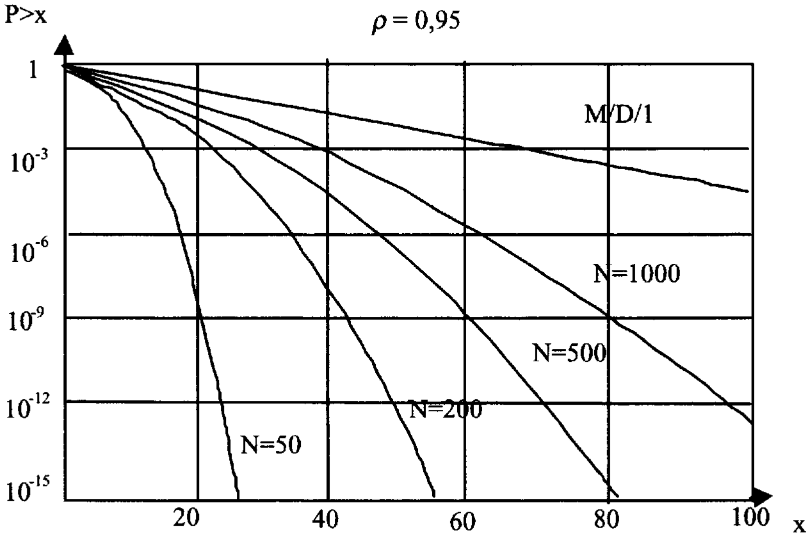


Figure 9.16. *Queue distribution for ND/D/1*

The difference is attenuated, however, at low loads, as shown in Table 9.3.

Table 9.3. *Comparison of M/D/1 and N/D/D/1*

X	N	ρ = 0.1	ρ = 0.3	ρ = 0.5	ρ = 0.7	ρ = 0.9
6	30	4.2e-11	5.0e-7	6.0e-5	1.7e-3	2.8e-2
	70	8.0e-11	1.1e-6	1.7e-4	5.8e-3	9.5e-2
	200	1.1e-10	4.4e-6	2.7e-4	1.2e-2	1.9e-1
	M/D/1	1.2e-10	1.7e-6	3.5e-4	1.3e-2	2.7e-1
12	30		1.7e-14	1.1e-10	4.4e-8	6.2e-6
	70		7.6e-13	7.1e-9	7.4e-6	1.3e-3
	200		3.8e-12	6.7e-8	7.1e-5	2.0e-2
	M/D/1		5.2e-12	1.9e-7	2.5e-4	7.6e-2
16	30			1.1e-15	2.0e-12	9.0e-10
	70			5.7e-12	3.2e-8	2.6e-5
	200			2.0e-10	1.9e-6	3.0e-3
	M/D/1			1.2e-9	1.7e-5	3.3e-2
32	70					1.8e-16
	200					1.5e-8
	1000					1.8e-4
	M/D/1					1.25e-3

Approximation of formula

For practical reasons, and in particular for having better visibility of the impact of the various parameters, it may be practical to have a very simple approximation for rapid calculations (during an "on line" performance audit for example, or when presenting the characteristics of a product), and in order to better grasp the impact of the various factors (load, etc.). For this purpose, we will now establish two simple expressions for the queues M/D1 and N/D/D/1.

A high traffic approximation of the M/D/1 queue is given by:

$$\mathrm{P}(>x) \approx e^{-2x\left(\frac{1-\rho}{\rho}\right)}, \text{ of the form } e^{-\alpha x} \tag{9-69}$$

(We will not develop the calculations here. For this the reader will refer to specialised books or publications concerning heavy traffic approximations and Brownian bridge approximations.)

Using the same method [ROB 96], the following approximation for N/D/D/1, also at heavy load, is suggested:

$$\mathrm{P}(>x) \approx e^{-2x\left(\frac{x}{N}+\frac{1-\rho}{\rho}\right)} \tag{9-70}$$

This expression takes the following form:

$$P(>x) \approx e^{-x(\frac{2x}{N}+a)}$$

Bearing in mind that for the M/D/1 queue we had the following very precise approximation, valid even for low loads (see Chapter 7):

$$P(>x) \approx \frac{(1-\rho)}{\rho e^{\beta_0}-1} e^{-\beta_0 x}$$

where β_0 is such that $\rho + \beta_0 - \rho e^{\beta_0} = 0$.

Let us use MacLaurin's expansion of $\ln(y)$; we thus have:

$\ln(\beta_0 + \rho) \approx 2\dfrac{\beta_0 + \rho - 1}{\beta_0 + \rho + 1}$, by retaining only the first term.

From the expression in β_0 we deduce $\beta_0 = \ln(\beta_0 + \rho) - \ln\rho$, thus:

$\beta_0 \approx 2\dfrac{\beta_0 + \rho - 1}{\beta_0 + \rho + 1} - \ln\rho$, hence:

$\beta_0^{\;2} + \beta_0(\rho + \ln\rho - 1) = -\ln\rho(\rho + 1) + 2(\rho - 1)$.

Given that $-\ln\rho(\rho + 1) + 2(\rho - 1) \approx 0$, by again using here the first term of MacLaurin's expansion of ln , thus: $\beta_0 \approx 1 - \rho - \ln\rho$.

And thus finally, for M/D/1, we obtain:

$$\mathrm{P}(>x) \approx -\frac{1-\rho}{\ln(\rho)} e^{-(1-\rho-\ln(\rho))x} \tag{9-71}$$

An expression of the form $ae^{-\beta x}$ from which we deduce for N/D/D/1, as seen previously, an expression of the form

$P > x \approx ae^{-x(\frac{2x}{N}+\beta)}$

Thus for N/D/D/1:

$$\mathrm{P}(>x) \approx -\frac{1-\rho}{\ln(\rho)} e^{-x\left(\frac{2x}{N}+1-\rho-\ln(\rho)\right)} \tag{9-72}$$

Table 9.4 indicates the precision of the approximation.

The precision can be considered to be quite sufficient for the first calculations aimed at obtaining orders of magnitude.

Table 9.4. *Approximation of M/D/1 and N/D/D/1*

X	N	ρ = 0.1		ρ = 0.3		ρ = 0.5		ρ = 0.7		ρ = 0.9	
		Exact	App.	Exact	App.	Exact	App.	Exact	App.	Exact	App.
6	30	4.2e-11	1.6e-10	5.0e-7	5.7e-7	6.0e-5	5.0e-5	1.7e-3	1.5e-3	2.8e-2	2.5e-2
	70	8.0e-11	6.3e-10	1.1e-6	1.1e-6	1.7e-4	2.0e-4	5.8e-3	5.8e-3	9.5e-2	9.8e-1
	200	1.1e-10	1.2e-9	1.4e-6	4.4e-6	2.7e-4	3.9e-4	1.2e-2	1.1e-2	1.9e-1	1.9e-1
	M/D/1	1.2e-10	1.7e-9	1.7e-6	6.3e-6	3.5e-4	5.6e-4	1.3e-2	1.6e-2	2.7e-1	2.7e-1
12	30			1.7e-14	4.7e-15	1.1e-10	2.9e-11	4.4e-8	2.2e-8	6.2e-6	5.5e-6
	70			7.6e-13	1.1e-12	7.1e-9	7.1e-9	7.4e-6	5.2e-6	1.3e-3	1.3e-3
	200			3.8e-12	1.6e-11	6.7e-8	1.0e-7	7.1e-5	7.5e-5	2.0e-2	1.9e-2
	M/D/1			5.2e-12	6.9e-11	1.9e-7	4.3e-7	2.5e-4	3.2e-4	7.6e-2	8.0e-2
16	30					1.1e-15	1.4e-16	2.0e-12	8.9e-13	9.0e-10	1.4e-9
	70					5.7e-12	2.5e-12	3.2e-8	1.5e-8	2.6e-5	2.4e-5
	200					2.0e-10	2.8e-10	1.9e-6	1.8e-6	3.0e-3	2.7e-3
	M/D/1					1.2e-9	3.6e-9	1.7e-5	2.3e-5	3.3e-2	3.5e-2
32	70									1.8e-16	2.6e-16
	200									1.5e-8	4.7e-8
	1000									1.8e-4	4.7e-4
	M/D/1									1.2e-3	1.3e-3

9.2.2.2. *Multiplexing of bursty sources*

This time the sources are not periodic, but alternate between periods of transmission (*Ton*) and periods of silence (*Toff*) .

In transmission phase, the source is periodic with the period *D*, as previously. This type of source is called ON/OFF or "bursty".

Figure 9.17 describes the behaviour of a source.

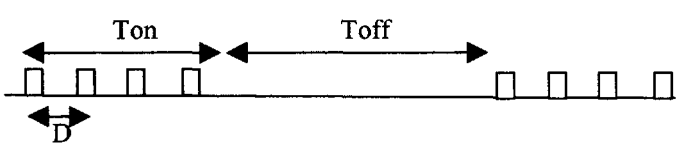


Figure 9.17. *Activity of an* ON/OFF *source*

We will begin with a few general considerations about the behaviour of a bursty sources multiplexer which, although relating to observations, will enable us to model the system studied relatively simply.

In the general case in which $N > D$, we observe the behaviour presented in Figure 9.18.

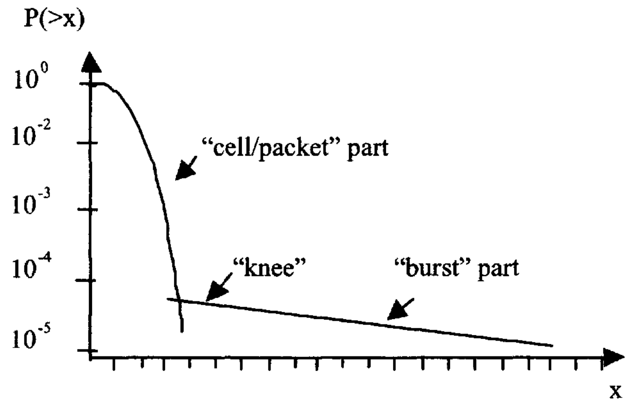


Figure 9.18. *Typical behaviour of a bursty sources multiplexer*

It will be noted that the distribution function of the number of *cells* waiting is composed of two clearly separate parts: one part called the *cell* or *packet* part with a rapid decay and one part called the *burst* part with a slow decay.

This is clearly explicable when one notes that in the first part the multiplexer is "weakly" saturated (a more "common" situation), only cells (or packets) are waiting, whereas in the second part the multiplexer is "strongly" saturated (a more "rare" situation), and bursts are kept waiting: the longer the bursts, the longer the queue and the slower the decay. The point of transition between the two parts is generally called the "*knee*".

The attentive reader will have noticed the analogy between the appearance of the cell/packet part and that of the multiplexing of N periodic sources as seen earlier. The behaviour in the two cases is identical, as we shall now verify.

9.2.2.3. *Modelling of cell/packet part*

The behaviour of the multiplexer in this part can be evaluated precisely in the case of identical sources. This reflects a situation in which the probability of saturation is negligible, i.e. the probability of having more than D sources in the ON state simultaneously is negligible. In this case we have the following remarkable result.

We will look at the case of N identical sources, and as before we consider the case in which $N < D$. This case will enable us to draw important conclusions for more general cases.

From the previous study on the superimposition of periodic sources, it is easy to see that the following can be written:

$$P(>x) = \sum_{n=0}^{N} P_n Q_{n,D}(x) \tag{9-73}$$

where $P_n = \binom{N}{n} p^n (1-p)^{N-n}$ with $p = \dfrac{Ton}{Ton + Toff}$ express the probability for n sources of being in the ON state.

By developing this result, we obtain:

$$P(>x) = Q_{N,D/p}(x) = Q_{N,D_{moy}}(x) \tag{9-74}$$

In fact:

$$P(>x) = \sum_{n=0}^{N} P_n Q_{n,D}(x)$$

$$P(>x) = \sum_{n=x+1}^{N} \binom{N}{n} p^n (1-p)^{N-n} \sum_{m=x+1}^{n} \binom{n}{m} \left(\frac{m-x}{D}\right)^m$$
$$\left(1 - \frac{m-x}{D}\right)^{n-m} \left(\frac{D-n+x}{D-m+x}\right)$$

given that: $\sum_{n=x+1}^{N} \sum_{m=x+1}^{n} = \sum_{m=x+1}^{N} \sum_{n=m}^{N}$ thus:

$$P(>x)=\sum_{m=x+1}^{N}\ \sum_{n=m}^{N}\binom{N}{n}\binom{n}{m}p^{n}(1-p)^{N-n}\left(\frac{m-x}{D}\right)^{m}$$

$$\left(1-\frac{m-x}{D}\right)^{n-m}\left(\frac{D-n+x}{D-m+x}\right)$$ and by introducing $k=n-m$

$$P(>x)=\sum_{m=x+1}^{N}\ \sum_{k=0}^{N-m}\frac{N!}{(N-k-m)!}\frac{1}{k!m!}p^{m+k}(1-p)^{N-k-m}$$

$$\left(\frac{m-x}{D}\right)^{m}\left(1-\frac{m-x}{D}\right)^{k}\left(\frac{D-k-m+x}{D-m+x}\right)$$

We can also write that: $\dfrac{N!}{(N-k-m)!}\dfrac{1}{k!m!}\dfrac{(N-m)!}{(N-m)!}=\binom{N}{m}\binom{N-m}{k}$, and so by grouping:

$$P(>x)=\sum_{m=x+1}^{N}\binom{N}{m}p^{m}\left(\frac{m-x}{D}\right)^{m}(1-p)^{N-m}\sum_{k=0}^{N-m}\binom{N-m}{k}\left(\frac{p}{1-p}\right)^{k}$$

$$\left(1-\frac{m-x}{D}\right)^{k}\left(1-\frac{k}{D-m+x}\right)$$

Let us introduce $\alpha=\left(\dfrac{p}{1-p}(1-\dfrac{m-x}{D})\right)$, so that the second summation is written:

$$\sum_{k=0}^{N-m}\binom{N-m}{k}(\alpha)^{k}\left(1-\frac{k}{D-m+x}\right)=(1+\alpha)^{N-m}-\frac{N-m}{D-m-x}\alpha(1+\alpha)^{N-m-1}=$$

$$(1+\alpha)^{N-m}\left(1-\frac{\frac{N-m}{D-m-x}\alpha}{1+\alpha}\right)$$

and thus by transferring into $P(>x)$ and after calculation, we obtain:

$$P(>x)=\sum_{m=x+1}^{N}\binom{N}{m}\left(p\frac{m-x}{D}\right)^{m}\left(1-p\frac{m-x}{D}\right)^{N-m}\left(\frac{D-p(N-x)}{D-p(m-x)}\right)$$

We thus arrive at (9-74):

$$P(>x)=Q_{N,D/p}(x)=Q_{N,D_{moy}}(x)$$

This result is remarkable as it relates the multiplexing of bursty sources to that of periodic sources: only the mean period, i.e. the mean bit rate, counts. And above all, once again, the performances are better than those of an M/D/1 queue (with the appropriate D). Given that the M/D/1 queue leads to queue sizes which are quite acceptable, very generally this model should be sufficient to evaluate quality of service, under these conditions.

This result can also be generalised to a number of sources $N > D$. One can conclude that, even for a number of sources $N > D$, as long as the probability of exceeding the output link bit rate of the multiplexer is very low, i.e. as long as the probability of having more than D sources in the ON state simultaneously is negligible, the system behaves like a multiplexer of periodic sources and better than an M/D/1 queue.

Finally, this property can be generalised to the case of different bit rate sources, as long as there is still a low probability of having a number of simultaneously active sources so that the link capacity is exceeded. The behaviour will be even better than that of an M/D/1 queue.

This will therefore be a pessimistic model for many practical cases of multiplexing and of quality of service evaluation, with regard to the *cell* or *packet* part. Finally, in the general case of variable packet sizes, it may be approximated to an M/G/1 queue.

9.2.2.4. *Modelling of the burst part*

In this part we are mainly interested in determining the value of the *knee* in the curve that we mentioned above.

We are assuming that $N >> D$, which corresponds to real-life situations whose purpose is to take advantage of the multiplexing of a large number of highly bursty sources.

Let us first consider the case in which the sources have an identical peak bit rate (identical bit rate in bursty periods) with an inter-arrival time for bursts from a source obeying a negative exponential distribution.

Under these hypotheses, the arrivals of the bursts can be considered as Poissonian. Accordingly, the multiplexer can be considered as a concentrator with D servers, and the saturation probability can be estimated simply by applying the Erlang law (or by using the Poisson law for very high D and N values).

The Erlang model is particularly well suited to real-life systems with large buffers, as in this case the multiplexer acts as a burst concentrator with D servers with a queue. If p is the probability for a source of being in transmit mode (ON state), then this gives $A = Np$ and:

$$Psat = E(D,A) \;\; \text{(Erlang)} \;\; \text{or} \;\; Psat = \sum_{D \le n}^{\infty} \frac{A^n}{n!} e^{-A} \;\; \text{(Poisson)} \tag{9-75}$$

Let us now consider the case of sources with different bit rates. Let us call more generally C the capacity of the link (C Mbit/s for example) and d_i the bit rate, or capacity, required by a source of type i, and finally a_i its activity (proportion of time in ON state).

We return here of course to the case of the multi-bit rate. Accordingly, we can apply the results that were previously established. And in particular, to simplify the calculations, apply the approximate formulae.

The x classes of traffic with bit rate d_i and activity a_i are replaced by a single class of traffic with activity A/z and a single bit rate z, such that:

$$A = \sum_{i=1}^{x} a_i \, d_i \quad \text{and} \quad z = \frac{\sum_{i=1}^{x} a_i \, d_i^2}{\sum_{i=1}^{x} a_i \, d_i} \tag{9-76}$$

z, it will be recalled, is simply the ratio of the variance of the traffic to its mean value: ($V = \sum_{i=1}^{x} a_i \, d_i^2$).

The model equivalent to the initial system is simply a multiplexer with a reduced capacity C/z, to which is offered a traffic of reduced intensity A/z, whose saturation probability is:

$$Psat \approx E(\frac{C}{z}, \frac{A}{z}) \tag{9-77}$$

9.2.3. *Equivalent bandwidth*

The concept of equivalent bandwidth is used to simplify the call acceptance and session mechanisms, etc., in systems supporting services with different or variable bit rate characteristics.

The aim is in fact to calculate the performance of the communication support if the new request is accepted, taking into account the characteristics of the calls already accepted, and to compare it with the required quality of service criterion. It will easily be understood that, if this could be limited to the simple problem of the pure sum of identical bit rates, as in the case of calls at 64 kbit/s, call acceptance would consist only of testing compliance with the maximum load allowed for the link, or the maximum number of calls in progress allowed on a link.

A great deal of research has been done on this issue, and we will develop here only those aspects which relate to link saturation evaluation, as in practice these are the ones that relate to the fundamental problem of multiplexing, as we saw earlier.

In the case of traffic with constant but different bit rates, we have seen that the saturation probability depends not only on the load but also on the mixture of bit rates. The problem is raised in the same way when it is a matter of multiplexing ON-OFF type sources with different bit rates. If we consider that call and burst arrivals are Poissonian, the solution is obtained by applying the Erlang multi-bit rate formula, or approximately by the *peakedness factor* method, as already presented for the modelling of statistical multiplexing.

To avoid any long calculation times, the following empirical formula for equivalent bandwidth has been proposed [ROB 96]:

$$e_i = \alpha(B)m_i + \beta(B)\sigma_i^2 / C \tag{9-78}$$

where m_i and σ_i^2 are the mean and variance of the bit rate of calls of type i, α (B) and β (B) are factors depending on the required saturation probability B, and C is the capacity (total bit rate) of the link. Empirically, the factors α (B) and β (B) have been adjusted to:

$$\alpha = 1 - \frac{\log B}{50}, \beta / \alpha = -6 \log B \tag{9-79a}$$

For example, for B=10^{-4} and B=10^{-9}, the following is obtained:

$$e_{i(10-4)} = 1{,}08 m_i + 26 \sigma_i^2 / C \text{ and } e_{i(10-9)} = 1{,}18 m_i + 64 \sigma_i^2 / C \tag{9-79b}$$

It is then quite simple to check, each time a new call request occurs, that the sum of the equivalent bandwidth is lower than the capacity of the link C to decide whether or not to accept the call.

This empirical approach can be very simply explained, in the case of a large number of sources, by considering first that the traffic offered is approximately a Poisson one, and second that the final distribution resulting from the superimposition of different types of sources can itself be approximated by the same distribution as in the case of a single type (Poisson or Erlang distribution). The quantile (at 10^{-x}) is then obtained with a formula such as (9-79) and with the same factors, for individual traffics as for the sum of traffics. Here again we have the type of approximation already used by the peakedness factor method.

To express this in more concrete terms, let us first consider the case of identical ON-OFF sources, with a peak rate d, and an activity a (proportion of time in the ON state).

The mean is then $m = ad$ and the variance is $\sigma^2 = m\ (d-m)$.

The equivalent bandwidth formula becomes:

$$e = \alpha m + \beta m(d-m)/C.$$

Let us call N the ratio C/d. If N is sufficiently large and m small with respect to d, we may write:

$e = \alpha m + \beta m/N$, and in general for a source of type i:

$e_i = \alpha\, m_i + \beta m_i/N_i.$

That is to say that the maximum load of the link, for a saturation probability of 10^{-x}, is given by $\rho_i = m_i/e_i$ (the link cannot accept more than C/e sources simultaneously with a mean bit rate m).

And thus we have the equation:

$$1/\rho_i = \alpha + \beta/N_i$$

We may now compare these results with those obtained using the Erlang formula (see Table 9.5).
It may be observed that under the conditions stated, i.e. with a large N, or a small d in relation to C, coherence is satisfactory. In fact the equivalent bandwidth formula is only a linear approximation of the Erlang distribution (or the Poisson law if N is very large).

Table 9.5. *Equivalent bandwidth and Erlang formula*

N = C/d	A (Erlang) at 10^{-4}	1/ρ = N/A	1/ρ = 1.08+26/N	A (Erlang) at 10^{-9}	1/ρ = N/A	1/ρ = 1.18+64/N
1200	1101	1.09	1.10	1013	1.2	1.23
600	525	1.14	1.12	470	1.27	1.28
300	246	1.22	1.17	210	1.42	1.39
75	48,6	1.54	1.43	34.5	2.2	2.03
60	36.6	1.64	1.51	25	2.4	2.24
40	21.4	1.87	1.73	13.2	3.03	2.78
20	7.7	2.6	2.38	3.5	5.7	4.38
10	2.26	4.4	3.68	0.65	15	7.58

The result can now be extended to the case of different types of sources with different bit rates. Here again, the peakedness factor method is used, which consists in fact of approximating the distribution resulting of the superimposition to an Erlang distribution for a system handling traffic equivalent with the single bit rate z.

We had:

$B = E\ (N/z, A/z)$

where z is the *peakedness factor*, a function of the mix of bit rates, $A = \Sigma a_i\ d_i$, (each type of traffic has an activity a_i and a peak rate d_i), $z = \Sigma a_i\ d_i^2\ /\Sigma a_i\ d_i$. The same type of linear approximation as before thus gives:

$(N/z)/(A/z) = \mathrm{k} = \alpha(B) + \beta(B)/(N/z)$

Hence:

$N = \alpha\ \Sigma a_i\ d_i + \beta\ \Sigma a_i\ d_i^2\ /N = \alpha\ \Sigma m_i + \beta\ \Sigma m_i\ d_i$

And thus for example, for two types of source, if we call e_1 and e_2 the equivalent bit rate used by each class of traffic:

$e_1 + e_2 = \alpha\ m_1 + \beta\ m_1\ d_1\ /N + \alpha\ m_2 + \beta\ m_2 d_2\ /N$

Hence:

$$e_1 = \alpha m_1 + \beta\ m_1\ d_1\ /N, \tag{9-80}$$

$e_2 = \alpha m_2 + \beta\ m_2 d_2\ /N,$

$1/\rho_1 = \alpha + \beta\ /N_1,\ 1/\rho_2 = \alpha + \beta\ /N_2$

We thus find again of course the basic relations of the equivalent bandwidth for each source. In these application cases (large N), and for approximate calculations, the peakedness factor method and the equivalent bandwidth method are identical. They may be considered to be "symmetrical": one will be preferentially used for dimensioning and the other will be used for call acceptance. Finally, it may be noted that the actual capacities of the links are such that the assumption that N is large is very generally true. This is why these models are so useful.

9.2.4. *Modelling and multiplexing of IP traffic*

In this model we will consider that IP traffic has the following fundamental characteristics:

– the traffic originates from a large number of users and the process of individual requests at session level, as at call level in telephony, forms a Poisson process. Similarly global demand, at these levels, is predictable, and results from the daily activity of the users;

– the IP traffic can be broken down into three main levels (see Chapter 1):

- session level: the sessions arrive in accordance with a Poisson process of parameter ρ. For a link with a capacity C, if λ is the arrival rate and v is the mean volume per session (in bits, for example, the product of mean bit rate and duration), the load of the link is then noted $\rho = \lambda \ v/C$,

- flow level: each session is made up of a succession of flows and periods of silence. The flows consist of the transfer of files, e-mails, images, etc. Traffic at flow level is "*bursty*", and the volume of flows is also extremely variable (we will see how to deal with this problem by drawing a distinction between two main types of flow),

- packet level: traffic at packet level is extremely bursty and has so-called *self similarity* characteristics, particularly because of the interaction of the origin traffic with flow control mechanisms (TCP) and error correction mechanisms;

– IP traffic may be structured into two main categories of flow:

- real-time flows: such flows consist of the real-time transmission of voice or video type data, usually under the control of the UDP protocol (no flow control, no retransmission). Performance is mainly characterised by the minimisation of the transfer delay (impact on the perception of interactivity by the user, possible need of echo cancellers), by the minimisation of jitter and compliance with an intrinsic rate (necessity of image and voice samples synchronisation),

- elastic flows: such flows consist of the transfer of files, e-mails, Web pages, etc., usually under the control of the TCP protocol (flow control, retransmission). The real time constraint is less restrictive: for example, transfer delays in the order of one second remain acceptable. Performance is primarily characterised by the total transfer duration, or the effective average bit rate corresponding to the ratio of volume to duration.

The IP traffic multiplexing model will be based on these characteristics as follows. On one hand, a distinction is drawn between *real time* traffic and *elastic* traffic, and most importantly (non-pre-emptive) priority is given to real time traffic. This means that initially the required bandwidth can be evaluated independently for each traffic category; the gains that are possible by integration are then considered. Furthermore, because of the properties of the *session* levels and *flows*, it is possible to overcome the complex problems of packet level self similarity.

9.2.4.1. *Real time traffic*

For such traffic, performance will be expressed in terms of loss rate or service delay. Our problem is therefore to evaluate the bandwidth necessary to ensure that the delays or losses to which packets are subjected are very low. The simple models we present below enable this objective to be achieved, and apply to most cases that arise in practice.

The bit rate generated by traffic and its characteristics (e.g. constant bit rate or variable bit rate) are first determined by the coding characteristics.

For example, we may have a flow consisting of 1 packet of 200 bytes every 20 ms for a voice source handled in G711, that is a constant bit rate of 80 kbit/s. (Note that voice coded at 64 kbit/s generates one byte every 125 µs, and that 160 are put in every 20 ms in a G711 packet intrinsically containing a header of 40 bytes.) In this case, the flow corresponds to a session.

On the other hand, if the voice is coded according to G729A plus VAD (*voice activity detection*), i.e. with silence suppression, we will have a variable bit rate of series of packets of 60 bytes, with the average bit rate now being less than 24 kbit/s. A session will be composed of flows corresponding to the active periods (i.e. when no silence).

If we consider an MPEG 2 video coding session, and assume image by image transmission, we would have for example the transmission of a flow of a variable volume of data in the order of 20 kbits to 40 kbits on average every 40 ms, i.e. bursts of 2 to 4 packets on average every 40 ms (based on the maximum size of the Ethernet frame which is 1500 bytes). The volume obviously depends very greatly on the type of coding (MPEG 1, MPEG 2, MPEG 4). Other packeting scenarios can of course also be considered.

The main coding characteristics of some fundamental types of real time services are set out below.

VoIP

G.711: 1 sample of 1 byte every 125 µs (μ *law*), grouped in packets every 20 ms. Packet size = 160 bytes (20 ms) + 40 bytes (*header*) = 200 bytes (*header* = 20 IPV4 bytes + 8 UDP bytes + 12 RTP bytes).

50 packets/s = 80 kbit/s per direction.

G.729A: 1 sample of 10 bytes every 10 ms, grouped in packets every 20 ms. Packet size: 20 bytes (20ms) + 40 bytes = 60 bytes.

50 packets /s = 24 kbit/s per direction.

G.729A + VAD (silence detection): bandwidth reduction in the order of 50%.

Videoconference, visiophony

H261, H263 coding (64-128-384 kbit/s, 30 to 10 images/s).

Resolution: CIF (*Common Intermediate Format*) and QCIF (*Quarter CIF*).

Inter-arrival times of video frames:

Table 9.6. *Visio coding and time inter-frames*

Resolution	**CIF**	**QCIF**
Codec		
H261	499 ms	214 ms
H263	340 ms	143 ms

On-demand or pay-per-view TV

MPEG: time compression (*key I, and delta images B, P*) and space compression (jpeg, DCT):

– MPEG 1 for video (TV, CD) peak rate up to 3.5 Mbit/s (typical value 1.5 Mbit/s);

– MPEG 2 for better quality, > 5 Mbit/s to 15 Mbit/s (prediction);

– MPEG 4: TV HD. Possibility of low bit rates, of 64–128 kbit/s, but also high bit rates. Use of VOP concept (*video object*).

In calculating the bandwidth necessary to multiplex this traffic, a distinction must be drawn between constant bit rates and variable bit rates.

Constant bit rate case

The dimensioning of the necessary resources can be simply carried out as follows. We first assume a call or session acceptance mechanism for each service class. Acceptance or rejection of the session will be performed simply by applying the Erlang formula, service type by service type, as the aim is to guarantee a given rejection probability per service type (and not a mean probability). For a service class of type i, bit rate d_i, activity (in erlangs) A_i, and for a rejection probability P_{Ri}, we have:

$$P_{Ri} = E(N_i, A_i), \tag{9-81}$$

which gives N_i, the maximum number of possible simultaneous sessions, i.e. the maximum bit rate to be handled, $N_i d_i$.

Because of the multiplexing properties of constant bit rates as previously studied (see the multirate concentrator/multiplexer model), the application of an M/G/1 type model gives a good approximation of the bandwidth necessary to handle this traffic (we are in the *cell/packet* zone), with an acceptable quality of service. In practice, a simple rule could be adopted such as compliance with a maximum utilization factor of 90%. This gives the necessary bit rate capacity:

$$C_c = (N_i d_i)/0{,}9. \tag{9-82}$$

Variable bit rate case

Real time variable bit rate traffic (here we will speak also of *stream* traffic) can be modelled by a source generating *bursts* of packets at a peak rate d_i. It is therefore an ON-OFF type source.

As previously, the maximum number of sessions N_i that can simultaneously coexist is determined, for an acceptable rejection probability, simply by applying the Erlang formula. For a service class i, with a peak rate d_i, with an activity (in erlang) of A_i (at session level) and for a rejection probability P_{Ri}, we have:

$$P_{Ri} = E(N_i, A_i) \tag{9-83}$$

which gives N_i, the maximum number of possible simultaneous sessions.

Let us first consider the case of flows of identical peak rate d. This model corresponds to a system such that bit rate d is imposed, and chosen to comply with sources with a higher peak rate (e.g. video).

For a relatively large number N of sessions constituting a flow of Poissonian arrivals, it is then possible to evaluate the number of bursts simultaneously in progress and thus the necessary capacity C_s simply by means of the Erlang or Poisson formula, as in the case of ON-OFF source multiplexing considered earlier. C_s being the total capacity and d the bit rate of the bursts, and if we call D the ratio C_s/d, the system behaves like a concentrator with D servers. If p is the probability for a source being in transmission phase (ON state), then:

$$Psat = E(D,\alpha)\text{, or with the Poisson law: } Psat = \sum_{D \le n}^{\infty} \frac{\alpha^n}{n!} e^{-\alpha} \qquad (9\text{-}84)$$

where $\alpha = Np$

(Note that it would also be possible to apply the Engset formula in the case of a limited number of sources.)

Let us now consider the case of sources with different bit rates. This corresponds to the case of a system in which the same peak rate is not imposed for all flows, in particular to try to take advantage of better multiplexing gains, as we have seen that this depends substantially on the equivalent number of servers D. Note that it is necessary to be able to distinguish between flows. Let us call d_i the bit rate or the capacity, required by a source of type i, and α_i its activity at *burst* level (proportion of time in ON state).

We thus return again of course to the multirate case. We can now apply the results previously established, and to simplify calculations, apply approximate formulas.

The x classes of traffic of bit rate d_i and activity α_i are replaced by a single class of traffic with an activity α/z and a single bit rate z, such that:

$$\alpha = \sum_{i=1}^{x} \alpha_i d_i \text{ and } z = \frac{\sum_{i=1}^{x} \alpha_i d_i^2}{\sum_{i=1}^{x} \alpha_i d_i}\text{, where } \alpha_i = N_i p_i$$

The equivalent model to the initial system is again simply a multiplexer of reduced capacity C_s/z, to which is offered a traffic of reduced intensity α / z, whose saturation probability is:

$$Psat \approx E\left(\frac{C_s}{z}, \frac{\alpha}{z}\right) \qquad (9\text{-}85)$$

Finally, the equivalent bandwidth approach will of course also be possible.

9.2.4.2. *Elastic traffic*

The performance of these traffics tends to be expressed in terms of processing duration and response time for a *flow*, rather than at packet level.

Our problem is therefore to determine the capacity C_e (in bandwidth) which is necessary to obtain a response time T that is acceptable at flow level.

We still assume there are a large number of independent sources, and that as a result the arrivals comply with a Poisson process. In this case, in view of the role of the flow management protocols such as TCP, the system can be modelled by a fluid system, such that the effective bit rate is immediately adjusted to the number of flows to be processed simultaneously, assuming that there is equitable sharing of the bandwidth between all the flows. Once again we are considering the *processor sharing* model. The important point to be stressed here is that it can be demonstrated that this model is insensitive to the distribution of the "length" of "tasks", in this case the size of the flows. We can thus overcome the difficulty of the great variability of flow level that characterises this type of traffic.

Let λ_e be the flow arrival rate. Let $\bar{x}$ be the mean volume (size) of the flows (in kbits for example, or product of source bit rate by flow duration). If C_e is the capacity dedicated to this type of traffic, we have:

$$\rho_e = \lambda_e \bar{x} / C_e,$$

The distribution of the number of flows simultaneously in progress is given by:

$$P(N = n) = \rho^n (1 - \rho_e) \qquad (9\text{-}86)$$

The average response time is:

$\bar{T}(x) = \dfrac{x}{C_e(1-\rho_e)}$, for a flow of size x, and

$$\bar{T} = \frac{\bar{x}}{C_e(1-\rho_e)}, \text{ for all flows.} \qquad (9\text{-}87)$$

Thus, by setting objectives on $\bar{T}$, we can determine very simply the necessary capacity C_e.

9.2.4.3. *Global capacity*

The global capacity in this model can be obtained either by a model without integration (i.e. a separate dimensioning for each type of flow), or by a model with integration (that is by globally taking account of all flows).

For the model without integration we thus have:

$$C = C_c + C_s + C_e \tag{9-88}$$

For the model with integration, we suggest:

$$C = Max(C_c + C_s; C_c + C_e + C_{sm}) \tag{9-89}$$

In this formula C_{sm} designates the mean total flow of stream traffic (real time variable). It may be considered that the bandwidth reserve, necessary to correctly handle the total real time variable bit rate and constant bit rate traffic, will be sufficient to handle the elastic traffic, provided of course that the bit rate thus required is greater than the total of the mean bit rates of all traffic. This is justified by the fact that elastic traffic has no strong real time constraint, and does not have priority. Guaranteeing the corresponding mean bit rate is therefore sufficient. Checks are however necessary to ensure that variations in real time traffic strongly penalise elastic traffic only on exceptional occasions (i.e. with a very low probability), so as not to create periods of instability (a small reserve may also be taken against C_e).

We will see an example of the application of these results in Chapter 10.

9.2.5. *Series queue model (M/M...M/1)*

This is the name given to a succession of servers with a queue in series. This model may correspond to certain realisations and routing strategies in signalling networks or more generally in data networks. We will only deal here with the simple (and extreme) case of a series of identical servers that only process a single message flow (Poissonian at input), and whose processing time is directly proportional to the length of packets (or messages), a length that is governed by an exponential distribution. Under these conditions, the service time varies with each message but is the same whatever the server for a given message. We will however also show that this model can also be applied, in certain conditions, to the general case of a meshed network and in particular to the case in which converging flows are added.

Figure 9.19 shows the network architecture considered.

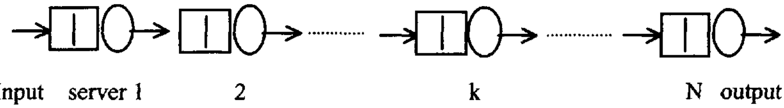


Figure 9.19. *Network of queues in series*

The basic problem is as follows: messages grouped at a node k ($k \geq 2$), remain grouped in the following nodes and an agglutination phenomenon occurs which grows with k, leading to the formation of increasingly large *bursts* as they pass through the network. This phenomenon is intuitively obvious, but is difficult to quantify.

Let us make the phenomenon explicit. There is something specific about nodes 1 and 2: node 1 is of course an ordinary server of the M/M/1 type, and agglutinations may occur in the queue. However, these groups may be broken at node 2. Let us assume that a message of index m_1 is in service at node 1 and is followed by a message m_1+1 that is waiting. At the end of service of m_1 at node, the service of m_1 at node 2 begins and the service of m_1+1 at node 1 begins. If m_1+1 is longer than m_1 then the end of service of m_1 at node 2 takes place before that of m_1+1 at node 1, and the *burst* is broken.

On the other hand, if the message m_1+1 is shorter than message m_1 then the messages remain grouped at node 2. It is easy then to check that at the output of a node k, if $k \geq 2$, for any burst, we have:

– all messages belonging to a burst have a length shorter than or equal to that of the leader (the first message of the burst);

– all bursts thus formed are conserved at the output of all the following nodes: the messages that constitute it remain definitively attached.

Without going into detail, we can thus intuitively write the following properties that are easy to verify:

– all messages belonging to a burst at node k have a sojourn time at this node equal to the service duration of their leader;

– all bursts whose leader is shorter than the leader of the previous burst will be (after some time) agglutinated with the previous burst (the more rapid messages finally catch up with the slower messages).

We can thus deduce that, if we call (n, k) the index of the burst of number n at node k, then the necessary and sufficient conditions such that *burst* $(n+1, k-1)$ catches up with *burst* $(n, k-1)$ at node k are:

– its arrival date (the date of arrival of its leader) at node $k-1$, which is necessarily later than that of the previous message, is in an interval x that is shorter than the service duration of the leader of the previous burst;

– the service duration of its leader is shorter than or equal to that of the leader of the previous burst.

All these results can be found by carefully considering Figure 9.19, which shows the successive arrivals of the messages with the following notations:

$B(n,k)$: *burst n* at the output of node k,

$b_k^n(i)$: message i of *burst* $B(n, k)$,

$b_k^n(1)$: first message of *burst* $B(n, k)$, called the leader,

$b_k^n(j)$: last message of *burst* $B(n, k)$,

t_0 : date of start of service of *burst n* at node $k-1$, *burst* $B(n, k-1)$,

t_1 : date of end of service of *burst n* at node $k-1$,

t_2 : agglutination instant at node k of bursts n and $n+1$ of node $k-1$.

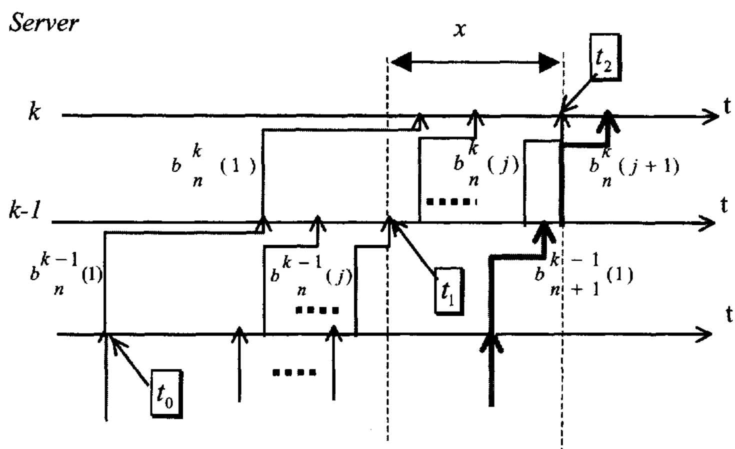


Figure 9.20. *Burst formation*

NOTE: For the sake of simplicity, only the first message (in bold) of burst $n+1$ at node $k-1$ is shown, noted $b_{n+1}^{k-1}(1)$.

Now that these properties have been explained, we need to quantify the agglutination phenomenon.

An initial approximate result can be easily obtained: this is the mean value of the busy period (in number of messages) at each node, i.e. the mean length of the bursts.

Based on the previous analysis and in particular the agglutination conditions, we know that agglutination occurs at each stage for a time interval x which is equal to the service duration of the leader, which is also the sojourn time of each of the burst messages at each stage. At input into the network, because of the Poissonian arrivals, the mean number of messages agglutinated, behind a leader, is of course:

$$\bar{n} = \frac{\rho}{1-\rho}$$

After a long time $k\bar{x}$, all the groups of messages formed at the input between two long messages, i.e. during successive *sub-busy periods* behind shorter messages, will be agglutinated at stage k (behind a long message), and their average number will therefore be approximately:

$$\bar{n} = k\frac{\rho}{1-\rho}$$

It is therefore possible to write the following approximate relation. The average length of bursts $n(k)$, in number of messages at each node k ($k > 2$), is such that:

$$n(k) \underset{k\to\infty}{\cong} k\frac{\rho}{1-\rho} \tag{9-90}$$

Figure 9.21 shows that this result matches up well with the simulation results.

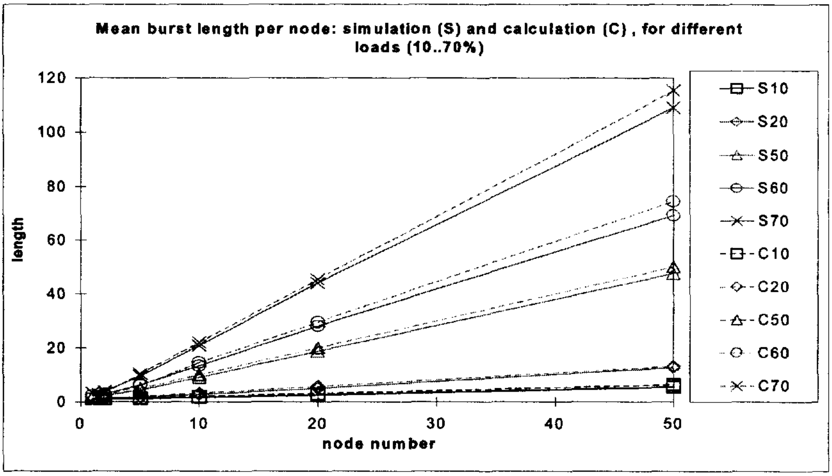


Figure 9.21. *Mean burst length per node*

Let us now study the wait at each node. To resolve this complex problem we make use of the Pollaczek method. We will mainly follow here Le Gall's approach [LEG 94a].

According to the general method, let us first establish the stochastic relationship. We may write $N = m + 1$ (the network consists of $m + 1$ nodes in series). Let us call, at node k, for the nth message:

w_n^k : waiting time,

T_n^k : service time,

s_n^k: sojourn time ($s_n^k = w_n^k + T_n^k$),

Y_{n-1}^k : inter-arrival time (between messages $(n-1)$ and n),

and let us introduce:

$T_n(2,k) = T_n^2 ... + T_n^k$: global service time from node 2 onwards,

$S_n(2,k) = s_n^2 \cdots + s_n^k$: global sojourn time from node 2 onwards.

Let us first consider the case of two servers in series. Let us write:

$$S_n(1,2) = s_n^1 + s_n^2 \tag{9-91}$$

We can then write:

$$S_n(1,2) = Max\left[s_n^1 + T_n^2, S_{n-1}(1,2) + T_n^2 - Y_{n-1}^1\right] \tag{9-92}$$

This relation simply expresses whether or not there is a wait at node 2.

Let us now term e_n^k: the free time period of server k (k = 1.2), if it exists, between messages (n–1) and n. We have:

$$T_n^1 + W_n^2 = Max\left[T_n^1, s_{n-1}^2 - e_n^1\right] \tag{9-93}$$

This relation simply expresses first whether or not there is a wait for message n at node 2 ($Max = \left[T_n^1\right]$ if there is no wait), and second whether the service of message n–1 at node 1 is finished ($e_n^1 \neq 0$), or not before the arrival of message n.

These relations are easy to verify by referring to the type of chart presented earlier (see Figure 9.20).

As the service durations are identical at each stage, we have:

$$T_n^1 + W_n^2 = T_n^2 + W_n^2 = s_n^2 \tag{9-94}$$

and thus:

$$s_n^2 = Max\left[T_n^1, s_{n-1}^2 - e_n^1\right] \tag{9-95}$$

This is the basic stochastic relation from which we will begin to study the general case.

Our network thus consists of m + 1 servers in series. It is possible to verify by recurrence, by referring to the previously established properties, that the previous relation can be generalised as follows:

$$S_n(2,m+1) = Max\left[mT_n^1, S_{n-1}(2,m+1) - e_n^1\right] \tag{9-96}$$

This is the fundamental stochastic relation of our network with $m+1$ nodes in series. Thus, with a simple change of variable $t \to \frac{t}{m}$, we return to the stochastic relation of the network with two nodes, and we can thus apply the results (at the second server) of the network with two nodes to our system.

Furthermore, if we call $\overline{S(m+1)} = \frac{S(2,m+1)}{m}$, the mean sojourn time per node (from node 2 onwards), we have the following approximate recurrence relation:

$$\overline{S(m+1)} = m \ \overline{S(m)} - (m-1) \ \overline{S(m-1)} \tag{9-97}$$

and for the wait at node k ($k = 2, m+1$), taking the mean service duration as the time unit:

$$\overline{W(k)} = (k-1) \ \overline{S(k-1)} - (k-2) \ \overline{S(k-2)} - 1 \tag{9-98}$$

Calculation of sojourn time in the case of two nodes

In the case of two servers in series, with Poissonian arrivals (of rate λ) at the first server, and an identical service law at all servers $F(t)$, Boxma [BOX 79] has demonstrated the following result for the sojourn time at the second server:

$$S(t) = L(t)J(t)F(t) \tag{9-99}$$

where:

$$L(t) = (1-\rho)\frac{z(t)}{\lambda[1-F(t)]} \text{ and } J(t) = \exp\left(-\int_t^\infty z(v)dv\right) \tag{9-100}$$

where $z(t)$ is the single real and positive root of the equation in z:

$$z - \lambda\left[1 - \int_0^t \exp(-zu)dF(u)\right] = 0, \tag{9-101}$$

$z(t)$ being small, we make the following approximation:

$\exp(-zu) \approx 1 - zu$.

And we thus obtain:

$$z(t) = \frac{\lambda[1 - F(t)]}{1 - \alpha(t)} \text{ where } \alpha(t) = \lambda \int_0^t u dF(u) \text{ and } \alpha(\infty) = \rho$$

and thus also $L(t) = \dfrac{1 - \rho}{1 - \alpha(t)}$

Furthermore, by integrating by parts $\alpha(t) = \lambda \int_0^t u dF(u)$, we have:

$$\alpha(t) = \lambda \int_0^t u dF(u) = -\lambda t[1 - F(t)] + \rho \frac{1}{E(T)} \int_0^t [1 - F(u)] du,$$

where $\rho = \lambda E(T) < 1$.

Given that $\dfrac{1 - F(u)}{E(T)}$ is simply the expression of the density of the residual service time (see Chapter 7) and thus:

$$\int_0^t \frac{[1 - F(u)]}{E(T)} du = F(t)$$

Hence, where t is large, $t[1 - F(t)] \to 0$ and:

$$\alpha(t) \approx \rho F(t)$$

And finally:

$$L(t) = \frac{1 - \rho}{1 - \rho F(t)}, z(t) = \frac{\lambda[1 - F(t)]}{1 - \rho F(t)}$$

$$\text{hence } J(t) \approx \frac{1 - \rho}{1 - \rho F(t)} \approx L(t) \tag{9-102}$$

Application to the network of m+1 servers in series

In view of the stochastic relation we can treat each stage as stage 2 with the change in variable $t \to \dfrac{t}{m}$. The variable t now takes the value $\dfrac{t}{m}$.

Hence:

$$J(t) \to \exp\left(-\int_t^\infty z(\frac{v}{m})dv \right) = \exp\left(-m \int_{t/m}^\infty z(v)dv \right) = \left[J(t)\right]^m \tag{9-103}$$

We thus have: $S(t;m) = L(t)\left[J(t)\right]^m F(t)$ and in view of the previous result $J(t) \approx L(t)$, we have:

$$S(t;m) \cong \left(\frac{1-\rho}{1-\rho\ F(t)} \right)^{m+1} F(t) \tag{9-104}$$

The *mean* sojourn time by node (from node 2 to node to (m+1)), is thus:

$$\overline{S(m)} = \int_0^\infty [1 - S(t;m)]\ dt\ ,\ (\overline{S(m)} = \int_0^\infty t\ \frac{dS(u;m)}{du}(t)\ dt) \tag{9-105}$$

And the total sojourn time of node 2 up to node m+1 is therefore:

$$m\ \overline{S(m)} \tag{9-106}$$

Let us evaluate this expression in the case of an exponential service duration distribution.

$$\text{In this case, } \forall t \geq 0\ \ F(t) = 1 - e^{-t} \tag{9-107}$$

$$\text{Hence: } \overline{S(p)} = \int_0^\infty [1 - \left(\frac{1}{1 + [\rho/(1-\rho)]\ e^{-t}} \right)^{p+1} (1 - e^{-t})]\ dt \tag{9-108}$$

Let us write: $u = e^{-t}$, which gives us:

$$\overline{S(m)} = \int_0^1 [\frac{1}{u} - \left(\frac{1}{1+[\rho/(1-\rho)]u}\right)^{m+1} (\frac{1-u}{u})] \; du$$

Let us write $v = \dfrac{\rho}{1-\rho} u$, which gives us:

$$\overline{S(m)} = \int_0^{\frac{\rho}{1-\rho}} [\frac{1}{v} - \left(\frac{1}{1+v}\right)^{m+1} (\frac{1-[(1-\rho)/\rho]v}{v})] \; dv$$

$$\overline{S(m)} = \int_0^{\frac{\rho}{1-\rho}} \frac{1}{v}\left(1 - \left(\frac{1}{1+v}\right)^{m+1}\right) dv \; + \frac{1-\rho}{\rho} \int_0^{\frac{\rho}{1-\rho}} \left(\frac{1}{1+v}\right)^{m+1} dv$$

Given that $\int_0^{\frac{\rho}{1-\rho}} \left(\frac{1}{1+v}\right)^{m+1} dv = \dfrac{1}{m}[1-(1-\rho)^m]$

And $1 - \dfrac{1}{(1+v)^{m+1}} = \sum_{n=0}^{m}\left[\left(\frac{1}{1+v}\right)^{n} - \left(\frac{1}{1+v}\right)^{n+1}\right] = \sum_{n=0}^{m} \dfrac{v}{(1+v)^{n+1}}$

Hence $\overline{S(m)} = \sum_{n=0}^{m} \int_0^{\frac{\rho}{1-\rho}} \dfrac{1}{(1+v)^{n+1}} \; dv \; + \dfrac{1}{m}\dfrac{1-\rho}{\rho}[1-(1-\rho)^m]$

And finally:

$$\overline{S(m)} = \sum_{n=1}^{m} \frac{1}{n}[1-(1-\rho)^n] \; + \frac{1}{m}\frac{1-\rho}{\rho}[1-(1-\rho)^m] \; + \ln(\frac{1}{1-\rho}) \qquad (9\text{-}109)$$

Finally by using the recurrence relation:

$$\overline{W(k)} = (k-1) \; \overline{S(k-1)} - (k-2) \; \overline{S(k-2)} - 1$$

we obtain the mean wait at a node k, $\forall \; k \geq 2$:

$$\overline{W(k)} = \sum_{n=1}^{k-2} \frac{1}{n}[1-(1-\rho)^n] \; + \ln(\frac{1}{1-\rho}) \qquad (9\text{-}110)$$

For sufficiently large k values, we can write:

$$-\sum_{n=1}^{k-2}\frac{(1-\rho)^n}{n} \cong \ln(\rho)\text{, and also, } \sum_{n=1}^{k-2}\frac{1}{n} \cong \ln(k-2)+\gamma+\frac{1}{2(k-2)}$$

where γ is the Euler constant ($\gamma \cong 0.577$)

and thus:

$$\overline{W(k)} \cong \ln[(k-2)\frac{\rho}{1-\rho}]+\gamma+\frac{1}{2(k-2)} \tag{9-111}$$

An asymptotic value of the wait is thus:

$$\overline{W(k)} \underset{k\to+\infty}{\cong} \ln[(k-2)\frac{\rho}{1-\rho}]+\gamma \tag{9-112}$$

or in the case of very high values for k:

$$\overline{W(k)} \underset{k\to+\infty}{\cong} \ln(k\frac{\rho}{1-\rho})+\gamma \tag{9-113}$$

The result is remarkably simple in view of the relative complexity of the calculations.

Figure 9.22 shows the very good coherence between the model and the simulation.

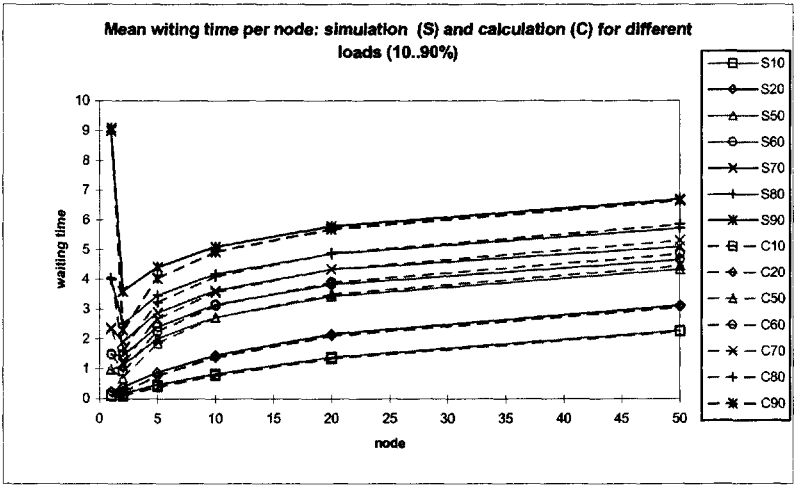


Figure 9.22. *Mean waiting time per node*

Generalisation for the case of convergent intermediate flows

This time we consider the possibility of additional flows converging on to the same server.

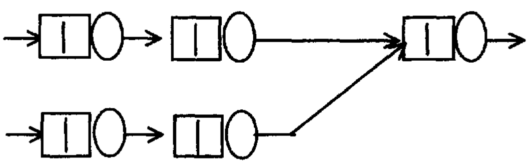


Figure 9.23. *Converging queues in a meshed network*

It is then possible to demonstrate [LEG 94a, LEG 94b] that this network can be reduced to an equivalent network of queues in series by changing the number of servers m to an appropriate value m_0. Similarly, the transversal flows that have existed upstream can be ignored. We will not indicate these results in detail. But it is important to note, on one hand, that these properties enable treatment of the general case of meshed networks, and on the other hand that agglutination phenomena may remain noticeable in very general cases (provided of course that the physical realisation gives rise to this situation).

Conclusion concerning the series queues model

An expert in queueing theory will note the necessity of making use in certain cases of global end-to-end modelling of the system. The behaviour at each node, and particularly the arrivals, depends on the behaviour at the previous nodes. As the messages can be grouped because of the agglutination phenomenon, they become indiscernible and as such they cannot be processed individually. Strictly speaking, the habitual concept of local traffic source is no longer pertinent. This explains the relative complexity of the problem.

The performance expert will primarily take note of the strong impact of this phenomenon of queues in series on the busy period, and on the local wait value, when the conditions of realisation are met. He will take care to ensure above all that these realisations should not lead to uncontrolled agglutination phenomena. At transport level, techniques such as flow mixing, the limitation of the maximum length of messages, the use of high bit rate links, etc. will make it possible to avoid this type of phenomenon. At the system control level, the chaining of excessively long tasks must be avoided, as this creates the "hiccup" phenomenon which has an avalanche effect on processing queues. These problems will then need to be handled on the same basis as exceptional situations such as simultaneous arrivals (on failure and switchover, on game synchronisation pulses, etc.). We are here entering the field of the defence of systems and networks in extreme situations, and the analysis must then take as much account of the deterministic approach as of the probabilistic approach.

9.2.6. *Switching matrix*

We will here study the case of an ATM switching matrix which switches packets of constant length. The modelling of this system, although specific, will enable us to deduce relatively general results.

Let us consider a matrix with output queues, switching M input links to M output links, organised as indicated in Figure 9.24.

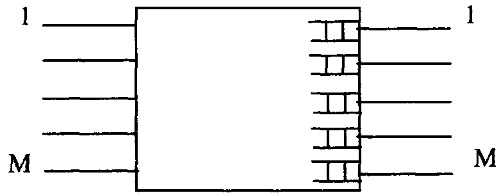


Figure 9.24. *Switching matrix with output queues*

Let p be the probability of occupation of a cell time of an input link. We therefore assume a Bernoulli process for occupation of the cell times on this link, which relatively well represents the actual behaviour of a link on which various traffic currents have been multiplexed.

We therefore also assume perfect equiprobability of traffic routing, which means that each cell has a probability of $1/M$ of going towards one of the outputs. The probability for an output link of having a cell time occupied because of an input is therefore p/M.

Note also that the output link load, or the cell time load, is $= p$. Let us now calculate the number of cells in the system, awaiting sending on an output link. For this purpose, let us calculate the number of arrivals on this link over a cell time.

The probability of i cells arriving simultaneously at a given output at a given instant is:

$$P(x_i = i) = \binom{M}{i}(p/M)^i(1-p/M)^{M-i} \qquad (9\text{-}114)$$

The characteristic function of $P(x)$ is then:

$$\phi(z) = \sum_k p_k e^{zx_k}$$

or in terms of the generating function $F(z) = \sum_k p_k z^k$.

And therefore (see also Chapter 4):

$$X(z) = \sum_{i=0}^{M} P(x_i) z^i = \sum_{i=0}^{M} z^i \binom{M}{i} (p/M)^i (1 - p/M)^{M-i} =$$
$$(1 - p/M + zp/M)^M \tag{9-115}$$

We can now apply the Pollaczek relation for the number of customers (cells in this case) in the system (see Chapter 7):

$$N(z) = X(z) \frac{(1-\rho)(1-z)}{X(z) - z} \tag{9-116}$$

Hence:

$$N(z) = (1-\rho)) \frac{(1 - p/M + zp/M)^M (1-z)}{(1 - p/M + zp/M)^M - z} \tag{9-117}$$

The moments can be deduced by differentiation relative to z, and for $z = 1$ (see Chapter 3).

$$E(N) = N'(1), \text{ and } \sigma^2(N) = N''(1) + N'(1) - N'^2(1)$$

$$E(N) = \frac{\rho}{1-\rho}(1 - \frac{\rho}{2} - \frac{\rho}{2M}) \tag{9-118}$$

$$\sigma^2(N) = \rho(1-\rho) + \frac{M(M-1)}{M^2} \frac{\rho^2(3-2\rho)}{2(1-\rho)} + \left(\frac{M(M-1)}{M^2} \frac{\rho^2}{2(1-\rho)} \right)^2 +$$
$$\frac{M(M-1)(M-2)}{M^3} \frac{\rho^3}{3(1-\rho)}$$

At this stage, it is interesting to compare this with the classical result of the M/D/1 queue.

We have (see Chapter 7):

$$E(N) = \frac{1}{2} \frac{\rho^2}{1-\rho} + \rho = \frac{\rho}{1-\rho}(1 - \frac{\rho}{2})$$

and:

$$\sigma^2(N) = \rho(1-\rho) + \frac{\rho^2(3-2\rho)}{2(1-\rho)} + \left(\frac{\rho^2}{2(1-\rho)}\right)^2 + \frac{\rho^3}{3(1-\rho)}$$

This shows that as soon as M is large enough (> 10), even without really being very large, our system is equivalent to an M/D/1 queue.

Finally let us note the expression of mean waiting time, obtained by applying the Little formula (the arrival rate is $1/p$):

$$E(W) = (\frac{M-1}{M})\frac{1}{2}\frac{\rho}{1-\rho}\tau \text{ , where } \tau \text{ is the cell time} \tag{9-119}$$

Once again we note the similarity with the M/D/1 queue.

This result is of course obvious when M is tending towards infinity. What is remarkable is the rapidity of the convergence. The approximation to the M/D/1 queue can thus be applied in most real-life cases. And more generally, in the case of packets of variable length, approximation to an M/G/1 queue will be valid.

9.2.7. *Switching network*

The switching network, or connection network, is an equipment that enables the establishment of a physical path for the transport of information between a data input and a data output for the duration of a call, or session, or any communication phase whose duration and integrity justify the establishment of such a path. It may of course be a matter of establishing virtual paths (circuits) between virtual input and output circuits, supported by physical links. A switching network necessarily includes several subsystems, usually called *matrices*, arranged in stages and interconnected. The purpose of this assembly is to construct networks with a very high capacity using elements of limited capacity. Figure 9.25 illustrates the principle of a 3-stage network in schematic form. The end stages include m matrices with n inputs and k outputs, and the central stage includes k matrices with m inputs and m outputs. If the capacity of each input or output link is c (e.g. c channels at 64 kbit/s or c Mbit/s) the network is said to have a capacity of nxmxc. But this capacity of course depends on the interconnection capacity of the end matrices by the central stage, and thus on the value of k, and on the capacity of the internal links.

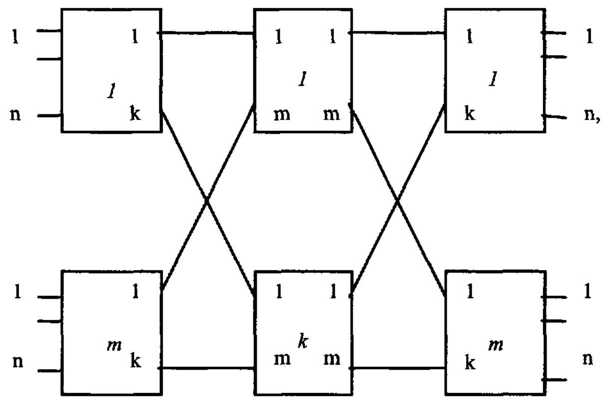


Figure 9.25. *Three stages switching network*

Very generally speaking, a network is characterised in traffic terms by three main parameters:

– its blocking, i.e. the probability that no free path will be found between the free input and the free output considered (point-to-point blocking). Blocking will occur for example if no physical path is found with the minimum bandwidth required by the communication;

– its transfer delay, i.e. the characteristics (mean, dispersion, quantiles, etc.) of the total of the waiting times and service times to which the elementary information (the packet for example) is subjected at each switching stage. In a synchronous network, such as a conventional telephone network, there will be no waiting time except for some fixed resynchronisation delays. On the other hand, in an asynchronous network (ATM or packets), each stage may include queues and thus cause variable delays. (Note here the maximum delays given by Q.551, see Chapter 2.);

– the probability of information loss. In a packet or ATM connection network, queue overflow will result in information loss.

9.2.7.1. *Blocking calculation: Lee-Le Gall model*

This model is very general, and encompasses calculation methods such as those of Lee and Jacobeus.

Let us return to our example of three-stage network architecture. For the sake of simplicity, we assume that all the links have an identical capacity *c*, i.e. we allocate a capacity of *c places*, or virtual circuits, or *resources*, per link represented. We call links A and B the links between the input stage and the central stage on the one hand, and between the central stage and the output stage on the other hand.

To facilitate the reasoning, we first show the possible paths for a communication, between a given input and a given output, using a so-called *Lee graph* (see Figure 9.26). We first establish what Le Gall has called the "fundamental function" of the network [LEG 62], assuming a Bernoulli distribution for the occupation of places and paths at each stage. It is generally assumed that the stages are independent of each other.

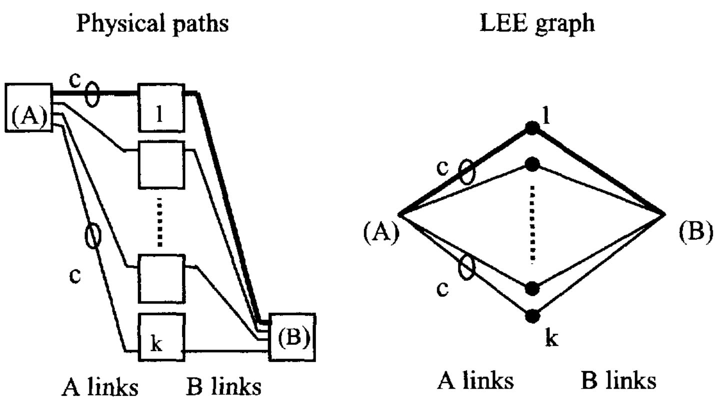


Figure 9.26. *Graph of paths in a three stages switching network*

Thus if we call p_1 the probability that a resource in a link A is occupied, and p_2 the probability that a resource in a link B is occupied, then the probability B that all the possible paths between an input of A and an output of B are occupied, assuming that the paths are independent, is:

$$B = (1-(1-p_1^{\,c})(1-p_2^{\,c}))^k \tag{9-120}$$

A path is in fact only free if there is a free place on link A and a free place on a corresponding link B. Blocking occurs if there is no free path on any link.

To simplify writing, let us now write $\pi_1 = p_1^{\,c}$ the probability that a link A is occupied (the c places are occupied), and $\pi_2 = p_2^{\,c}$ the probability that a link B is occupied. The previous relation is now written:

$$B = (1-(1-\pi_1)(1-\pi_2))^k = (\pi_1 + \pi_2(1-\pi_1))^k \tag{9-121}$$

And by applying the Pascal binomial law:

$$(a+b)^k = \sum_{i=0}^{k} \binom{k}{i} a^i \, b^{k-i}$$

we obtain:

$$B = \sum_{i=0}^{k} \binom{k}{i} \pi_1^{\,i} \pi_2^{\,k-i} (1-\pi_1)^{k-i} = \sum_{i=0}^{k} \sum_{j=0}^{k-i} (-1)^j \pi_1^{\,i+j} \, \pi_2^{\,k-i} \binom{k}{i} \binom{k-i}{j}$$

This is the *fundamental function* of the network, which is written as follows:

$$B = \sum_{i=0}^{k} \sum_{j=0}^{k-i} (-1)^j \pi_1^{\,i+j} \, \pi_2^{\,k-i} \quad \frac{k!}{i!\,j!(k-i-j)!} \tag{9-122}$$

This clearly indicates three terms. The two π terms independently express the probabilities of occupation of the places and paths at each stage based on the Bernoulli assumption. And the term with the factorials, independent of the occupation probabilities, reflects the impact of the network meshing type. The model can be generalised to all meshed structures, obviously with analyses of greater or lesser complexity, but which remain in the field of counting and combinational analysis.

In a second stage, we do away with the binomial hypothesis.

Clearly the blocking expression in the previous form is not particularly useful based on the binomial distribution only, and the $(1 - p)$ type expression such as (9-120) would fully suffice in this case. In practice, we will have to take other distributions, first depending on whether the edge matrices concentrate the traffic or not, and secondly depending on the free path seek mode (e.g. equally traffic distribution, sequential hunting..), which result in a greater or lesser degree of interdependence between the resources. However, in most cases we will find necessary to make simplifying assumptions.

Initially, let us abandon the independence assumption concerning the independence of the links of the same matrix. In fact, the traffic from a matrix A is offered globally to all the k outgoing links and thus the probability of occupation of a given link is not independent of the state of other links. Thanks to the previous formulation, by replacing $\pi_1^{\,i+j}$ by H_{i+j} and $\pi_2^{\,k-i}$ by H_{k-i}, we can write:

$$B = \sum_{i=0}^{k} \sum_{j=0}^{k-i} (-1)^j H_{i+j} \cdot H_{k-i} \quad \frac{k!}{i!\,j!(k-i-j)!} \tag{9-123}$$

H_j represents the probability of having j links out of k occupied, whatever the state of the remaining k-j links, that is to say, the probability of having $j \times$ c places occupied out of c $\times$ k places. In fact, defining whether a link is occupied is equivalent to defining that the c given places that constitute it are occupied.

Secondly, we must express this probability H. As explained previously, the distribution to be used may vary depending on the nature of the traffic offered and the mode of network operation. Let us first calculate H_r, the probability of occupying r given places out of N, on the often verified hypothesis of place occupation equiprobability.

Let x be the total number of places occupied on the N, and let $r \leq x$ the places with which we are concerned. The number of ways of having the r places given amongst the x is the number of ways of placing the remaining x–r places in the x, that is $\binom{N-r}{x-r}$. And the number of ways of occupying x places amongst the N is $\binom{N}{x}$.

The probability of having r specified places occupied is thus:

$$\frac{\binom{N-r}{x-r}}{\binom{N}{x}}. \tag{9-124}$$

Palm Jacobeus formula

A rather pessimistic assumption consists of taking the Erlang law as the law of distribution for the occupation of x places out of all the N places, with A being the offered traffic. We then obtain by summating all the possible cases:

$$H_r = \frac{\sum_{x=r}^{N} \frac{A^x}{x!} \binom{N-r}{x-r} \Big/ \binom{N}{x}}{\sum_{n=0}^{N} \frac{A^n}{n!}}, \text{ also written } H_r = \frac{\frac{A^N}{N!}}{\sum_{n=0}^{N} \frac{A^n}{n!}} \frac{\sum_{x=r}^{N} \frac{A^{x-r}}{(x-r)!}}{\frac{A^{N-r}}{(N-r)!}}$$

And thus:

$$H_r = \frac{E(N,A)}{E(N-r,A)} \tag{9-125}$$

which is the well-known Palm-Jacobeus formula.

Extension to Engset

We can easily obtain, in the same way, a similar expression in the Engset case. This exercise can be carried out by the reader, who will verify that we then obtain:

$$H_r = \frac{\mathrm{E}(N,S,A)}{\mathrm{E}(N-r,S-r,A)} \tag{9-126}$$

where $\mathrm{E}(N,S,A)$ is the Engset formula for N servers, S sources and traffic A.

This expression can be used in the case of traffic from a limited number of sources.

Blocking

In our case, assuming Erlang traffic, the probability of having j links occupied is that of having $r = j \times c$ places occupied amongst $k \times c$, and therefore:

$$H_j = \frac{E(ck,A)}{E(c(k-j),A)}, \text{ hence}$$

$$H_{i+j} = \frac{E(ck,A)}{E(c(k-i-j),A)}, \quad H_{k-i} = \frac{E(ck,A)}{E(ci,A)},$$

assuming that traffic is identical in both stages.

The network blocking probability is therefore:

$$B = \sum_{i=0}^{k} \sum_{j=0}^{k-i} (-1)^j \frac{E(ck,A)}{E(c(k-i-j),A)} \frac{E(ck,A)}{E(ci,A)} \frac{k!}{i!j!(k-i-j)!} \tag{9-127}$$

Table 9.7 gives some examples of numerical results.

Table 9.7. *Blocking as a function of the capacity and the number of links between stages*

k \ c	1	5	10	25	50	100	200
4	A=0.235 B=4.79 10^{-4}	A=7.771 B=2.44 10^{-4}	A=21.4 B=2.2 10^{-4}	A=69.3 B=2.07 10^{-4}	A=156.2 B=2.029 10^{-4}	A=338.4 B=2.011 10^{-4}	A=714.3 B=2.010 10^{-4}
8	A=1.42 B=7.51 10^{-4}	A=21.4 B=2.48 10^{-4}	A=52.7 B=2.17 10^{-4}	A=156.2 B=2.058 10^{-4}	A=338.4 B=2.023 10^{-4}	A=714.3 B=2.016 10^{-4}	A=1481.7 B=2.010 10^{-4}
16	A=5.34 B=1.17 10^{-3}	A=52.7 B=2.76 10^{-4}	A=120.9 B=2.36 10^{-4}	A=338.4 B=2.11 10^{-4}	A=714.3 B=2.054 10^{-4}	A=1481.7 B=2.028 10^{-4}	A=3037.8 B=2.019 10^{-4}
32	A=15.60 B=1.68 10^{-3}	A=120.9 B=3.45 10^{-4}	A=264.75 B=2.68 10^{-4}	A=714.3 B=2.273 10^{-4}	A=1481.7 B=2.138 10^{-4}	A=3037.8 B=2.074 10^{-4}	

It is easy to verify that if $k \times c$ is large compared with k, the blocking is very well approximated by:

$$B = 2E(ck, A) - E(ck, A)^2 \qquad (9\text{-}128)$$

This can easily be explained by the fact that the most frequent blocking cases are:

– either k links are totally occupied in the second stage, whatever the state of occupation in the first stage, which corresponds to $i = 0$ and $j = 0$,

– or k links are occupied in the first stage, whatever the state of occupation in the second stage, which corresponds to $i = k$ and $j = 0$.

For each of these cases, the blocking probability is $E(ck, A)$.

This practical result is important as it clearly shows the preponderant influence of c, capacity (or bit rate) of the internal links, relative to meshing.

To complete this model, let us consider the special and extreme case of the same network where c = 1. This simple case can be directly studied on the graph.

As previously, blocking occurs if the free places amongst the k possible places in each stage do not match up. If we call i the number of occupied places in stage A and j the number of occupied places in stage B, such that $i + j \geq k$, the blocking probability is:

$$\frac{\binom{i}{k-j}}{\binom{k}{j}}.$$

This is the number of ways of placing the $k - j$ free places in the second stage in front of the i occupied places in the first stage, divided by the number of ways of placing the j amongst the k. And so, still assuming the Erlang law, and summating all the possible cases, we obtain:

$$B = \sum_{i=0}^{k} \frac{A^i / i!}{\sum_{x=0}^{k} A^x / x!} \sum_{j=k-i}^{k} \frac{A^j / j!}{\sum_{x=0}^{k} A^x / x!} \frac{\binom{i}{k-j}}{\binom{k}{j}}$$

which can also be written:

$$B = \frac{1}{\left(\sum_{x=0}^{k} A^x / x!\right)^2} \sum_{i=0}^{k} A^i / i! \sum_{j=k-i}^{k} A^j / j! \frac{\binom{i}{k-j}}{\binom{k}{j}}$$

and if we introduce $u = j - (k - i)$, we obtain:

$$B = \frac{1}{\left(\sum_{x=0}^{k} \frac{A^x}{x!}\right)^2} \sum_{i=0}^{k} \frac{A^i}{i!} \sum_{u=0}^{i} \frac{A^{u+k-i}}{(u+k-i)!} \frac{\binom{i}{i-u}}{\binom{k}{u+k-i}} = \frac{1}{\left(\sum_{x=0}^{k} \frac{A^x}{x!}\right)^2} \sum_{i=0}^{k} \frac{A^i}{i!} \sum_{u=0}^{i} \frac{A^{u+k-i}}{u!} \frac{i!}{k!}$$

$$B = \frac{\dfrac{A^k}{k!}}{\left(\displaystyle\sum_{x=0}^{k} \frac{A^x}{x!}\right)^2} \sum_{i=0}^{k} \sum_{u=0}^{i} \frac{A^u}{u!} = E(k,A) \frac{1}{\displaystyle\sum_{x=0}^{k} \frac{A^x}{x!}} \sum_{i=0}^{k} \sum_{u=0}^{i} \frac{A^u}{u!} =$$

$$E(k,A) \frac{1}{\displaystyle\sum_{x=0}^{k} \frac{A^x}{x!}} (1 + (1+A) + (1 + A + \frac{A^2}{2!}) + \cdots)$$

Hence:

$$B \approx E(k,A)(1 + k - A) \tag{9-129}$$

This result is remarkable, because of its simplicity (after a few developments however, which explains the usefulness of the Lee-Le Gall method for more complex cases), but above all because it again demonstrates the relative influence of k and c. As there are not many possibilities for each link (here there is only one, $c = 1$), a very high value of k is required to obtain low blocking with good efficiency.

9.2.7.2. *Multi-bit rate network case*

The analysis we have presented is very general, and applies as already explained to TDM (*time division multiplexing*) networks, ATM (*asynchronous transfer mode*) networks, or packet networks, if the connections are established on the basis of a virtual circuit and at an identical bit rate. The blocking evaluation in the case of the mixing of connections of different bit rates is far more complex, but can be simplified and brought into compliance with the previous case by some simple considerations. In most cases the bit rates to be transported remain low compared with the bit rate of the links. This is in fact a prerequisite to obtain significant gains in multiplexing. In this case, we can apply with sufficient precision the *peakedness factor* model on each of the links, as presented in the multi-bit rate concentrator and multiplexer cases, and thus arrive at the "reduced" network model. The *reduced network* consists of the initial network but with the capacity of each link divided by z, the *peakedness factor* value. For example, the previous network path graph thus becomes that indicated in Figure 9.27.

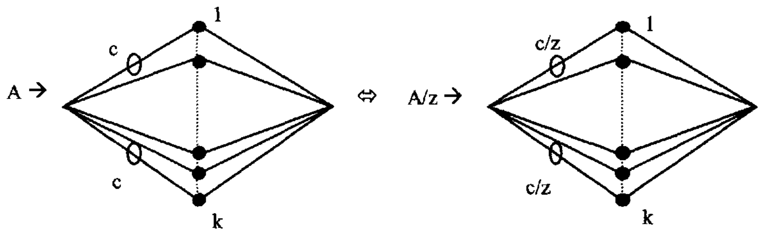


Figure 9.27. *Equivalent graph with the "reduced" network of a multi-bit rate network*

In the previous expressions therefore we need only to replace the Erlang formula by its equivalent multirate formula: $E(\frac{c}{z},\frac{A}{z})$, where c is the number of places corresponding to the smallest bit rate traffic taken as the unit. Thus for example for the network considered, the blocking given by the approximate formula $B = 2E(ck, A) - E(ck, A)^2$ becomes quite simply:

$$B = 2E(\frac{c}{z}k,\frac{A}{z}) - E(\frac{c}{z}k,\frac{A}{z})^2 \tag{9-130}$$

Note that, consequently, for a given blocking probability and a given link saturation probability, call acceptance may be simply based on the equivalent bandwidth approach.

9.2.7.3. *Non-blocking networks*

This section would not be complete if it did not consider the fundamental characteristics of non-blocking network structures. It is also the role of the performance engineer to identify and compare solutions, which although clearly more expensive in hardware terms, can also simplify problems relating to path seeking, bandwidth reservation, and QoS.

Non-blocking connection network structures are frequently referred to as *Clos structures*.

The principle is simple in the light of the previous analyses.

In view of the number of possible inputs and outputs on a network, the aim is to increase the number of paths sufficiently so that at least one free path always remains between a free input and a free output (here also, we are only interested in point-to-point blocking; the case in which all inputs or outputs are occupied is an access problem).

Let us look again at the previous network structure (see Figure 9.28).

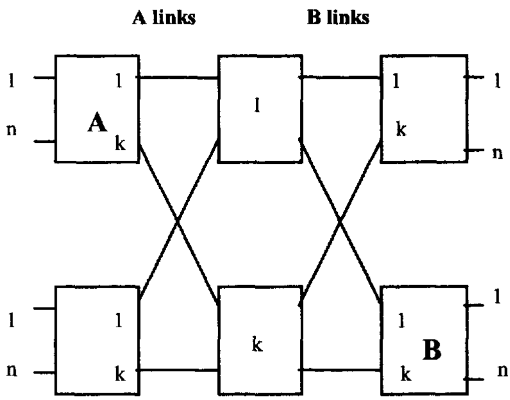


Figure 9.28. *Non blocking structure (k = 2n – 1)*

a) We will first consider the simple case in which $c = 1$ (one single place per link).
The non-blocking condition is:

$$k \geq 2n - 1 \tag{9-131}$$

As a matter of fact, if it is intended to establish a connection between a free input of A and a free output of B, given that this input and this output are free, then there are at most $n - 1$ inputs taken on A and $n - 1$ outputs taken on B. Therefore there are at most $x = n - 1$ internal A or B links taken on each side of the network. For there to be at least two links free (A and B) in correspondence, the necessary and sufficient condition is that:

$k \geq 2x + 1$, therefore $k \geq 2n - 1$.

This is what is called an expansion factor network of $e = 2 - 1/n$. In fact the number of internal links is e times higher than the number of external links. This is the price to be paid to ensure that there is no blocking.

b) Let us now study the case in which $c \geq 1$. A place is said to be free if there is bandwidth available on the link (input, output or internal) that corresponds to the bit rate required by the connection.

As previously, when we want to establish a connection between a free input of A and a free output of B, there are at most $nc-1$ inputs taken on A and $nc-1$ outputs taken on B. And therefore there are at most $x = \left\lfloor \dfrac{nc-1}{c} \right\rfloor$ internal A or B links taken on each side of the network. This can also be written $x = \left\lfloor n - \dfrac{1}{c} \right\rfloor = n-1$.

For there to be at least two free links (A and B) in correspondence, the necessary and sufficient condition is that:

$$k \geq 2x+1, \text{ therefore } k \geq 2n-1 \tag{9-132}$$

We thus obtain the previous condition.

c) Let us now study the case of mixing connections with different constant bit rates.

Let D be the bit rate of a link (still supposed to be the same for all links). Let d be the highest bit rate of the connections to be established. We want to establish a connection *whatever its bit rate* $d_i \leq d$. There are at most $nD - d$ inputs taken on A and $nD - d$ outputs taken on B. And therefore there are at most $x = \left\lfloor \dfrac{nD-d}{D-d+d_0} \right\rfloor$ A or B internal links taken on each side of the network, with $d_0 << d$ depending on the modularity ("granularity") of the bit rates to be transported:

Thus, if all the services have an identical bit rate d and if $D = cd$, we have:

$x = \left\lfloor \dfrac{nD-d}{D} \right\rfloor = \left\lfloor \dfrac{nc-1}{c} \right\rfloor = n-1$. We obtain the previous condition, that is $k \geq 2n-1$.

If the connection bit rates are different, the expansion factor may become far larger, depending on whether d is high or not relative to D. For example, let us consider the case in which $d = D/2$. We obtain: $x = \left\lfloor \dfrac{nD - D/2}{D/2 + \varepsilon} \right\rfloor = \left\lfloor \dfrac{2n-1}{1+\varepsilon'} \right\rfloor = 2n-2$.

And thus with the condition $k \geq 2x+1$, we obtain:

$$k \geq 4n-3 \tag{9-133}$$

This extreme case shows the role of the parameter d_0 which reflects the impact of connections of a lower bit rate than d. In practical terms, it should be understood that a simple bit rate connection $d_i << d$ alone may prevent the establishment of a bit rate connection d, even if a bit rate $d - d_i$. remains.

Conclusion

From a practical viewpoint, when the maximum bit rate to be handled is clearly lower than that of the link, for example 2 Mbit/s on a 600 Mbit/s link, the network behaves as a conventional network. It is possible to guarantee *for any connection with a bit rate lower than d*, either zero blocking, by carrying out expansion to $2n - 1$, or accept a certain blocking, lower than that of the network transporting only connections with the bit rate d. The models and methods presented will enable the identification of the structures that are most suitable for the chosen objective.

9.2.8. *Traffic matrix, Kruithof method*

We will now consider the problem of evaluating the distribution of traffic flows between the various directions of a telecommunications network in the broadest sense (international, national and local network). This knowledge is of course absolutely necessary to determine the capacity of the equipment to be installed.

In many cases, it will be a matter of having to estimate a new distribution, because of traffic growth, the introduction of new services, or the installation of new network and subscriber equipments. The *Kruithof* method is an attempt to resolve this problem.

The problem faced is that of the *extrapolation of traffic data*. To be more precise, we know the present distribution of the flows in the network. This distribution is given by the traffic matrix, the A matrix whose element $a_{i,j}$ gives the value of the traffic circulating from node i to node j (see Chapter

1). The number of nodes in the network is noted N. The sums by lines and by columns, which are respectively designated by $A_{i,.}$ and $A_{.,j}$, represent the traffic leaving node i or entering node j:

$$A_{i,.} = \sum_j a_{i,j}, \ A_{.,j} = \sum_i a_{i,j} \tag{9-134}$$

This matrix is assumed to be known for the current instant t and the problem faced is predicting its value for a future period, which can be noted $t + 1$.

Forecasting tools, based on the general planning methods, taking socioeconomic data into account, enable the prediction of the change in *global traffic* $A_{i,.}$ and $A_{.,j}$ (these traffics are linked for example to the composition of the clientele connected to the nodes and to changes in its behaviour). A direct forecast of the $a_{i,j}$ is however impossible. The purpose of the Kruithof method is to construct a new matrix, whose structure is closest to the initial matrix, in terms of affinity between lines and columns, and complying with the global traffic forecasts.

Note also that, because of its principle, the method enables the calculation of a probable matrix based solely on knowledge (or assumption) of global traffics and proportional distribution of traffics. This is particularly useful in the case of network creation or introduction of new services.

The method is iterative as described hereafter.

Let us note as $a_{i,j}^{(m)}(t+1)$ the traffic forecast for period $t + 1$ at the end of iteration m.

ITERATION 1: On each line a proportional transformation is carried out with the ratio $A_{i,.}(t+1)/A_{i,.}(t)$:

$$a_{i,j}^{(1)}(t+1) = a_{i,j}(t).\frac{A_{i,.}(t+1)}{A_{i,.}(t)}, \ i,j = 1,...N \tag{9-135}$$

ITERATION 2: Clearly, the movement on the lines will disrupt the columns, such that the sums $A_{.,j}^{(1)}$ will not be identical to the forecast. The second iteration attempts to overcome this difference, by renormalising relative to the target value:

$$a_{i,j}^{(2)} = a_{i,j}^{(1)} . \frac{A_{.,j}(t+1)}{A_{.j}^{(1)}(t)}, \ i,j = 1,...N$$

ITERATION 3: As the normalisation of stage 2 has in turn disrupted the sums of the lines, we therefore reiterate stage 1:

$$a_{i,j}^{(3)}(t+1) = a_{i,j}^{(2)}(t+1) . \frac{A_{i,.}(t+1)}{A^{(2)}{}_{i,.}(t+1)}, \ i,j = 1,...N$$

ITERATION 4 AND SUBSEQUENT ITERATIONS: Iteration 4 is identical to 2, and the process continues until a stop criterion is verified.

One possible criterion would be a test on the quadratic error:

$$\varepsilon = \sqrt{\frac{1}{2N}\left\{ \sum\left[A_{i,.}^{(k)}(t+1) - A_{i,.}(t+1)\right]^2 + \sum\left[A_{.,j}^{(k)}(t+1) - A_{.,j}(t+1)\right]^2 \right\}} \tag{9-136}$$

Example of use

Let us assume a matrix of traffic between 5 given nodes, as shown in Table 9.8.

Table 9.8. *Initial traffic matrix*

To From	A	B	C	D	E	Total outgoing
A	10	20	25	18	24	**97**
B	9	12	21	11	18	**71**
C	12	21	30	13	10	**86**
D	19	23	14	10	19	**85**
E	14	13	18	21	22	**88**
Total incoming	**64**	**89**	**108**	**73**	**93**	**427**

Let us now assume that the incoming and outgoing traffic forecasts are given in Table 9.9.

Table 9.9. *Global traffic forecasts*

To / From	A	B	C	D	E	Total outgoing
A						**110**
B						**100**
C						**90**
D						**105**
E						**120**
Total incoming	**70**	**100**	**120**	**100**	**135**	**525**

On the first iteration, we will multiply the elements of line 1 by 110/97, those of line 2 by 100/71, etc. There is thus an initial iteration. The column sums, giving the incoming traffics, differ from the target, and we thus carry out iteration 2, advancing by columns.

Iteration 1

	A	B	C	D	E	Σ
A	11.34	22.68	28.35	20.41	27.22	**110**
B	12.68	16.9	29.58	15.49	25.35	**100**
C	12.56	21.98	31.4	13.6	10.47	**90**
D	23.47	28.41	17.29	12.35	23.47	**105**
E	19.09	17.73	24.55	28.64	30.0	**120**
Σ	**79.14**	**107.7**	**131.16**	**90.5**	**116.5**	

Iteration 2

	A	B	C	D	E	Σ
A	10.03	21.06	25.94	22.56	31.54	**111.12**
B	11.21	15.69	27.06	17.12	29.38	**100.46**
C	11.11	20.41	28.72	15.03	12.13	**87.4**
D	20.76	26.38	15.82	13.65	27.2	**103.81**
E	16.89	16.46	22.46	31.64	34.76	**122.21**
Σ	**70**	**100**	**120**	**100**	**135**	

The same process is continued, and convergence is very rapid. Iteration 5 supplies the matrix that we will adopt as definitive (see Table 9.10).

Table 9.10. *Final forecast traffic matrix*

To From	A	B	C	D	E	Total outgoing
A	9.91	20.78	25.62	22.39	31.31	**110**
B	11.14	15.57	26.88	17.09	29.33	**100**
C	11.43	20.96	29.54	15.53	12.53	**90**
D	20.97	26.6	15.98	13.85	27.6	**105**
E	16.55	16.1	21.99	31.14	34.22	**120**
Total incoming	**70**	**100**	**120**	**100**	**135**	**525**

NOTE: For prediction problems, refer to the UIT planning manual (see bibliography).

This page intentionally left blank

10

Performance Methodology

In this chapter, we present a methodology which, when applied to the various phases of equipment and network development, should ensure that performance and quality of service (QoS) objectives are met. As in Chapter 2, we will stress the generic aspects of the approach, while drawing on concrete examples of existing networks and equipments.

In the preceding chapters, we presented an overview of performance parameters and the corresponding objectives which are generally required by standards or by suppliers of networks and services. We have also presented the tools necessary for calculating these parameters.

We will now show how these various elements can be applied as part of a methodology which is truly integrated with the development life cycle of products and networks. This methodology is aimed at guaranteeing to the equipment supplier that the targets will be met within the required time frames, to the network supplier that the equipments comply with the specifications, and finally to the user that the service provided will meet his/her expectations.

In practice, we will show here the various facets of the performance evaluation profession, focussing in particular on the necessity of a global and coherent approach, from the start of the project up to the experimentation and the operational phase.

10.1. Project life phases

The development of a system, or the setting up of a telecommunications network usually follows what is called a *V-shaped life cycle*, as shown in Figure 10.1.

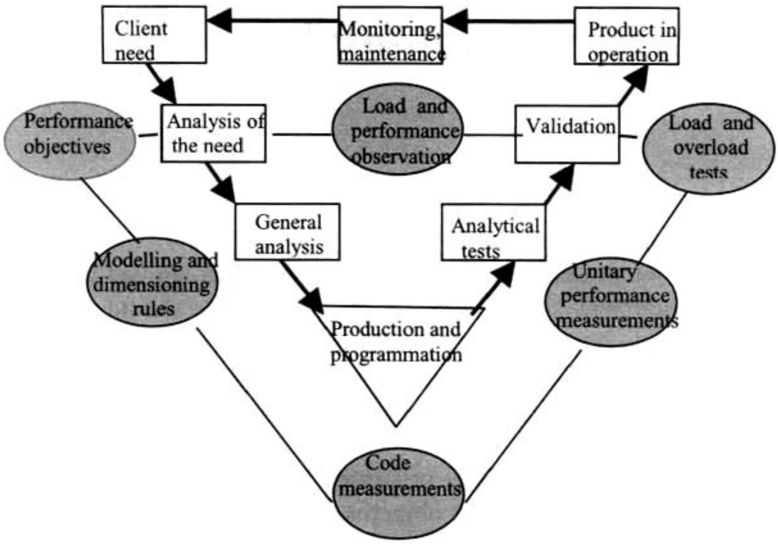


Figure 10.1. *Project life cycle and performance activities*

Six major activities can be identified here: analysis of need (or contract definition), general analysis (or architecture definition), production, analytical tests, validation and monitoring-maintenance.

The performance evaluation methodology is itself associated with six other activities in parallel: performance objectives definition, architecture modelling, code measurements, unitary performance measurements, on-load tests and monitoring of performance during operation.

For each of these activities, which we will consider in detail, one or more documents will normally be compiled, and these are listed in Table 10.1.

We will now set out the main tasks in detail and provide examples

Table 10.1. *Performance activities and documents*

Performance documents		
Project phase	Title	Object
Contract definition	1-Reference environments and performance objectives	Specifies traffic hypotheses, mechanical and climatic conditions, capacities and performance objectives under normal load and overload conditions.
Definition of architecture	1-Modelling of traffic performance	Models architecture in terms of clients and servers, evaluates capacities and response times. Proposes solutions.
	2-Modelling of dependability performances	Models architecture in terms of redundancies, defence mechanisms and deduces availability, etc. Proposes solutions.
	3-Dimensioning rules	Defines the capacities of each element and the rules for calculating the number necessary to handle the traffic.
	4-Traffic and dependability test strategy	Defines the list of tests to be performed to verify performances, and the necessary test facilities.
Development of code	1-Definition of code measurements.	Defines the code parts to be measured, and the means.
	2-Measurements and results	Interprets the measurements and provides an overview of results.
Analytical tests	1-Definition of unitary performance measurements	Defines the types of calls and processing to be measured (CPU time, etc.), and the means.
	2-Measurements and results	Interprets measurements and provides an overview of results.
Validation/ Integration	1-Definition of on-load and overload tests	Defines the call mixes to be tested, the means, and the on-load capacities and overload capacities to be checked.
	2-Definition of dependability tests	Defines the faults to be injected, and the defence mechanisms to be checked.
	3-Tests and results	Interprets tests and provides an overview of results.
Client support	1-Presentation of performances	Presents performances and dimensioning rules to clients.
	2-Monitoring of sites	Observes on-site behaviour. Analyses, explains and corrects site problems. Draws conclusions.

10.2. Analysis of need

Definition of performance objectives and reference environments

The analysis of need, also called contract definition phase, is upstream of the life span of the project and in fact is at the basis of the birth of the project. It consists of analysing the external specifications, standards and specifications, to identify in terms of services (functions and capacities) the content of the project, its duration and the associated realisation costs. During this phase, the performance engineer will participate in the definition of the global capacity targets of the projected equipments, and identify the main performance parameters and objectives to be met, and the reference environments to be taken into account (both in terms of traffic and dependability).

These three aspects are intimately linked: determining a traffic flow capacity is only meaningful if the traffic hypotheses are defined, i.e. the mixes or the proportions of calls of each type to be handled, together with the expected response times or loss rates. For example, the bit rate required by a video service (several Mbit/s) is not the same as that required for voice (a few kbit/s). Similarly, a telephone call with additional services requires more resources than an ordinary call, and an unsuccessful call (failing because of congestion for example) occupies network resources for less time than a successful call resulting in a call of several minutes, or several tens of minutes for IP sessions. Finally, response time requirements of greater or lesser severity may considerably change the number and nature of equipment to use, and thus the cost of the products developed, and finally the project cost.

10.2.1. *Reference environments in traffic*

When defining performance and particularly capacity targets, it is clearly necessary to define which service types are to be handled and for which traffic volumes. Clearly also, it is not a question of defining a product or network for a single value and single traffic mix, nor at the other extreme designing a product that is ideal for everything. This would either be unrealistic or uncompetitive. The aim therefore is rather to define one or more reference models, which will be termed *reference mixes*. As we shall see, they will reflect proportions of activation for each traffic type, and on this basis it will simply be a matter of determining and testing the characteristics of architectures to meet performance targets.

Beginning with a *typical population of users,* described by the *proportion of each category* (professionals, residentials, large and small companies, etc.), services, and what are known as *service penetration rates* and *service activation rates*, are associated. For example, it will obviously be understood that a professional user (such as a large company) will make more use of the broadband videoconference facility than a residential user (the normal subscriber in his/her home will not be equipped with the necessary equipment, and will use a simpler service). Based on a similar penetration rate, we can also conclude that the utilisation or activation rate of a service such as voice calls or the exchange of e-mails will be far higher for a professional than for a residential user (there is no comparison between the two cases in terms of number of calls per time unit and number of erlangs per user).

The other dimension to be considered when establishing a reference mix is that of *traffic flows* relating to the determination of the *traffic matrix*. Whether at network or network element level, it is necessary to distinguish between incoming flows, outgoing flows, local and transit flows, as explained in Chapter 1.

To explain what this means in practice, we give below a few examples of mixes, in the field of fixed telephony, mobile telephony and IP. We will not set out in detail the method used to obtain each of these mixes, but we will give an example, and will finally show how the result must above all be synthetic.

For the first example in fixed telephony, a distinction is generally drawn between four user categories: residential users, small companies, medium-sized companies and large companies. Furthermore, it is necessary to distinguish between users equipped with broadband lines and others (ISDN subscribers, ADSL subscribers, etc.). An equipment rate and a utilisation rate is allocated to each of these categories. Then, allowing for the distribution of population, the characteristics of mean demand are obtained.

Tables 10.2 and 10.3 sum up this approach for fixed subscribers in a country with a high equipment rate and high efficiency.

These are only indicative values illustrating an approach which is very widely applied to all types of networks and services as we will see later.

The mean call time is taken here to be equal to 150 seconds., which can easily be verified: e.g. $(0.96 \times 150)/3600 = 0.04$. In view of the distribution of subscribers, an average value is obtained for the average subscriber:

Table 10.2. *Telephone service: penetration and activity*

Penetration and activity				
Subscriber category	Number of 64 kbit/s accesses	Activity in erlangs/access	Activity in BHCA/access	Distribution %
Residential	1	0.04	0.96	84.4
Small companies	1.2	0.12	2.88	14.1
Medium companies	2.7	0.15	3.6	1.49
Large companies	92	0.3	7.2	0.01

Table 10.3. *Telephone service: average penetration and activity*

Average subscriber	Number of 64 kbit/s accesses	Activity in erlangs/access	Activity in BHCA/access	Distribution %
	1.06	0.06	1.44	100

This type of approach will be used for various types of access and service. At this stage we therefore obtain an expression of *demand* (in terms of calls and erlangs) per subscriber. It is very important to note that this expression will enable us to link up equipment capacities in terms of calls to subscriber connecting capacities: an equipment capable of handling 1 call/second will be capable of handling the traffic of 3600/1.44= 2500 "average" subscribers.

To determine the processing capacity, we now need to specify the call mix. In the relatively simple case of *fixed telephony*, to begin with, a distinction is generally drawn between two typical mixes: the high efficiency mix for countries which are very well equipped, and the low efficiency mix for less well equipped countries, such as developing countries. Efficiency is an important parameter as the handling of calls and the resources engaged differ substantially, as mentioned earlier. For the same reasons, it is necessary to specify the rate of utilisation of *subscriber* IN services such as CNIP (*call name identity presentation*), credit card, free phone, voice mailbox, etc., and *networks* services such as portability, the prepaid service (your account is credited for a certain number of calls), and black and white lists (call filtering). These services can be broadly divided into two categories: simple and complex. From the processing viewpoint, these services are added as extras to basic calls (outgoing, incoming). Tables 10.4 and 10.5 sum up the main characteristics of a fixed subscriber centre. The values are of course indicative but stem from real-life cases.

Table 10.4. *Fixed telephone service: high efficiency mix*

High efficiency mix				
Call type	Internal	Outgoing	Incoming	Transit
% Flow	*17*	*40*	*33*	*10*
% Successful	58	58	70	70
% No reply	15	15	18	18
% Busy	10	10	12	12
% Incorrect dialling	17	17	0	0
General characteristics and supplementary services				
Average call time	160 s			
Average conversation time	228 s			
% simple IN services / call	30			
% complex IN service / call	5			

Table 10.5. *Fixed telephone service: low efficiency mix*

Low efficiency mix				
Call type	Internal %	Outgoing %	Incoming %	Transit %
% Flow	*5*	*50*	*30*	*15*
% Successful	40	36	47	42
% No reply	10	9	11	10
% Busy	25	23	29	26
% Incorrect dialling	25	22	13	12
% Network failures	0	10	0	10
General characteristics and supplementary services				
Average call time	60 s			
Average conversation time	100 s			
% simple IN services / call	10			
% complex IN services / call	5			

The reader will observe the marked differences in busy rates (a private individual's set acts as a "telephone kiosk" in a country where there are few telephones), incorrect dialling and network failure (the network is sometimes undersized or partially out of order). Note that these models change rapidly as the countries concerned develop.

Let us now consider another important example: mobile telephony.

This example is interesting for more than one reason. Theoretically, a detailed approach is needed to characterise the various types of subscriber (pedestrians, motorists, etc.) in terms of mobility, depending on cell size, and network organisation. In fact, very rapidly a macroscopic characterisation has emerged very similar to that of fixed networks: i.e. in terms of call types, as OC (*originating calls*), TC (*terminating calls*), GW (*gateways, or rerouted call or call in transit),* while adding the typical attributes of mobility, i.e. HO rates (*handover, or change of coverage zone)* and *LR rates* (*location register* i.e. subscriber location), thus reflecting the average mobility observed on the network considered.

Furthermore, the operators have found it more convenient to directly associate the value of the activity with the different types of calls. This leads to a single table, which is a merging of the previous activity and call mix tables. This is only a matter of presentation, and can change depending on use and needs.

Table 10.6. *Mobile telephone service: low traffic mix*

Average traffic mix					
Call type	OC	TC	GW	OSMS	TSMS
Number of calls/hour/MS	*0.487*	*0.3*	*0.357*	*0.15*	*0.15*
% Successful	70	55	70	100	50
% No reply	10	20	10		50
% Busy	All	5	20		
% Called party not accessible		20			
% Unsuccessful rerouted to voice mailbox		80			
Mobility	HO intra	HO inter	LR intra	LR inter	
Events/hour/MS	*0.27*	*0.10*	*0.6*	*1.2*	
General characteristics and extra services					
Average call time	120 s				
Average conversation time	150 s				
% simple IN services / call	20				
% complex IN services / call	10				

Table 10.7. *Mobile telephone service: high traffic mix*

High traffic mix					
Call type	OC	TC	GW	OSMS	TSMS
Number of calls/hour/MS	*3*	*2.1*	*2*	*0.05*	*0.05*
% Successful	70	68	70	100	50
% No reply	20	1	20		50
% Busy	10	2	10		
% Called party not accessible		29			
% Unsuccessful rerouted to voice mailbox		50			
Mobility	HO intra	HO inter	LR intra	LR inter	
Events /hour/MS	*0.13*	*0.05*	*0.9*	*0.9*	
General characteristics and extra services					
Average call time	80 s				
Average conversation time	100 s				
% simple IN services / call	20				
% complex IN services / call	10				

In practice, here also a distinction has been drawn between two main types of mix: the average traffic mix that in fact corresponds to the normal case of countries that are already well equipped with fixed telephones, and the high traffic mix found in countries which have given a very high priority to the development of their mobile network (See Tables 10.6 and 10.7).

As in fixed telephony, there are extra IN services such as the prepaid service, and also a particular and very attractive type of call: the SMS (*short messages service,*) which uses only signalling resources (as for other calls we will speak of originating and terminating SMS).

Let us now consider a final complementary example, that of the characterisation of a multi-service IP environment. Although this domain is changing fast, the main characteristics can be identified (See Chapter 9):

– division into three main levels of activity: call or session level, flow level and packet level;

– division into two main traffic categories: stream traffics with real time constraints and elastic traffics, with no strong real time constraint.

Modelling studies for the performance and sizing of equipment concerned with these traffic types have demonstrated that (see Chapter 9):

– for elastic traffic it is enough to know the mean volume of flows offered to the network and its equipment, with the sizing and performances being evaluated by the processor sharing model;

– in the case of stream type traffic, knowledge of the mean and variance of flow bit rates is sufficient for the sizing and evaluation of performances by the use of Erlang multi-bit rate models.

Table 10.8 illustrates the case of a prospective study for a user group of an IP multiservice network using the following stream services: voice service with coding at 64 kbit/s (G.711), or with G.729 coding with silence suppression, video conference services (H.261) and videocommunication (H.263), real time video services for both high quality (MPEG 2), and lower bit rate (MPEG 4), and finally elastic services such as the Web (e-mail, etc.) and file transfer (FTP). Note that we are concerned here with an "average" subscriber of course.

Table 10.8. *Mix in a multi-service environment*

Service	**VoIP G711**	**VoIP G729A**	**H261**	**H263**	**MPEG2**	**MPEG4**	**WEB**	**FTP**
Penetration/ user	0.5	0.5	0.01	0.01	0.1	0.2	1	0.01
Activity (erlangs)	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Bit rate/flow - peak (kbit/s)	80	24	384	128				
- mean	80	12	384	128	500	64	15	8
- variance					250^2	32^2		

Attentive readers will have noticed that for ON-OFF type traffic (G.729 with VAD), as described in Chapter 9, the peak bit rate and the mean bit rate are the only ones given, as the variance can be directly deduced from them. For video traffic however the variance is necessary. For elastic traffics (Web and FTP), only the mean value is necessary (see Chapter 9).

For complete characterisation of demand, it will be necessary to associate with this description values for efficiency and flows similar to those set out in the previous tables.

We will not provide any more examples, as the important thing is to indicate the approach, so that it can be applied and adapted to the various situations of a field that is constantly changing.

Let us now consider the aspects relating to dependability.

10.2.2. *Reference environments for dependability*

The aim here is mainly to define two main types of parameters: those that describe the climatic and mechanical operating conditions used to calculate the failure rates of components, boards, transmission media, etc., and those that characterise the maintenance policy used to calculate the availability of equipments and networks. Distinctions will therefore be drawn between equipments that are to operate on the ground, those carried on satellites and those subjected to a submarine environment.

These environments are usually well defined in what are known as *reliability handbooks (or guides)*, and from these rules are obtained (usually simply factors) to be applied to the formulae used to calculate equipment failure rates. As for the software, it is considered that its failure rate is practically independent of these constraints as the main factor is very much the rate of use of the code, and this is only a function of traffic. We set out below as an example the case of terrestrial telecommunications systems, in a protected room (air conditioned, earthquake-proof buildings, etc.). This environment is generally termed *ground benign*.

Table 10.9. *Mechanical and climatic environmental conditions*

Mechanical and climatic conditions		
Vibrations	Acceleration	1 m/s^2
	Frequency	200 Hz
Shocks	Amplitude	50 m/s^2
	Duration	22 ms
Relative humidity		40% to 70%
Temperature	Ambient at component level	40°C

With regard to maintenance (see Chapter 2), it is characterised by the various intervention times and the on-site repair times that the operator allows itself.

In fact this is a trade-off between the consequences on the service offered to the user and the target of facilitating maintenance, e.g. by authorising deferred interventions. Table 10.10 provides an example of average values adopted by several operators.

Table 10.10. *Maintenance conditions*

Maintenance conditions	
Intervention type	Intervention time
Immediate intervention	3.5 hours
Deferred intervention	12 hours
Non-imperative intervention	72 hours
Average on-site repair time: 30 minutes	

10.2.3. *Capacity and quality of service objectives*

Having defined the traffic and dependability reference environments, we now turn to the definition, under these conditions, of the capacity targets, mainly in terms of:

– number of erlangs,

– number of calls,

– number of packets to be handled,

– number of users,

– number of circuits, STM links, etc., that can be connected,

– QoS parameters required.

As far as the handling and connection capacities are concerned, the aim is in fact to fix the *marketing* and *modularity targets*.

On the one hand, the aim is to satisfy various sectors of the market, which means using either access equipments (able to connect clusters of users, local or remote)), or large switches or routers in the core of the networks. Furthermore, the equipments must be modular to adapt to the various needs in capacity terms and the clients' needs for evolution: the same operator may manage different networks (access, core, fixed, mobile), but must also provide for extensions to the capacity of its equipments, and evolutions of services. Handling capacity targets are mainly expressed in terms of processing capacity in BHCA, packets/s, or messages/s. Traffic handling capacity targets will be expressed in erlangs. Connection capacity targets are expressed in numbers of subscribers and connectable links (PCM, STM, etc.). Note that these targets will be stipulated mainly for Load B level (usually the most demanding) but also for Load A level (see Chapters 1 and 2).

With regard to QoS and the associated performance parameters, the aim is primarily to identify and select the parameters and standards that apply to the project in question: traffic standards, dependability standards, overload standards (see Chapter 2). Usually only a handful of fundamental parameters are considered. A brief example of the case of an access node is given below.

Traffic performances

Table 10.11. *Traffic performance objectives*

Parameter	Normal load (A)		High load (B)	
	Mean	**95%**	**Mean**	**95%**
Access outgoing call set-up delay (Q543)	600ms	800ms	800ms	1200ms
Incoming call indication sending delay (Q543)	650ms	900ms	1000ms	1600ms
Access call release delay (Q543)	250ms	300ms	400ms	700ms
Access call blocking probability (Q.543) for: – outgoing calls – incoming calls	 0.5% 0.5%	 N/A N/A	 3% 3%	 N/A N/A
Message transfer delay / real time packet (Q766/Y1541)	180ms/ 100ms	360ms	450ms	900ms
Variation of transfer delay of an IP packet at 10^{-3} (Y.1541)	50ms	Not specified	Not specified	Not specified

Dependability performances

Table 10.12. *Dependability performance objectives*

Unavailability of an access equipment for: – a user (Q541) – all users of this access (operators)	 30 min/year 3 min/year
Premature release at node level (Q543)	2 10-5
Maintenance load at access node level (operators)	< 15 failures per year per 10,000 users

For market-related reasons, more ambitious performance objectives may be specified. Above all, directives should be given concerning the distribution of these values between the various subsystems (what is the "budget" for each: connection, matrix, control) and ensure feasibility. By this stage, a substantial feasibility study should already have been carried out. The document for the *specification of performance objectives and reference environments* is therefore a crucial overview document for the success of the project. As such it should be discussed and approved both by the project managers and by the marketing managers.

10.3. Modelling

We now turn to modelling, which with tests forms the core of our profession. During the general analysis phase, when the general architecture of the system is defined, modelling translates real time operation into terms of *clients* and *servers* and evaluates their performances. This modelling will strongly contribute to the choice of the architectures most appropriate for supplying the service required, based on the required quality of service criteria.

The modelling task will be highly varied depending on the nature of the project, and on whether it is a completely new project from the hardware and software viewpoint, or a mere hardware or software upgrade. In order to cover the whole range of techniques used, we will illustrate this activity through the example of a totally new project.

The first stage consists of establishing the call set-up diagrams and the forwarding of the information flows exchanged through equipments and networks, in order to identify the processing to be performed and to identify the resources, or *servers*, that are involved from the quality of service viewpoint.

The control and user information message flow diagrams will be established for the various services, along the same lines as those presented in Chapter 2. Based on traffic mixes and capacity targets, the volumes of the various flows, the associated processing and the resources engaged will be determined.

An initial distinction will be drawn between two main types of resources: *processing units* and *defence unit*. These concepts correspond to entities that can be considered as independent from the viewpoint of traffic or dependability. They may relate to the control plane or to the user plane. A processing unit may for example be a functional block (a processing software entity), a processor, or a station with several processors, a transmission link, etc., such that the impact on traffic flow is clearly identified. Similarly, a defence unit may be a processor, a multiprocessor station, a network node or a link, etc. A defence unit is a set of hardware and software that can be put out of service in the event of a fault, and whose impact on service continuity is clearly identified.

These units will then be characterised in performance terms.

For processing units we will thus define the *type of service*: processor sharing, cyclic service, M/M/1 or M/D/1 server etc., and the queue and priority management mode. For this purpose, reference will be made to the theoretical models presented in the earlier chapters.

The *service time distributions* will be determined on the basis of the characteristics of the flows to be handled and allowing for the mixes. For example, call set-up software processing times can follow an exponential or constant distribution, while packet lengths will tend to have heavy tailed distributions. In each case, it is important not to neglect the influence of the mixes, which fortunately often bring us back to simple distributions (e.g. exponential). As for the service time itself, although it is easy to evaluate it for a message or packet on a transmission medium, it is more complex to estimate it for a software processing time. This is where we will use processor power forecasting methods such as those presented earlier: breakdown into primitives, extrapolation, *spec-integer,* etc.

In the case of defence units, the following will be defined: failure rates (hardware and software), redundancy type, switchover and re-configuration times. For this purpose, reference should be made to the models presented in Chapter 6. The following will also be defined: the number of sessions and calls simultaneously in progress and that may be affected by a failure. Similarly, messages and packets may be lost in the event of a failure, even with rerouting possibilities.

Finally, the last stage will consist of evaluating the performance parameters as expressed in the performance objectives specifications and of course the standards. This is where the theoretical calculation models developed in Chapters 6 and 7 will be used.

For this purpose, from the traffic viewpoint, for each parameter concerned we will establish the sequence of processing required and the series of resources engaged, in accordance with the diagrams in Chapter 2. Based on the previously defined traffic hypotheses and models, we will then evaluate the load of each entity and the response times, or individual blocking probabilities. Finally, we will evaluate the global behaviour of the elements in series based on the properties of the sum of random variables. Fortunately, we can often consider the elements in series as independent, and thus make use of the properties of the convolution products of independent variables as set out in Chapter 3. This is particularly the case if care has been taken, during the design of the system, to comply with the rules recommended in Chapter 9 concerning queueing networks. Even more simply, it is possible for example to approximate a global response time by a normal or Gamma law, whose moments will be the sum of the moments of the individual laws.

To make this more concrete, we will consider some examples, beginning from the lowest level and moving on to the most general level, i.e. from the equipment level to the network level.

10.3.1. *Message processing delay*

This initial evaluation is essential for the estimation for example of the capacity of a control equipment, and subsequently (as we will see later) for the evaluation of the response times of systems. As early as the objectives definition phase, initial evaluations of these processing delays, and evaluations of the number of messages to be processed depending on the mixes adopted, will enable the determination of the feasibility of a global capacity target, and the technology or the number of equipments to be employed. As we saw in Chapter 9, the exercise is extremely risky when we are considering both a new project and new technologies. However, by combining experience, analysis and measurements, it is possible to obtain the first significant elements.

Let us now look again at our modelling of the processor board.

It should first be stated that the concept of "message" used here, which corresponds more to messages actually exchanged on a bus, could equally well be applied to messages exchanged between software blocks.

The processing delay of a message can be broken down into three main parts: reception of the message with activation of the consumer process (*primitive receive*), application level processing of the message and the sending of the subsequent message (*primitive send*).

We will apply the execution delay evaluation model for a processor instruction described in Chapter 9. The simplified model, we recall, is the following:

$$I = h_{ip}.t_{ic} + (1-h_{ip})(t_{im}) + d_i(1-h_{dp})(t_{dm}) + d_e.(a_e + t_e)$$

where:

h_{ip} = instruction rate in the primary cache, t_{ic} = execution delay in the cache, t_{im} = execution delay for an instruction in private memory;

d_i = data read rate per instruction;

h_{dp} = data rate in the primary cache, t_{dm} = access delay to data in private memory;

d_e = data read rate in exchange memory, a_{e+} t_e = read delay in exchange memory.

Message reception

The data processed are mainly in local exchange memory, and the code is limited but executed intermittently, with an access rate to the data per instruction of d_i = 0.6. The size of the code is evaluated at about 200 instructions with h_{ip} and h_{dp} rates of around 0.8. For each message the following take place: management of queues (exploration, unchaining, occupation table management) and transfer of message into private memory.

We have thus returned to our example from Chapter 9. With a 266 MHz processor we had obtained: t_{ic} = 3.76 ns, t_{im} = 23 ns, t_{dm} = 151 ns, a_e = 1510 ns, t_e = 944 ns. Hence:

$t\ receive = 200 \times 0.8 \times 3.76\ ns + 200 \times 0.2 \times 23\ ns + 200 \times 0.6 \times 0.2 \times 151\ ns$
$+ 1 \times (1510\ ns + 944\ ns) \sim 7600\ ns$

Message sending

Without going into detail, let us assume a similar value for the sending of a message:

$t\ send = 7.6\ \mu s$

Let us now evaluate the "cost" of application processing. We are in the case of software with many context changes (processor sharing type on several processes of call set-up or release). The analysis of the code leads to a number of instructions in the order of 2500, with a h_{ip} rate of 0.6. In these phases, there are no external exchanges. We have an access rate to data, per instruction, of 0.6, and the data are also in the cache with a h_{dp} rate = 0.6. We thus have, as in our example in Chapter 9:

$t\ appli = 2500(0.6 \times 3.76\ ns + 0.4 \times 23\ ns + 0.6 \times 0.4 \times 151 ns) \sim 119{,}000\ ns$

And thus finally:

$t\ mess = 7.6 + 119 + 7.6 \sim 134\ \mu s$

It is advisable to set aside a precautionary margin of around 10% to 20%.

In conclusion, the message processing delay can thus be estimated at around 150-160 μs.

Bearing in mind that an "average" call requires the processing of about 50 messages (sending and reception), a processor of this type when 80% loaded could process around 100 calls per second. The subsequent modelling of the service and of all the other resources of the system will enable us to determine precisely the maximum load and the response times compatible with the quality of service targets. We consider this point in the following example.

10.3.2. *Incoming call indication delay*

Let us now consider a calculation example for a call set-up phase in a switch. This example will enable us to show what the modelling of a multiprocessor control system can be, and to show that the values specified by the standards for equipments tend to be on the pessimistic side, in view of the increased power of the new technologies.

In schematic terms, the system consists of several processors, which may or may not be specialised in various call set-up functions.

An essential distinction is drawn between so-called *main functions* corresponding to call processing supervision (CP) and so-called *secondary functions* corresponding to the realisation of services such as translation (TRA), charging (CHA), the management of extra intelligent network services (IN), and the processing of network signalling such as no. 7 signalling channel management (SIG), and finally "termination" (connection) units for subscribers (STU) and 64 kbit/s circuits (CTU), and of course the switching matrix (MCX).

The IN services are piloted by an IN server (SCP), which is generally common to several switches and accessible through the network.

The diagram in Figure 10.2 is a schematic of the switch with IN services, i.e. an SSP (*switching service point*), with its various stations interconnected via its local network.

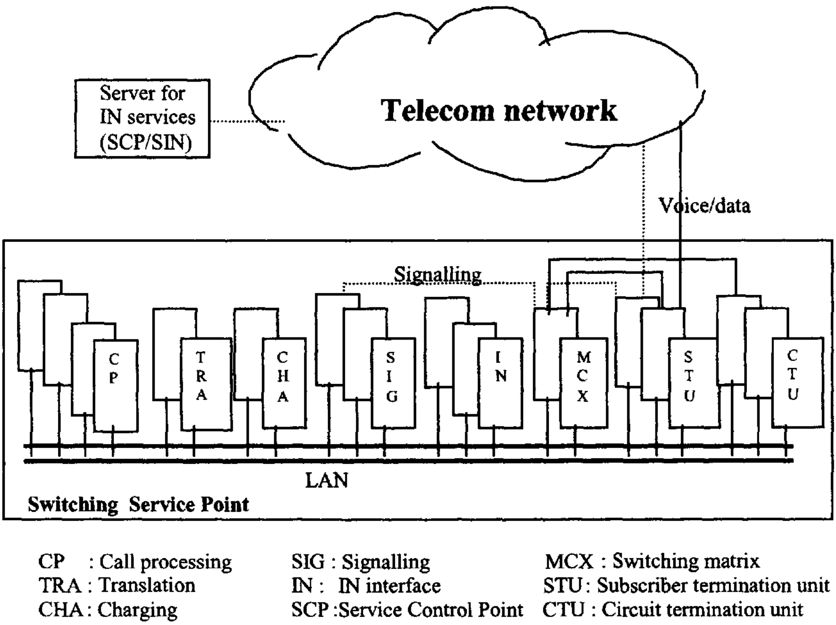


Figure 10.2. *Switching service point architecture*

To simplify the representation and the analysis, a function is associated with a station, which may actually be the case for very large switches, while for small exchanges functions may be concentrated on a single station. But this does not change the performance evaluation method in any way. To achieve the very large capacities desired, there are usually several stations of each type on which the new calls are distributed with equiprobability.

The functions are usually (because of the standards in fact) implemented in the form of functional blocks with clearly defined interfaces. To carry out a call set-up, the various machines and blocks exchange a series of messages sequentially through a local network (LAN). The messages correspond to service requests and their responses, thus enabling the chaining of the various elementary processings, until the set-up function has been completely realised. In view of the great number of independent calls in progress simultaneously, and the variety of the messages exchanged, it is justified, and verified, to consider all these machines as independent servers subject to arrivals of Poissonian requests.

We will consider the case of the set-up of an incoming call with an IN service of the free phone type, and estimate the set-up delay of the incoming call, corresponding to the standard for *incoming call indication delay*.

The diagram in Figure 10.3 provides an example of the messages exchanged and the processings performed on the various machines, with involvement of the IN server, between reception of a call set-up request from the network (IAM) and the transfer of this request to the connection unit of the subscriber requested.

The following are present in succession. First, the call set-up request signalling message (IAM), from the network, transmitted by the signalling processor (SIG) to call processing (CP) which will reserve an incoming circuit for the call. Then the CP will ask the translator (TRA) for the characteristics of the number requested (discrimination, destination, etc.). The translator detects whether it is a free phone number (0800, etc.). The CP then reserves a context in the charging unit (CHA) to charge for the service (to the service provider), and then presents the service request to the IN server via its IN interface and then via the network. The IN server common to several switches is accessed like the rest of the network by using network signalling. In the normal case, i.e. when the service can be provided, the IN server answers the switch by asking it to set-up the connection to the party called and to start up the charging. The CP then asks the translator to give it the designation, the outgoing direction corresponding to the number requested (the number must be translated into a physical equipment address, and the outgoing links from the switch to the subscriber connection unit of the subscriber requested must be determined). The incoming circuit is then connected to the outgoing circuit, and the set-up request message is sent to the subscriber connection unit of the called party.

Thanks to the study of the architecture of each machine (we will provide an example later), we can evaluate their response time by means of relatively simple models.

Thus for example the CP can be modelled by a processor sharing server such that for a task with a time of s_i, the mean and the variance of the response time are (see Chapter 9):

$$\bar{t}_i = \frac{1}{1-\rho} s_i$$

and:

$$\sigma_i^2 = \frac{2\rho s_i}{\mu(1-\rho)^3} - \frac{2\rho}{\mu(1-\rho)^4}(1 - e^{-\mu(1-\rho)s_i})$$

$1/\mu$ being the mean task duration, exponentially distributed.

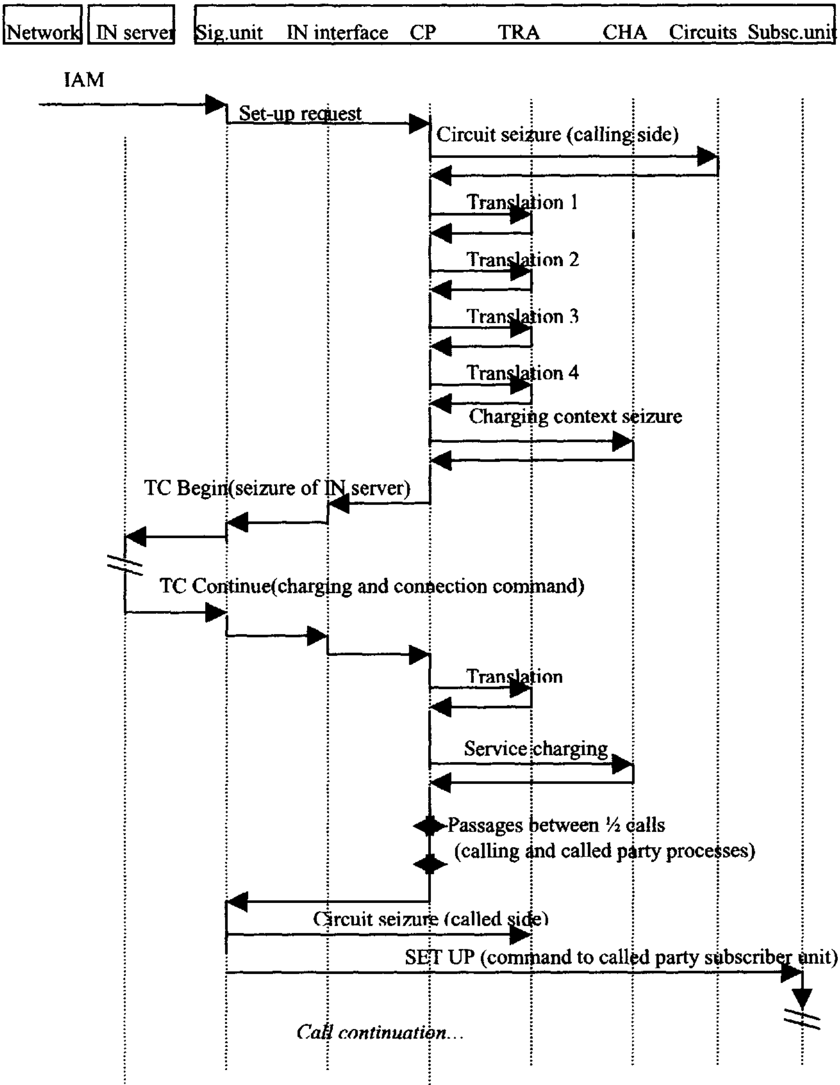


Figure 10.3. *Incoming call indication, with free phone type IN service*

The processor sharing model is extremely suitable for this type of machine, which shares its time equitably between several processes simultaneously in progress in the machine, such as call set-ups. In particular, it enables a fine estimation of the response time for tasks of different durations. However, as presented in the diagram, because of the variety of tasks involved in the call phase studied, we will be interested not in a single type of processing but in processing as a whole, and thus with the mean value $1/\mu$. From the previous formulae we can deduce for an average processing (see Chapter 9):

$$\bar{t}_{CP} = \frac{1}{1-\rho}\frac{1}{\mu} \tag{10-1}$$

$$\sigma^2{}_{CP} \approx \frac{2+\rho}{(1-\rho)^2(2-\rho)}\frac{1}{\mu^2} \tag{10-2}$$

The other ("secondary") machines are presented basically as elementary task servers, with a quasi-constant service time s. We therefore adopt the M/D/1 model, whose response time characteristics are as follows (see Chapters 7 and 9):

$$\bar{t}_{SEC} = \frac{1}{2}\left(\frac{2-\rho}{1-\rho}\right)s \tag{10-3}$$

$$\sigma^2{}_{SEC} = \frac{1}{(1-\rho)^2}\left(\frac{\rho}{3}-\frac{\rho^2}{12}\right)s^2 \tag{10-4}$$

To these response times must be added the sending and reception times of the messages exchanged on the LAN.

These times can be assimilated to a constant c (for transmission and reception), plus the response time for an exponential server. For it is a matter on the one hand of message acquisition, formatting and routing delays and, on the other hand the waiting time and the sending delay on the internal communication link. For the sake of simplification, this server will be approximated to an M/M/1 server (see Chapter 7). Thus:

$$\bar{t}_{LAN} = \frac{1}{1-\rho}\frac{1}{\mu_{LAN}}\text{, plus the constant } c \tag{10-5}$$

and:

$$\sigma^2{}_{LAN} = \frac{1}{(1-\rho)^2}\frac{1}{\mu^2{}_{LAN}} \tag{10-6}$$

We are simplifying the calculations here by not differentiating between tasks in detail, but simply by assuming orders of magnitude for the processing delays depending on the two types of servers. Thus, we will assume an average processing delay of 150 μs for the CP and a delay of 250 μs for a service processor. (We will see below to which processing capacity these values correspond.) We are assuming the LAN to be composed of 100 Mbit/s links, and messages of a mean length of 100 bytes, giving a transmission delay in the order of 8 μs per message. We assume c to be 50 μs. The processors are all assumed to be 80% loaded, and the LAN to be in degraded mode (there are *at least* two links for dependability reasons; in normal conditions, each link is loaded at 0.4).

We thus obtain:

$$\bar{t}_{CP} = 0.75ms\ ,\ \sigma_{CP} = 1.14ms$$

$$\bar{t}_{SEC} = 0.75ms\ ,\sigma_{SEC} = 0.577ms$$

$$\bar{t}_{LAN} = 0.032ms\ ,\sigma_{LAN} = 0.032ms$$

The response time of the IN server (SIN) is an item of data, external to the switch, of which we assume the $\overline{T}_{SIN}$ and σ_{SIN} characteristics to be known. In our example, we assume that:

$$\overline{T}_{SIN} = 20ms$$

$$\sigma_{SIN} = 20ms$$

bearing in mind that we include in them the access delays through the signalling network.

Based on the message diagram, we can write:

$$\overline{T}_{ICISD} = 12\bar{t}_{CP} + 15\bar{t}_{SEC} + 24\bar{t}_{LAN} + 24c + \overline{T}_{SIN} \approx 42ms$$

and:

$$\sigma_{ICISD} = \sqrt{12\sigma^2_{CP} + 15\sigma^2_{SEC} + 24\sigma^2_{LAN} + \sigma^2_{SIN}} \approx 21ms$$

and finally, still using our approximation:

$$\overline{T}_{(95\%)ICISD} = \overline{T}_{ICISD} + 2\sigma_{ICISD} = 84ms$$

It will be noted that these times are easily in line with our target. This is not surprising as the switches, as we will also see in the case of the routers and network links, use equipments whose technologies are increasingly powerful. However, let us not forget that we have considered a relatively simple service case to facilitate comprehension.

Finally, the important impact of the IN server is worth noting. As already stated, the IN server is usually a network element that is external to the switch. As such its response time characteristics will also be specified by a mean value and a 95% value. In our example, we have taken $\bar{t}_{SIN} = 20ms$ and $t_{(95\%)_{SIN}} = 60ms$. By using an approximation to the Gamma law, we deduced from this: $\sigma_{SIN} = (t_{(95\%)_{SIN}} - \bar{t}_{SIN})/2 = 20ms$.

10.3.3. *Transfer delay for packets through a network and variation in transfer delay*

Let us consider the following structure:

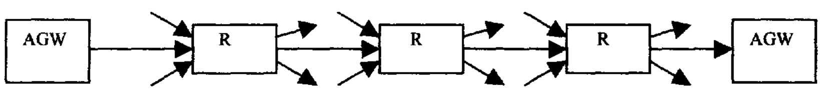


Figure 10.4. *Model for packet network transfer delay*

This diagram represents the succession of nodes and links passed through by a message or packet on the network(s) considered. The *access gateway* (AGW) connects the subscribers, "packeting" and "depacketing" the information, and the routers (R) forward the packets through the network.

We have adopted the simplifying assumptions (to be carefully verified) that the structure considered is strongly meshed (large number of directions at each node, and several flows on the same link), and that traffic flows are strongly mixed inside nodes, thus minimising series queue problems (see Chapter 9). This will not be verified for all routing policies, as some (such as tunnelling) favour convergence of flows on the same itinerary through a large number of consecutive nodes. This hypothesis may however be considered to be acceptable on an initial approximation. In these conditions, we can adopt the assumption of independence between nodes and neglect the series queue phenomena, and also consider the arrivals as Poissonian.

We will therefore consider the packet or message waiting times before sending, independent at each node, bearing in mind however that the packet transmission delay of course remains the same at each node for links with identical bit rates.

Before its queueing for transmission on the outgoing link, an incoming packet must be taken into account by the processor (input scheduling delay, internal routing, classification, etc.). These times are relatively short and, in our modelling, we consider them to be constant. Upon transmission, the packet may enter into competition with other packets of different sizes corresponding to different services that do however have the same priority. In the case of priority services, with important real time constraints, such as voice on IP, the node transfer delay will be modelled by the sum of a constant processing delay, a wait at an M/G/1 type server, and a packet transmission delay.

10.3.3.1. *Calculation of transfer delay*

With regard to transfer delays, the standards specify mean values and 95% values. The following evaluation method is therefore adopted: the holding time moments at each node, mean and variance, are calculated independently. By summation (independent variables), we obtain the mean characteristics and global variances. Finally we deduce from this by adjustment to the gamma (or Gauss) law the 95% value (see Chapters 4 and 7).

It will therefore be necessary to summate the time spent in the access gateways and the time spent in the routers.

Transfer delay at accesses

In the outgoing direction, this delay is the resultant of any transcoding delay, and the packeting delay. If we take the example of the Voice over IP service (VoIP), we would have the G711 (64 kbit/s) coding of the voice changed to G729 (8 kbit/s), to reduce the bit rate on the network. The voice samples are then put into packets. If we assume we were to remain in G711, in this case the voice samples are grouped into packets every 20 ms, thus leading to packets of 200 bytes (1 voice sample of 1 byte every 125 µs, grouped in packets every 20 ms, i.e. 160 bytes per packet plus 40 bytes of header). Associated with this packeting time, we should add a processing delay (< 1 ms), that we however consider to be negligible. We must finally allow for an additional processing delay corresponding to the acquisition and routing of the packets, and also a sending and transmission waiting time for the packet on the output link to the network. In the interest of simplicity, in our example we will consider such delays to be identical to those of the routers.

On arrival, we find the same operations reversed, but with an additional operation, "dejittering", i.e. jitter cancellation. As we will see later, the packets of the same call do not all take the same time to be transferred through the network. To comply with real time service constraints (one voice sample to be delivered to the user every 125 μs), compensation is made for the jitter by storing the packets in a buffer for a sufficient duration which is to be determined below. This fixed delay will therefore need to be added to the total transfer delay.

Router transfer delay

As already indicated, this transfer delay will consist of a constant processing delay, a wait in the sending queue, and the transmission delay on the link.

Transfer delay on the network

In front of the output link, at the access and in the routers, with a hypothesis of exponential packet length, and thus the M/M/1 model at each node, we obtain for the total mean wait and its variance:

$$W = \sum_{i}^{N} \frac{\rho_i}{1-\rho_i} \frac{1}{\mu_i} \tag{10-7}$$

and:

$$\sigma_W = \sqrt{\sum_{i}^{N} \frac{2\rho_i - \rho_i^2}{(1-\rho_i)^2} \frac{1}{\mu_i}} \tag{10-8}$$

We deduce from this by adjustment to the gamma law (or using Martin's formula, see Chapter 7), the 95% value:

$$W_{95\%} = W + 2\sigma_W \tag{10-9}$$

To these delays must be added processing delays, transmission delays and propagation delays (which depend on the length of the links), as previously described.

Let us assume a network made up of N transmission links with identical bit rates (including the part concerned by the access gateway), and evaluate the transfer delay for a VoIP packet.

Let us denote as c the processing delay (packet acquisition, routing, etc.) at each node, $1/\mu$ the average sending delay for packets on the link at each node, τ the sending delay of the voice packet and P the total propagation delay.

We will thus have for the transfer delay:

$$T_{mean} = t_{packeting} + (N+1)c + N.t_{meanwait(\mu)} + N.\tau + P + t_{dejittering}$$

or:

$$T_{mean} = t_{packeting} + (N+1)c + W_{(\mu)} + N.\tau + P + t_{dejittering}, \text{ and} \tag{10-10}$$

$$T_{95\%} = t_{packeting} + (N+1)c + W_{(\mu)} + 2\ \sigma_{(w)} + N.\tau + P + t_{dejittering} \tag{10-11}$$

The following is an example of values, in the case of VoIP with G711 coding. Let us consider the case of a national call. Each packet passes through 2 accesses plus 9 routers and covers a total distance of 1000 km with a propagation delay of 5 ns per metre. The processing delay of a packet in each node is 50 μs. A reference VoIP packet has a length of 200 bytes. All the links are at least at 155 Mbit/s and have loads of 60% in the most unfavourable case.

We assume a mean length for all the real time packets processed by the link of 500 bytes. We thus obtain:

$t_{packeting} = 20$ ms

$c = 50$ μs

$1/\mu = 25.8$ μs

$\tau = 10.32$ μs

$P = 5$ ms

$$W = 10\frac{0.8}{1-0.8} \times 25.8\,\mu s \approx 1\,ms$$

$$\sigma_W = \sqrt{10 \times \frac{2 \times 0.8 - 0.64}{(0.2)^2}} \times 25.8 \approx 400\mu s$$

We arbitrarily adopt a dejittering delay of 20 ms (we will estimate the probable value for our network a little later). And thus:

T_{men} = 20 *ms* + 11 × 0.05 *ms* + 1 *ms* + 10 × 0.01 *ms* + 5 *ms* + 20 *ms* = 46.65 *ms*

$T_{95\%}$ = 20 *ms* + 11 × 0.05 *ms* + 1 *ms* + 2 × 0.4 *ms* + 10 × 0.01 *ms* +5 *ms* + 20 *ms* = 47.45 *ms*

This calculation example, although based on arbitrary values, clearly shows that the impact of transmission through the IP network remains negligible when compared with packeting and dejittering delays, provided that the network is equipped with links with sufficiently high bit rates, such as STM1.

Finally it may be noted that these values easily meet the requirements of the standards ($\overline{T} < 100ms$, see the packet performance target specification at the beginning of this chapter). On the other hand, these values are well outside the requirements of Q.551: for IP transmission associated with analogue telephone equipments the use of echo cancellers is therefore essential.

Let us now study the variation of the delay.

10.3.3.2. *Calculation of transfer delay variation*

In this case the standards specify values with very low probabilities (e.g. 10^{-3}, 10^{-4}). Adjustment by the first two moments is thus no longer sufficiently precise. To solve our problem, we directly calculate the distribution of transfer delay probability. We will approximate the variation sought to the transfer delay value at 10^{-x}, minus the minimum transfer delay for the reference packet or message. This is a pessimistic evaluation as it amounts to considering that the previous message or packet was not subject to any delay.

Based on the assumption of independence, the distribution of the total delay is given by the convolution product of the distributions at each node. In our example, the service time is assumed to be exponential. We then have to calculate the convolution product of M/M/1 queues. As shown in Chapter 7, the waiting time probability density at a server of this type is:

$$w(t) = (1-\rho)u_0(t) + \rho\mu(1-\rho)e^{-\mu(1-\rho)t} \tag{10-12}$$

The convolution product for *n* queues such that $f_i(t) = A_i e^{-a_i t}$ $(\mathrm{i}=1,2,..\mathrm{n})$, is easily obtained by Laplace transforms:

$$F_{(n)}*(s) = F_1*(s).F_2*(s)...F_n*(s) = \frac{A_1A_2...A_n}{(s+a_1)(s+a_2)...(s+a_n)}$$

In the case of n identical queues, this of course gives:

$$F_{(n)}*(s) = \frac{A^n}{(s+a)^n}$$

We recognise the result already obtained for the *Erlang n* law, of which the original function is: $f_{(n)}(t) = \frac{A^n t^{n-1}}{(n-1)!} e^{-at}$.

Which gives us, applied to $w(t)$, and taking account of the probability at the origin (probability of not waiting):

$$W = \sum_{k=1}^{N} \binom{N}{k} (1-\rho)^{N-k} \rho^k \frac{[\mu(1-\rho)t]^{k-1}}{(k-1)!} \mu(1-\rho) e^{-\mu(1-\rho)t}$$

or: $W = (1-\rho)^N e^{-\mu(1-\rho)t} \sum_{k=1}^{N} \binom{N}{k} \lambda^k \frac{t^{k-1}}{(k-1)!}$ (10-13)

and the distribution function is:

$$W(<t) = 1 - \frac{1}{\mu}(1-\rho)^{N-1} e^{-\mu(1-\rho)t} \sum_{k=1}^{N} \binom{N}{k} \lambda^k \frac{t^{k-1}}{(k-1)!} \qquad (10\text{-}14)$$

The calculations therefore give for $N = 10$ and a probability of 10^{-3}, $t = 30 \text{x} 1/\mu$. Hence:

$$W(<1ms) \approx 10^{-3}$$

This value gives the variation of the delay as it is the only variable part of the transfer delay.

Once again, we find a very low variation, again due to the high bit rate of the transmission links.

These orders of magnitude remain true even in the case of lower bit rate links between *access gateways* and the network, such as E3 links at 34 Mbit/s. Indeed, in that case, we have on two links:

$1/\mu = 117.6\ \mu s$

$\tau = 47\ \mu s$

This has little impact on the overall result.

Based on this result, we could fix dejittering delays in the order of 10 ms, but we must also allow for cases in which the number of routers is far greater, and in which the number varies in the same call. We will therefore retain the initial value of 20 ms.

10.3.4. *Call set-up delay*

Let us now consider the evaluation of the end-to-end call set-up delay in the case of an NGN network. The configuration studied is represented in Figure 10.5.

The origin m*edia gateway controller* (MGC) sets up a call with the destination *media gateway controller* through the IP network, so as to connect the two local exchanges.

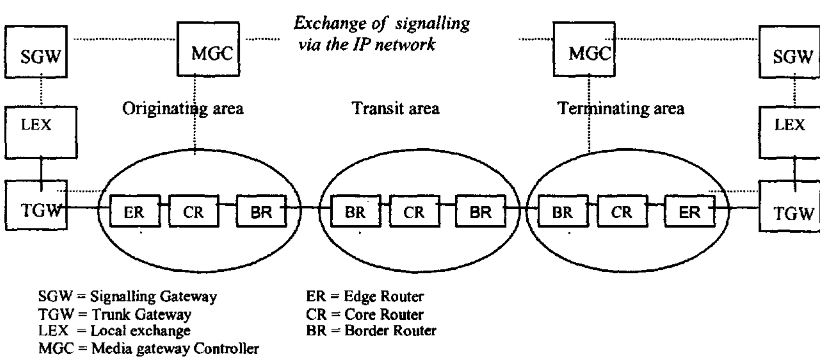


Figure 10.5. *NGN configuration*

Functioning

The outgoing *media gateway controller* (MGC) receives a call set-up message from the local exchange via the *signalling gateway* (SGW) whose role is simply to adapt the signalling.

The origin MGC sends via the IP network an IP address request to the origin *trunk gateway* (TGW), which provides it in return (ADD messages). Then the origin MGC sends the set-up request to the destination MGC (SET UP). The latter requests its IP address from the destination TGW, which in turn sends it back. The destination MGC resends the information to the origin MGC (SET UP ack), which resends it to the origin TGW (MOD). On the reply from the origin TGW, the MGC requests the set-up of the call by the destination MGC (SET UP), which then sends the IAM set-up message to the destination LEX via the TGW and the *signalling gateway*. In return, the origin LEX receives the call acceptance by the reverse path.

Figure 10.6 shows the messages exchanged and the processing carried out in the different elements of the network.

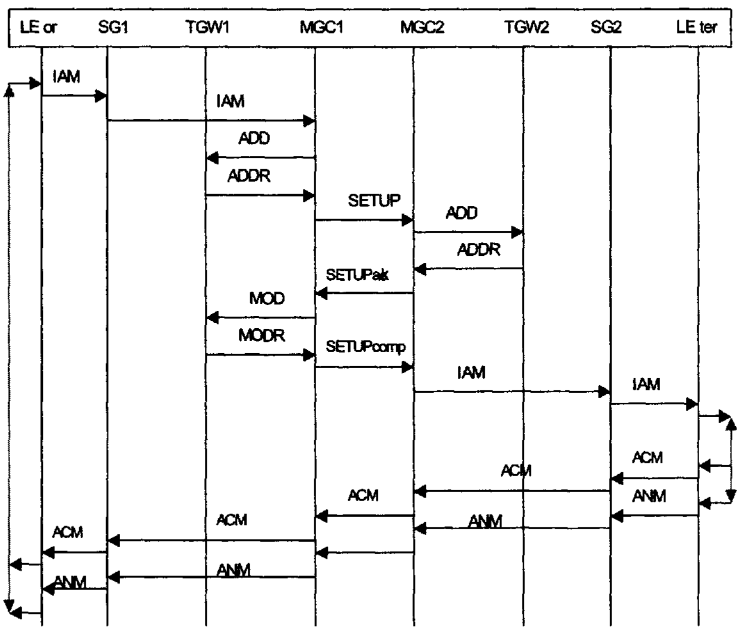


Figure 10.6. *Call set-up diagram in the NGN*

Determination of individual delays

To estimate the end-to-end set-up delay as defined by the standards, between IAM and IAM, we must refer to the values specified for the various components. If we adopt the position of the network designer, we must assemble different equipments whose behaviour we do not know in detail, but whose performances are specified by standards and by specifications.

Table 10.13 recapitulates values by way of example. These values have been established on the basis of existing recommendations, and may of course change, but also on the basis of possible time evaluations for new elements (TGW, MGC...), with new technologies as in the previous example.

Table 10.13. *Parameters and reference values for end-to-end set-up delay calculation*

	Network component	Parameter	Value	
			Mean	95%
$T_{LESETUP}$	Originating LE	Call set up delay (Table 30/Q543)	800 ms	1 200 ms
$T_{IMGC}=T_{GW}$	MGC and TGW	IAM processing (1/2Table 1/Q766)	130 ms	260 ms
$T_{LEICISD}$	Terminating LE	Incoming call indication sending delay (Table 4/Q543)	800 ms	1 200 ms
T_{SGW}	SGW	Message transfer delay (1/2Table 4/Q706)	20 ms	40 ms
T_{2MGC}	Transit exchanges	ANS processing (1/2Table 1/Q766)	80 ms	160 ms
$T_{LETRANS}$	Originating/ Terminating LE	Signalling transfer delay (Table 28/Q543)	350 ms	700 ms
T_{ROUTER}	Router	Packet Transmission delay (operator)	1 ms	3 ms
$T_{LINKTRANS}$	Transmission links	Transmission delay (Q.41)	5 µs/km	–

From these values, by applying as before the properties of the gamma law, (see Chapter 7), we can deduce the variances:

$\sigma = (t_{95\%} - t_{MEAN})/2$

and:

$\sigma_{LESETUP} = 200\ ms$, $\sigma_{1MGC} = 65\ ms$, $\sigma_{LEICISD} = 200\ ms$, $\sigma_{SGW} = 10\ ms$, $\sigma_{2MGC} = 40\ ms$,

$\sigma_{LETRANS} = 175\ ms$, $\sigma_{ROUTER} = 1\ ms$

Thus, in view of the assumption of independence between the various elements of the network, and bearing in mind that we are interested in mean values and the variance of the delay sought, we can estimate the characteristics of the global response time by the sum of the means and individual variances.

Determination of global set-up delay

According to the set-up diagram, we have:

$$\overline{T}_{SETUP} = \overline{T}_{LESETUP} + 4\overline{T}_{SGW} + 7\overline{T}_{1MGC} + 2\overline{T}_{2MGC} + 3\overline{T}_{TGW} + \overline{T}_{LEICISD} + \overline{T}_{LETRANS} + N\overline{T}_{ROUTERS} + L\overline{T}_{LINKTRANS}$$

and:

$$\sigma^2_{SETUP} = \sigma^2_{LESETUP} + 4\sigma^2_{SGW} + 7\sigma^2_{1MGC} + 2\sigma^2_{2MGC} + 3\sigma^2_{TGW} + \sigma^2_{LEICISD} + \sigma^2_{LETRANS} + N\sigma^2_{ROUTERS}$$

where N is the number of transfers through the routers and L the total length of the transmission links over which the messages pass.

Let us take as our example a distance of 500 km between the two MGC, 200 km between SGW and MGC, 200 km between MGC and TGW, 6 routers between the MGC and 3 routers between TGW and MGC.

According to the set-up diagram, we thus have a total length travelled by the messages of:

$L = 11 \times 200\ \text{km} + 4 \times 500 = 4200\ \text{km}$

Similarly we find a total number of router crossings of:

$N = 4 \times 6 + 6 \times 3 = 42$

The numerical application gives us:

$\overline{T}_{SETUP} = 3.5\text{sec}, \ \sigma_{SETUP} = 0.4\text{sec}$

and thus: $\overline{T}_{95\%SETUP} = 4.3\text{sec}$

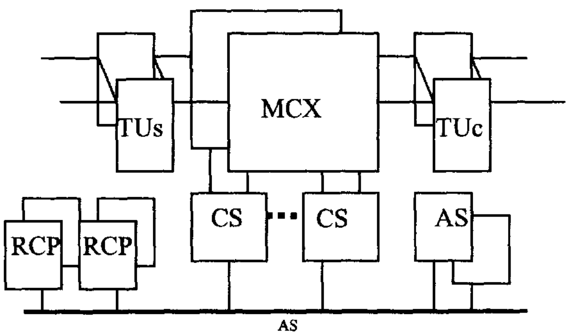
It will be noted that this network complies with standard objectives (see Chapter 2).

10.3.5. *Dependability*

We will now evaluate the performances of the project from the viewpoint of dependability. We will evaluate in succession the maintenance load, total unavailability and the probability of premature release of a set-up call.

Let us recall the main targets: total unavailability < 5.7 10^{-6} (3 min. per year); premature release < 2 10^{-5}.

The simplified diagram in Figure 10.7 represents, from the viewpoint of the defence units, the architecture of the project in its mobile centre application (the same system taken as an example carries out both functions: fixed centre and mobile centre).



LAN: Local inter station network
TUs : Subscriber access term. unit
CS : Control station (call proc, trad...)
AS : Auxiliary stations (signalling...)
MCX : Switching matrix
TUc : Circuits termination units
RCP : Mobile server (radio control point)

Figure 10.7. *MSC (Mobile Switching Center) architecture*

The data forming the basis for our evaluations are the redundancy types and the predicted failure rates of the various equipments.

Redundancies and failure rates

LAN: duplicated and self-healing.
MCX: the switching network is duplicated, redundancy is active, calls are established in parallel on both matrices, one matrix (on its own) can handle all the traffic.
TU: redundancy type $N+k$, i.e. there are k TU on standby for N TU of connection.
CS: the control stations are in $N+1$ redundancy, of the *stand-by* type.
RCP: each mobile server is duplicated, in individual redundancy of *stand-by* type.
AS: the auxiliary stations are in $N+1$ redundancy of the *stand-by* type.

The configuration studied, corresponds to a centre with about 500,000 subscribers.

The number and the hardware failure rate (or software failure rate if applicable), of the equipments of each type are as follows (the following rule is adopted: $\lambda_{\text{software}} = \lambda_{\text{hardware}}$):

MCX: there are two branches composed of several boards. These planes may be equipped to a greater or lesser extent depending on the capacity desired. In our case, the equipment leads to a system equivalent to two matrices, each with a failure rate of $\lambda_{MCX} = 5\ 10^{-6}$/h.

TUs: there is only one very high capacity equipment ($N = 1$), plus a standby equipment, as these equipments are placed behind the mobile subscriber access network (BSC, UTRAN)), having already concentrated the traffic on very high capacity links. $\lambda_{TUs} = 10\ 10^{-6}$/h.

TUc: there is a large number of equipments of this type ($N = 20$) for the network directions are numerous and the modularity is great (STM1 link connection). $\lambda_{TUc} = 5\ 10^{-6}$/h. There are 3 standby equipments for all the other 20 equipments.

CS: the modularity of the control stations enables adaptation to the various processing capacity needs. For the configuration studied here, the number of control stations is $N = 2$, plus one standby, each one with a failure rate of $\lambda_{CS} = 50\ 10^{-6}$/h. (A sizing example will be considered later).

RCP: there are two RCP servers (with standby). $\lambda_{RCP} = 60\ 10^{-6}$/h.

AS: the auxiliary stations are organised in modular mode, as are the control stations. Here we have $N = 5$, plus a standby station, and $\lambda_{AS} = 50\ 10^{-6}$/h.

We do not take into account the impact of the LAN as its failure rate is negligible compared with those of the other equipments.

Maintenance load

The maintenance load can already be evaluated. Only hardware failure rates are involved here: it is considered that software failures are "repaired" by very rapid restarts of the station (see Chapter 6). We have the global failure rate:

$\lambda_G = 2\ \lambda_{MCX} + 2\ \lambda_{TUs} + 23\ \lambda_{TUc} + 3\ \lambda_{CS} + 4\ \lambda_{RCP} + 6\ \lambda_{AS} = 2 \times 5\ 10^{-6}/\text{h} + 2 \times 10\ 10^{-6}/\text{h} + 23 \times 5\ 10^{-6}/\text{h} + 3 \times 50\ 10^{-6}/\text{h} + 4 \times 60\ 10^{-6}/\text{h} + 6 \times 50\ 10^{-6}/\text{h}$,

and thus $\lambda_G = 835\ 10^{-6}$/h, i.e. 7.3 interventions per year on average.

In this matter, in accordance with the conditions specified in the objectives, we are assuming an intervention policy which makes a distinction between *deferred* intervention (the standby equipments are used but there is not yet any loss of service) and *immediate* or *urgent* intervention (which is necessary when there is a loss of service).

We will denote the deferred intervention rate as μ_d ($\mu_d = 8\ 10^{-2}$/h, corresponding to a mean time for travel plus repair of 12.5 hours) and the immediate intervention rate μ_i ($\mu_i = 0.25$/hour, corresponding to a time of 4 hours).

Total unavailability

All the functions previously identified are in series from a reliability viewpoint. We will examine the reliability diagram, shown in Figure 10.8, in more detail below.

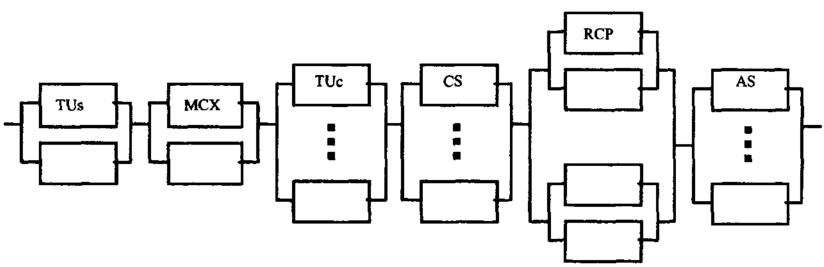


Figure 10.8. *Reliability diagram*

We must first state what we mean by total failure, particularly with relation to the concept of degraded mode.

In the case of control stations (CS), the failure of a single station does not cause any deterioration of the service; this is because the processing capacity is maintained by the replacement of the failed station by the standby station. However, the simultaneous failure of two control stations means that it is no longer possible to handle all the traffic; thanks to the standby station, half of the traffic can be handled. This is degraded mode. However, if the three control stations fail simultaneously, unavailability is total.

In the case of RCP, only the simultaneous failure of the two RCP will cause total unavailability. The failure of a single RCP means that half of the traffic can still be handled. Furthermore, as the RCP is itself a machine with standby processor, the failure of an RCP means in fact the simultaneous failure of both of its elements.

In the case of auxiliary stations (AS), depending on the number of stations that have simultaneously failed, the states will be more or less degraded. Because of the number of stations and standbys, total unavailability will be considered to be negligible.

Similarly, the loss of some network connection (termination) units does not lead to an unavailability sufficient to be considered to be total, because of the large number of equipments of this type and redundancy. They will therefore be considered to be negligible.

As the other equipments are duplicated, total unavailability will occur in the event of simultaneous failure of the two equipments.

Here also, only hardware failures are significant (software failures are repaired by very rapid restarts).

We therefore have the following expression:

$$I = 2\frac{\lambda_{TUs}^{\ 2}}{\mu_d\mu_i} + 2\frac{\lambda_{MCX}^{\ 2}}{\mu_d\mu_i} + 3x2\frac{\lambda_{CS}^{\ 3}}{\mu_d\mu_i^{\ 2}} + 4x3x2\frac{\lambda_{RCP}^{\ 4}}{\mu_d\mu_i^{\ 3}}$$

We will not detail here the Markov graphs, the reader has only to refer to Chapter 6 to solve these simple cases of redundancy. We should state however that, for RCP, we consider that an intervention is urgent as soon as two equipments have failed, wherever they are (on the same RCP or on different RCP).

The numerical application gives:

$$I = 1\times10^{-8} + 2.5\times10^{-9} + 1.5\times10^{-10} + 2.4\times10^{-13} \approx 1.3\times10^{-8}$$

We have given the details of the calculations as they reflect the impact of each function. The overwhelming impact of the access interfaces shows the importance of ensuring a high degree of dependability for elements that have a strong concentrating effect. In this case it would also perhaps be advisable to carry out a more detailed modelling of these equipments. It is also clear that as soon as modularity is large, total unavailability becomes low. Clearly, we can conclude that this architecture easily enables the meeting of the objective set of 5.7 10^{-6}, but note that we have considered the rate of "severe" software failures as negligible (see Chapter 6). This can only be guaranteed if the product has been well tested. Hence the importance of the test phase that we will consider a little later.

Premature release

Let us assume that a call has been set up and analyse the failure events that could lead to a premature release of the call.

CS and AS: In the event of failure of this type of station, all the calls set up are lost when there is a switchover to the standby station (loss of context). The fresh traffic is of course immediately taken over by the standby station. We assume that the call involves two CS and two AS (which is the most usual case with the "half call" approach). Furthermore, this time we must take into account the *software failure rate*.

The occurrence rate of this event *for a given call* is thus 2 × ($2\lambda_{CS} + 2\lambda_{AS}$).

On the other hand, for the RCP server, it is assumed that there is no call release: the contexts are saved (in real time or periodically) on the RCP on *standby*, which is easy in the case of single redundancy.

Similarly, for the TU and switching matrices, there is no loss. It is only a matter of physical transport and the system is such that the switchover to the standby TU is instantaneous, and (as we pointed out for the switching network) the calls are set up on both branches.

Finally, call release only occurs if there is traffic to handle. We will adopt the pessimistic assumption of the equivalent of 10 loaded hours per 24 hours.

Taking as the reference an average call time τ of 100 s, we thus obtain the probability of release of a set-up call, by the relation:

$$P_r = \frac{10}{24} 2(2\lambda_{CS} + 2\lambda_{AS})\tau = 5\ 10^{-6}$$

The objective of 2 10^{-5} has thus been met.

The study demonstrates the importance of this criterion in the choice of architecture (software and hardware).

10.4. On-load tests

The global performances of the project will be validated during the integration phase. We then have a complete configuration of the equipments and means to simulate the reference environments to be tested. The *list of tests* to perform and the definition of the configuration(s) to be tested will have been defined at an earlier stage. The test facilities will enable the generation of the flow of messages or packets, corresponding for example to the calls to be set up, which may come from the network or from the accesses (gateways, subscriber connection units). The aim is then to minimise the hardware used, and to make the tests feasible with very high traffic volumes. Traffic *simulators* are therefore developed, replacing subscribers, circuits, and signalling and data links, etc. These may be autonomous external devices, or software developed on equipments in the system. To properly clarify problems when large systems are validated, it is advisable to break the tests down into one preliminary phase consisting of the validation of certain subsystems, and then a phase of "lighter" global tests.

To make this clearer, let us consider the concrete case of a network constructed around accesses, switches, routers and IN servers. First, each of these entities is validated under load and overload conditions by simulating (generating by simulator) its traffic environment in respect of the reference traffic mix, before the validation of the correct functioning of the whole interconnected system (the whole network). For example, to test a switch, we will simulate the messages exchanged with the accesses, with the server, etc. For this purpose, we will have developed, at access or server level, simulation software that carries out only elementary chaining of the messages exchanged at the interfaces, as described below. Thus all the processing to be performed by the switch will be performed, including connections, the sending of ringing, etc. But as there is no terminal really connected, there will be neither reception of ringing nor voice samples exchanged. These functions will be subjected to separate tests. For this purpose, there will also be available a limited number of actual accesses and terminals to carry out detailed user-viewpoint quality of service measurements, and standard subscriber simulators (approved by the operators) which are faithful replicas of the subscriber terminals.

The principle of the *traffic simulators* is as follows. Once again, reference is made to the call set-up and release message diagrams to define the scenarios or message sequences exchanged at an *external interface* and corresponding to the service call type (outgoing, incoming, etc) to be tested. These *lists* of messages, requests and responses are recorded in the simulator as *scenarios*. The scenarios are run randomly, in proportions that correspond to the call mix to be tested, as defined in the objectives document. Note here the importance of the random nature of this call request generation. It is necessary to first reproduce normal operating conditions, in order to distinguish "normal" congestion phenomena from specific phenomena relating to malfunctions. The precision required here on the random aspects is however far less great than in the case of a detailed simulation, such as that of a queue, and it is therefore sufficient to use simple algorithms (see Chapter 8). Once the initial call requests have been launched, the exchange process between the simulator and the device tested is chained, call by call. Each message from the system tested generates another message from the simulator to the system, and so on, until the end of the call which itself generates the launch of another call, etc. It is the rate of regeneration of these call requests which will define the volume of traffic offered to the system (see example below).

It will therefore be necessary to have available, in addition, standard subscriber equipment or circuit simulators approved by the operators. This is in order to generate a certain (low) proportion of actual traffic, and thus measure in detail the quality of service perceived by the user, such as set-up delay, good reception of ringing and tones, and the quality of transmission in the conversation phase, etc. It will also of course be necessary to have available real terminals to carry out test calls, e.g. in severe overload conditions, to test the capacity to handle priority calls. Finally, by taking up a position at the standardised interfaces (e.g. no. 7 signalling channel interface), it will be easy to check the exactness of simulation, as the message sequences exchanged by the internal simulators and by the standard external simulators must be exactly the same.

Test phases

Distinctions may be drawn between three major test phases: traffic performance tests, overload tests and dependability tests. We will now describe these phases.

10.4.1. *Traffic performance tests*

During this phase, the system will be progressively subjected to traffic with two important levels, Load A traffic and Load B traffic, in accordance with the objectives.

Compliance with the call mix will be verified and measurements will be carried out of traffic handled, response times and processor loads. Delays objectives such those specified in the standards will be verified.

The system observation functions will be implemented to obtain these results (number of calls of different types presented, number of calls rejected, response times). Use will also be made of the observations of standard external simulators, call counters, and fault counters, in order to confirm the results. Thanks to the observation of processor load, it will be possible to validate and if necessary refine the modelling and the dimensioning rules.

In many cases this phase will be important for the development of the system, not just in terms of processing capacities, but also from a functional viewpoint. This is because during the tests the following will be identified: processing delays and exchanges between machines to be optimised, and also cases of conflicts of access to common resources, or defence faults following a queue or time out overflow.

This explains the importance of the work done in advance during modelling. The capacity of the system to handle the forecast traffic in the stipulated project time frame (according to the roadmap) will be all the greater if the evaluations of the loads and real-time behaviour of processors and hardware (such as ASIC) were accurate. Furthermore, the more detailed the modelling, the easier will be the identification, comprehension and resolution of problems (we will know which parameters are to be measured and what "normal" values should be).

10.4.2. *Overload tests*

The aim here is to subject the system to load levels that are exceptional, but nonetheless likely to occur in real life.

It is worth recalling the basic phenomena: exceptional events such as TV game shows, catastrophes, special days such as New Year's Day, and the impact of indiscriminate call renewal that occurs when the level of congestion is noticeable. It is not of course possible to simulate call arrivals in detail under such circumstances. Fortunately, however, as proven by observation, for a large switch for example, the traffic presented can be assimilated to Poissonian traffic of very large volume, which also lasts for a very long time.

Furthermore, observation also shows that a distinction must be drawn between incoming overloads and outgoing overloads. For example, for a TV game show or a disaster, calls will arrive in excessively high numbers at a given point in the network, but from all directions. Above all, an incoming overload will arise. On the other hand, during a great gathering (the Soccer World Cup, or if people are blocked somewhere by adverse weather conditions, e.g. snow on a ring), all the mobiles will be activated simultaneously, creating a rush of outgoing traffic. Finally, on particular days of cultural or national importance (such as the start and end of Ramadan, New Year's Day), it is the whole network (both outgoing and incoming) that will be loaded. We will look at this situation in more detail a little further on.

In each case, we will draw the system *response curve* (traffic handled as a function of traffic offered), and a check will be made to ensure that the pattern required by the standards (see Chapter 2, Figure 2.6), and the validity of the priority strategy between call rejections are complied with.

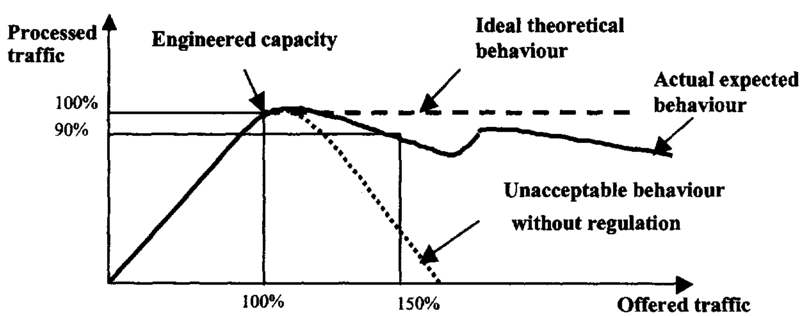


Figure 10.9. *Traffic handling under overload conditions*

This will primarily be a phase of regulation threshold adjustment on all the elements of the system. The aim is to detect an overload at an early stage so as to preserve the satisfactory handling of calls already accepted: in other words, the queues for the processing of subsequent messages, like the resources associated with the transport of user level information corresponding to sessions in progress, must never be saturated. The call processing delay and processor load measurements must also be completed by measurements of call rejection costs, so as to fully validate the modelling of the system. Figure 10.9 sets out the type of response curve that is expected (see Chapters 2 and 9).

10.4.3. *Dependability tests*

While the system is handling Load A and Load B traffic, faults are injected on to the equipments, or units will simply be set to an out of order state by an operator command.

The expected reactions of the defence mechanisms will then be verified: failure detection, isolation of the faulty element, switchover to standby and reconfiguration. Measurements will of course be taken of any losses of calls and of unavailability times. The values obtained will enable the confirming and refinement of the dependability model parameters.

Besides the measurement of the reaction times, the role of failure injection is also to verify the efficiency of the mechanisms used to *locate* the faulty element (board), as, normally, in complex systems any hardware or software malfunction is detected very rapidly at application level and leads to the setting of machines to the "out of order" state.

Whereas making settings by commands are relatively simple to carry out, as they result from normal operating procedures, the injection of failures is on the other hand more complicated. Generally speaking, boards will be put on a connector and points will be grounded. It is here that sampling methods must be applied: the points affected are drawn at random after weighting for the predicted failure rates of the components. But in this respect, allowance should also be made not only of the probability of failure of an element, but also of the impact of its failure on quality of service. For example, any failure of a common element or part of an element that could lead to total non-availability must in all cases be tested (e.g. a time base, certain bus elements).

Here again we see all the importance of having first modelled the system from the dependability viewpoint, to better identify the parameters and elements to be tested, and to enable a better judgment of the results obtained, compared with the objectives.

10.4.4. *General organisation of tests*

The diagrams below sum up the main phases and the expected results.

This is only an example of a sequence of operations, and the relative times may vary, depending on the difficulties encountered.

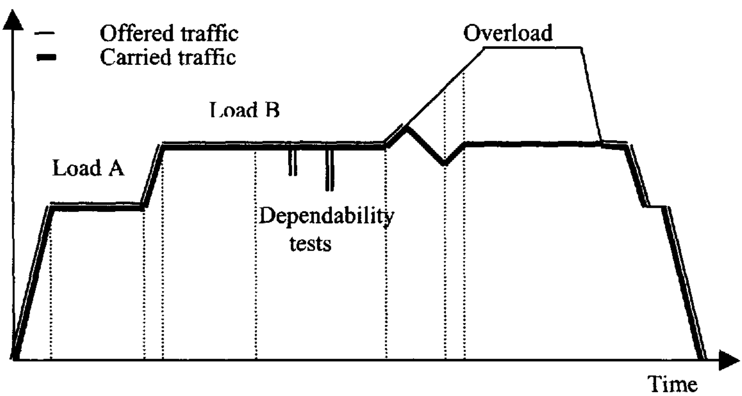


Figure 10.10. *Load test sequence*

In an overload phase, it will be noted that it is necessary to distinguish between several stages, in accordance with the model (see Chapter 9). In the first stage, the capacity of the system to handle priorities may be observed: for example emergency calls (fire service, etc.) and incoming calls. Then, as the overload continues, tests are performed on its capacity to maintain the level of traffic handled at an acceptable level, with a very heavy overload over a very long period of time.

It should also be noted that it is necessary to verify the return to normal operation, in reasonable delays, after each "stress" phase, defence test or overload.

Traffic generation example

We present a method which, although it introduces a certain bias to the Poissonian nature of the traffic, has the advantage of very great simplicity in implementation, while giving quite significant results.

We wish to generate traffic of outgoing, internal, incoming and transit calls complying with the high efficiency mix presented in the objectives and set out below in simplified form.

Table 10.14. *Traffic mix for load test*

High efficiency mix				
Type of call	Internal	Outgoing	Incoming	Transit
% Flow	*20*	*35*	*35*	*10*
% Efficient	58	58	70	70
% No reply	15	15	18	18
% Busy	10	10	12	12
% Incorrect dialling	17	17	0	0

For this purpose, we have available one (or more) simulator(s) managing on the one hand couples of calling subscriber and called subscriber to generate internal outgoing and incoming traffic, and on the other hand couples of circuits to simulate transit traffic.

The diagram in Figure 10.11 presents the principle of traffic generation organisation.

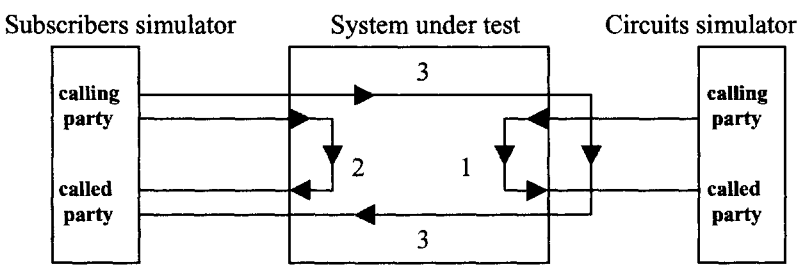


Figure 10.11. *Organisation of traffic generation*

The fact of working by couples of the same nature (subscribers or circuits) introduces a certain correlation between outgoing and incoming calls, but which in normal operating conditions may be neglected. On the other hand, for overload tests, allowance must be made for this (as the rejection of an outgoing call also automatically suppresses an incoming call), and in some cases use more complex organisations, with the management of subscriber plus circuit couples.

For each couple, the simulator program will first draw random numbers according to an *exponential law* with a parameter $\theta = 1/\lambda$ (see Chapter 8) and defining the inter-call delay. This value will define the arrival rate of the calls offered to the system. As soon as a call is initiated, the initiation date of the next call is calculated. For example, if the intention is to generate a traffic of 500 calls per second (1.8 MBHCA), the parameter θ will be fixed at 2 ms. Each time a call is made, the call type is drawn at random, in accordance with a *uniform law* corresponding to the reference mix. To simplify implementation and provide a better reflection of reality, the draw is carried out in several stages. For our mix, we will therefore have the following operations:

1) draw at random the traffic nature of the call (internal call or outgoing-incoming call): the random number drawn between 0 and 1 is compared to the values 20/55 and 35/55;

2) draw at random the efficiency type of the call (efficient call, busy, no reply, etc.): for example, for an internal call, a new random number is compared to the values 58/100, 15/100, 10/100, 17/100.

In the same way, a decision may be taken about the signalling type, and the existence and the nature of an IN service, etc.

Then the call set-up request message (IAM, see set-up diagram) is initiated in the system with the number of the called party (which corresponds to that of the other element in the couple). As the appropriate routing and forwarding information have been prepared in advance in the system (translation tables, routing, etc.), the incoming call message (*set up*) returns to the simulator, which from the number requested, recognises the couple concerned and its characteristics (busy, no reply, efficient, etc.) and thus sends the appropriate response back to the system. This continues in accordance with the set-up and call release diagrams. Defence mechanisms (time out, etc.) enable the release of the couple in the event of absent or erroneous messages.

10.5. Dimensioning

The definition of the dimensioning rules for the equipments and the network is a key activity from an economic viewpoint. The aim is to determine the optimum number of equipments to be installed to guarantee that the service is supplied with the required quality. The aim in this phase of activity is to define the *basic rules* used for sizing, bearing in mind that specialist teams will translate these basic rules into calculation programmes for numbers of boards, racks, etc., and price calculations.

The first rules are supplied at the modelling phase, so that the products can be launched on the markets (this demonstrates the importance of the modelling, which must ensure that the risk of error is minimised). The rules are then fine-tuned in the light of the performance tests. During these tests, the *"costs"* in UC load (c) of the various types of calls and processing, are refined, as well as the maximum admissible *utilisation factors* (i.e. the maximum UC load or *capacity* C,) compatible with the response times required. The dimensioning rules then consist of calculating the number of equipments necessary to handle a given traffic, bearing in mind that the capacity of each equipment is given by the basic relation:

$$N(call/s) = \frac{C(\max load)}{c(load/call/s)} \qquad (10\text{-}15)$$

A distinction should be drawn between two main types of resources which can be sized: control plane processors and user plane transport resources. Two examples of traffic-related applications are considered below: calculation of the number of processors necessary to handle calls in a mobile network and calculation of the throughput and number of links necessary to handle IP multimedia traffic. We then give an example of calculating the size of the maintenance stock (i.e. the number of spare parts needed).

10.5.1. *Dimensioning of the control of a mobile centre*

The aim here is primarily to describe the approach, the use of the Load A and Load B concepts, and in particular the use of the data attached to the site (operator domain), and that attached to the constructor equipments (equipment vendor domain).

We will consider the case of a "mobile" centre and more precisely the dimensioning of a CSCN in a GSM environment. Let recall that the CSCN (*circuit switch core network*) is the control part of the *core network* that connects mobile subscribers via the BSS (radio stations) and sets up connections with the PSTN (*public switched telephone network*).

In accordance with the functional breakdown set out in Chapter 1, we will consider this centre composed of two main physical entities: the mobile server, or RCP (*radio control point*) and the switching centre or SSP (*service switching point*).

The mobile server (RCP) acts as the interface with the radio stations that connect the mobile subscribers and is in charge of the mobility functions (*hand over*, etc.). This is the RCF function (*radio control function*) with which it combines the VLR function (*visitor location register*).

The SSP carries out the classical switching functions: call set-up, call release, call to IN server (the RCP can also make calls to the IN server), etc., and thus establishes the connections with the fixed network to reach other subscribers (fixed or mobile).

This is of course only an architecture hypothesis, which we have chosen as it enables distinctions to be drawn between processing without making the explanation excessively complicated.

The first stage consists of entering the *traffic environment* data, in order to evaluate the average processing cost per call per machine. These are "site" data (usually provided by the operator).

For the site in question, we have for example 500,000 mobile subscribers (MS) generating an average traffic per subscriber of 0.025 erlangs (0.77 OC plus TC per hour, with an average duration of 120s), at *Load A*. In view of the subscriber and network characteristics, the traffic mix is as follows (see the paragraph on objectives and environments, with a simplified mix as the aim is simply to demonstrate the approach).

Table 10.15. *Site mix*

<table>
<tr><th colspan="8">Site mix</th></tr>
<tr><td>Call type</td><td>OC</td><td colspan="2">TC</td><td>GW</td><td>IN</td><td colspan="2">OSMS/TSMS</td></tr>
<tr><td>Number of calls/hour/MS</td><td>0.57</td><td colspan="2">0.2</td><td>0.25</td><td>0.33</td><td colspan="2">0.08</td></tr>
<tr><td>Mobility</td><td colspan="2">HO intra</td><td colspan="2">HO inter</td><td colspan="2">LR intra</td><td>LR inter</td></tr>
<tr><td>Events /hour/MS</td><td colspan="2">0.08</td><td colspan="2">0.04</td><td colspan="2">0.8</td><td>0.4</td></tr>
</table>

It is important to note at this point the considerable difference between the total number of *events* (OC, TC, GW, HO, LR) to be processed and the number of *calls* (OC, TC) seen from the subscriber viewpoint. This is a key characteristic of mobile telephony.

The second stage consists of associating with these call types the *"machine" costs* which are those of the *manufacturer data*. These costs are usually expressed in machine time.

We thus have for example:

Table 10.16. *Machine costs (processing costs)*

RCP costs					
Call type	OC	TC	GW	IN	OSMS/TSMS
RCP processor cost (seconds)	*0.0052*	*0.0063*	*0.0025*	*0.0018*	*0.0036*
Mobility	HO intra	HO inter	LR intra	LR inter	
Events /hour/MS	*0.002*	*0.0045*	*0.002*	*0.0045*	

SSP costs					
Call type	OC	TC	GW	IN	OSMS/TSMS
SSP processor cost (seconds)	*0.0128*	*0.0128*	*0.0128*	*0.026*	*0*
Mobility	HO intra	HO inter	LR intra	LR inter	
Events /hour/MS	*0.0075*	*0.023*	*0*	*0*	

The costs are different of course depending on the machine, and some types of processing are not available on certain machines.

The third stage consists of calculating the average cost per subscriber on each machine.

The RCP processor cost of an "average" call of the mix is thus:

$c = 0.57 \times 0.0052 + 0.2 \times 0.0063 + 0.25 \times 0.0025 + 0.33 \times 0.0018 +$
$0.08 \times 0.0036 + 0.08 \times 0.002 + 0.04 \times 0.0045 + 0.8 \times 0.002 + 0.4 \times 0.0045 =$
0.0095 s/h/MS.

The machine must process this traffic mix in *Load B* conditions (see Chapter 2). The Load B value is "site" data. It is generally accepted in the standards that: Load B = 1.2 × Load A. Under these conditions, the machine cost per MS thus becomes:
$c = 0.0095 \times 1.2 = 0.0114$ s/h/MS.

Similarly, the SSP processor cost of an "average" call from the mix is:

$c = (0.57 + 0.2 + 0.25) \times 0.0128 + 0.33 \times 0.026 + 0.08 \times 0.0075 +$
$0.04 \times 0.023 = 0.0232$ s/h/MS.

As the machine must process this traffic mix in *Load B* conditions, the machine cost per MS becomes:

$c = 0.0232 \times 1.2 = 0.0278$ s/h/MS

The fourth stage consists of calculating the number of machines necessary.

It is necessary to first find out the *maximum load* of the machine at which quality of service can be maintained (response time, etc.). This is also *constructor data* (from the modelling). It may not be explicitly stated in the rules. It is then simply taken into account in the costs (i.e. the cost is given with reference to 100% maximum load), but the drawback of this is that it makes it difficult to interpret load measurements during tests.

Let us assume that the *maximum load* of a machine, RCP or SSP, at which quality of service requirements are met (response time, etc.) is C = 80%.

We can now calculate the equipment required.

RCP case

In our architecture hypothesis, the machine is of the server type. We merely need to determine the number of servers of this type that are necessary.

The *maximum Load B capacity* of an RCP machine is $(0.8 \times 3600)/0.0114$ ~252,600 MS.

As our centre must handle the traffic of 500,000 subscribers (MS), we must equip it with 2 RCP.

We have assumed that redundancy problems are resolved in each individual machine (each RCP machine is itself a station with a processor and standby).

Let us now consider the SSP function

The approach is the same, but this time the machine structure is different. The structure consists of a set of multiprocessor stations, of modular capacity, managed by *configurations* which are more or less powerful. These configurations are *constructor data* and it is a matter of determining the right configuration, which is best adapted to the needs of the site.

The *maximum Load B capacity* of an SSP processor is $(0.8 \times 3600)/0.0278$ ~103,600 MS.

The *configurations* proposed for example are classified as *small* (S), *medium* (M), and *large* (L), depending on the number of processors with which they are equipped (see the chapter on models). These are *constructor data.* For example we have:

Table 10.17. *Configurations*

Configuration	S	M	L1	L2	L3	L4	L5	L6
n processors/station	1	3	4	4	4	4	4	4
N stations	1	2	2	3	4	5	6	7

The S configuration thus consists of 1 station, equipped with 2 processors. The M configuration consists of 2 stations equipped with 3 processors. The L6 station is equipped with 7 stations equipped with 4 processors. All the configurations are backed up on the basis of one redundancy at $N + 1$, for example, the M configuration will in fact have 2 + 1 stations, one of which is on standby.

As our centre must handle the traffic of 500,000 subscribers (MS), we must equip it with a configuration of 5 processors in all. As this is not proposed, we must therefore choose the M configuration. However, it would be advisable when making this choice to carefully reconsider the traffic hypotheses with their uncertainties, and the installed base evolution strategy, particularly in view of the progressive extension facilities offered by this type of configuration structure.

10.5.2. *Dimensioning of the links of a gateway*

The example considered is that of an IP access gateway. We apply the properties of the model previously defined for IP services. Dimensioning is performed in two stages.

In the first stage, dimensioning is performed at call level, in order to determine (for a given rejection probability) the number of subscribers that can be connected to the *gateway.* As the aim is to guarantee a given rejection rate for each service, the Erlang formula is simply applied for each service or group of services. It is thus assumed that call acceptance is performed on each call category.

In the second stage, dimensioning is performed at flow level, to guarantee the quality of service at packet level. For real time services, we must first verify that all peak bit rates are of a lower order of magnitude than the bit rate of a link. This is the case if the links used have a bit rate of 155 Mbit/s or higher. We then apply the multi-bit rate formula, plus the *time sharing* formula for elastic traffic.

Let us apply this approach to a *gateway* that concentrates the traffic of n = 20,000 subscribers.

To simplify, we are allocating the same rejection probability, for example $P_R = 10^{-3}$, to all services. For each service we have for a type i, a penetration rate p_i and an activity per subscriber a_i:

$$P_{Ri} = E(N_i, A_i)\text{, where } A_i = p_i.a_i.n$$

and in view of the activity hypotheses defined in the performance objectives, we obtain, for example, for the VoIP G711 service: $A_1 = 0.5 \times 0.1 \times 20{,}000 = 1000$. And by applying the Erlang formula for a loss probability of 10^{-3}, we obtain $N_1 = 1070$.

If this is repeated for all the services, we arrive at Table 10.18.

A distinction must be drawn between the calculation *rule*, which applies to all traffic hypotheses, and its *application*, which is intended to quantify the resources for each specific traffic mix case. In this case we only take the performance target reference mix as an example. In reality each site will constitute a special case.

Table 10.18. *Multi-service mix*

Service $n = 10^4$ users	**VoIP G711**	**VoIP G729a**	**H261**	**H263**	**MPEG2**	**MPEG4**	**WEB**	**FTP**
Penetration/user (p)	0.5	0.5	0.01	0.01	0.1	0.2	1	0.01
Activity in erlang (a)	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Total activity (A)	1000	1000	20	20	200	400	2000	20
Number (N), of simultaneous calls at 10^{-3}	1070	1070	34	34	237	450	2094	34

Let us now determine the needs at flow level. As defined in the models chapter (Chapter 9), we draw a distinction between real time traffic and elastic traffic.

Real time traffic

These types of traffic consist in fact of the VoIP, H26x, and MPEG services. Amongst these services, another distinction should be drawn between constant bit rate services and variable bit rate services.Constant bit rate services: These services are simply sized on the basis of a link occupation rate of 0.9 (safety factor). If applied to our hypotheses, this gives:

Table 10.19. *Constant bit rates: calculation of bandwidth C_c*

Service N = 10,000 users	**VoIP G711**	**H261**	
Number of simultaneous calls N (at 10^{-3})	1070	34	
Bit rate by flow (kbit/s)			
- peak	80	384	
- mean	80	384	
- variance	0	0	
Total bit rate Mbit/s (c)	85.600	13.056	98.656
Link ($C_c = c/0.9$)			109.618

Variable bit rate services: For each service in the call phase, we have the mean bit rate requested m_i and the peak bit rate d_i, or the variance of the bit rate σ_i^2. When we have the peak bit rate d_i, we determine the variance by:

$$\sigma_i^2 = m_i d_i \tag{10-16}$$

For example, for the VoIP G729A service, the peak bit rate is 24 kbit/s, and the activity is of the ON-OFF type, at around 50%, and thus the average bit rate is 12 kbit/s and the variance is 288.

Table 10.20. *Stream traffic: calculation of the bandwidth C_s*

Service n = 10 000 users	**VoIP G729A**	**H263**	**MPEG2**	**MPEG4**	
Number of simultaneous calls N (10^{-3})	1070	34	237	450	
Bit rate/flow (kbit/s)					
-peak d_i	24	128			
-mean m_i	12	64	500	64	
Total m_i	12840	2176	118500	28800	162316
- variance σ_i^2	288	8192	250^2	32^2	
Total σ_i^2	308160	278528	14812500	460800	15859988
Z					97.71
C_s (10^{-6}) Mbit/s					179.592

We then apply the *peakedness factor* method (see Chapter 9) for a very low congestion rate P_s at the flow level, for example 10^{-6}.

The total traffic is: $M = \sum_i N_i m_i$. The total variance is: $\sigma^2 = \sum_i N_i \sigma_i^2$

We also have: $$z = \frac{\sum_i N_i \sigma_i^2}{\sum_i N_i m_i} \tag{10-17}$$

and the application of the Erlang formula gives the required capacity C_s:

$$P_s \approx E\left(\frac{C_s}{z}, \frac{M}{z}\right) \tag{10-18}$$

Based on the hypotheses of section 10.2, we thus obtain Table 10.20 (a capacity of about 179 Mbit/s, for an offered traffic of 162 Mbit/s) .

Elastic traffic

We apply the processor sharing model. As noted in Chapter 9, this model gives response times that correspond to bit rates $d = C_e(1-\rho\ ))$. By adopting a maximum load of 0.8, we obtain the results shown in Table 10.21.

Table 10.21. *Elastic traffic: calculation of the bandwidth C_e*

Service N = 10000 users	**WEB**	**FTP**	
Number of simultaneous calls N (10^{-3})	2094	34	
Bit rate by flow (kbit/s) - peak - mean	 – 15	 – 8	
Average number of 64 (or 56) kbit/s sources	561	4.86	
Total Mbit/s: c_e	31.410	0.272	31.682
$C_e = c_e/0.8$			39.602

Finally, we apply the rule for integration of stream and elastic traffic:

$$C = \text{Max}(C_s + C_c;\ C_e + C_{sm} + C_c) \tag{10-19}$$

$$C = \text{Max}(289.210;\ 311.536) = 311.536 \text{ Mbit/s}$$

In this example, there is no enormous bit rate gain using the integration rule compared with sizing without integration:

$$C = C_c + C_s + C_e = 109.618 + 162.316 + 39.602 = 328.812 \text{ Mbit/s}$$

This is quite simply because of the large proportion of real time traffic.

Finally, allowance must be made for failures. The capacity thus calculated must be ensured in the event of loss of a link (the occurrence of a double failure is not considered). The gateway will therefore be sized with four 155 Mbit/s links, i.e. three links in degraded mode, representing a capacity of 465 Mbit/s.

10.5.3. *Dimensioning of maintenance stock*

This activity is of course fundamental in guaranteeing the dependability of telecommunications systems, which consist of repairable systems, at least as far as the terrestrial part is concerned.

Calculations are generally simple, because we can very often assume a situation equivalent to a continuous re-stocking strategy, and almost always the number of equipments in the installed base of each type is sufficiently large compared with the stock to enable the simple application of the Poisson law (see Chapter 6). However, let us look at some practical aspects based on an example.

Let us consider a network comprising several sites for which we wish to dimension a common maintenance stock. Over the whole of the network concerned, we list the following types of equipment, together with their number and their respective failure rates (see Table 10.22).

Table 10.22. *Characteristics of the installed equipments*

Type i	N_i	λ_i 10^{-6}/h
1	6	3.5
2	6	3.5
3	6	1.5
4	6	4
5	12	3.3
6	78	0.16
7	78	6.3
8	624	0.9

As there are several types of equipment, the first problem is the definition of a stockout. In practice, we often define the overall stockout probability P_s, which is the probability of a stockout whatever the type of equipment affected.

We have:

$$P_S = \frac{\sum_i N_i \lambda_i P_{Si}}{\sum_i N_i \lambda_i} \qquad (10\text{-}20)$$

where P_{Si} is the probability of individual stockout for each type of equipment. To carry out this calculation, we begin by assuming that the P_{Si} values are equal to P_S. We then readjust the value of the P_{Si} to get as close as possible to the P_S that has been fixed.

Another practical problem also stems from the very low number of failures generated by certain types of equipment (either because there is a small number of such equipments or because they have a very low failure rate). The safe solution is to put at least one equipment of this type in stock. However, if the actual individual stockout probability is very considerably lower than that fixed, for example 10 times lower, it may be acceptable not to have this type of equipment in stock. But this then means that for this type of equipment $P_{Si} = 1$, which could have a very strong impact on P_S.

Let us apply the above to our example with a restocking time (rotation time) of 90 days (we are considering the case of sites abroad) and an overall stockout probability target of 2%. We thus obtain the following stock.

Table 10.23. *Dimensioning of spare parts*

Type $_i$	s_i	λ_i 10^{-6}/h
1	2	3.5
2	2	3.5
3	0	1.5
4	2	4
5	2	3.3
6	2	0.16
7	5	6.3
8	20	0.9

The reader will verify the fact that because the number of equipments of each type is necessarily a whole number, this gives an effective P_s of 1.38%, with P_{Si} of 2.4%.

The results of course depend strongly on the values adopted as the objective, on the restocking modes adopted, and fundamentally on the validity of the failure rates adopted. It should not be forgotten that a stockout could possibly have an impact on the availability of the system (repair times are lengthened), although usually in the event of a critical situation an emergency procedure will then be adopted for restocking.

In all cases, these strategies must be decided on and adjusted by the operator, based on failure rate forecasts and observations. These strategies may form the subject of contracts between the equipment supplier and the operators.

10.6. Operating monitoring

The aim here is to monitor the behaviour of systems in the operational life phase in the networks, in order to verify the validity of our forecasts, and our sizings, and also to gather more information from *in situ* observation.

We give below an example of equipment reliability monitoring.

Principle of operational reliability monitoring

The monitoring of the operational reliability of the equipment (boards, components, etc.) is usually the responsibility of the quality departments. The role of the performance engineer is partly to define, in close conjunction with these departments, procedures enabling the gathering of significant data enabling the evaluation of equipment reliability, and to provide assistance in the mathematical interpretation of results (trends and evolution over time, degree of confidence, etc.).

We give below an example of the procedure and the type of data to be gathered. The raw operational reliability data are provided by the quality departments based on returns of equipment recorded in repair centres. In the case of a telecommunication equipment manufacturer, this monitoring applies for example to the boards installed both on the test models and on clients' sites. The operating time of the boards is calculated on the basis of a reference data, such as the date of the issue of the product to the client plus 2 months (to allow for the model stage).

The raw return data are analysed by specialists (hardware designers, for example), object by object, based on fault report sheets opened by the repair centre for each object processed. This fine analysis enables the focussing of lines of improvement and weaknesses, if any, and the dividing up of the returns into "false failures" and "genuine failures": these purely symbolic terms are meant to help in drawing a distinction between manipulation and installation faults, and other faults.

Genuine failures and false failures can be classified as follows:

– genuine failures: component faults; manufacturing faults (brazing fault, pin twisted, copper breakage, etc.); doubts (faults not clearly identified); various faults (non repairable faults, standard exchanges);

– false failures: boards that are physically damaged (mainly caused by a lack of care during manipulations, transport, etc.); non-compliant cases (components missing or non-compliant); returns due to quality alerts; faults resulting from erroneous software configurations; cases in which there is no problem.

From the genuine failures and operating hours, we deduce:

– a "corrected" "real" operational failure rate, including all the genuine failures: we calculate a mean value and a confidence interval by applying the Chi-2 law:

$$\lambda_{op\_Real} = \frac{Nb\_genuine\_failures}{Nb\_operating\_hours} \tag{10-21}$$

– a "component" operational failure rate calculated solely on the basis of component failures:

$$\lambda_{op\_comp} = \frac{Nb\_component\_failures}{Nb\_operating\_hours} \tag{10-22}$$

These "real" operational failure rates are then compared to the predicted failure rates for possible action, depending on the result of the ratio. Thus, if the failure rate observed is very much greater than the predicted failure rate (ratio > 2:1), component analysis (visual checks, expert appraisals) and possibly corrective actions (replacements of certain components, quality alerts) are then undertaken.

Table 10.24 shows the typical structure of a reliability monitoring report, and presents examples of results observed on equipments (boards in this case), which have in most cases recently been commissioned on site.

It is important to recapitulate the approach:

– comparison with expected values;

– comparison between raw data and data after fault analysis: in this case, we are looking to draw a distinction between that which is purely a matter of catalectic reliability (and which can be estimated by the predicted failure rate) and that which is a matter of fine tuning equipment (poor manufacture, poor manipulations, poor installations, etc.), and which after identification will be corrected;

– importance of the cumulative number of hours;

– importance of having reports at different dates so as to monitor estimates over time.

Table 10.24. *Operational reliability monitoring report*

Eq^t	CUMULATIVE results at xx(date)									Raw RELIABILITY			Reliability after analysis		
N°	Total returns	Standard exchanges	No pb		Irreparable	Repaired		In progress	Hours of oper^{ion} (10^6)	λ pred (10^{-9})	λ oper (10^{-9})	Ratio $\lambda o/ \lambda p$	Genuine failures	λ real (genuine fail.)	Final ratio $\lambda r / \lambda p$
			W/out intervention	With intervention		Components	Others								
1	4	0	0	3	0	3	0	1	3.24	850	926	1.1	3	926	1.1
2	116	2	19	16	1	27	49	2	9.50	3000	7995	2.7	21	2209	0.7
3	33	1	1	16	0	4	11	0	4.56	970	3286	3.4	4	876	0.9
4	1	0	0	1	0	0	0	0	1.16	990			0	0	0.0
5	8	0	0	7	0	0	1	0	1.74	500	574	1.1	0	0	0.0
6	84	0	3	33	2	16	24	4	15.00	1590	2690	1.7	13	874	0.5
7	30	0	0	8	0	11	8	3	11.00	1600	1668	1.0	8	702	0.4
8	18	1	0	5	0	4	6	2	2.13	940	4687	5.0	4	1875	2.0
9	85	1	8	11	2	35	27	1	2.74	3950	22580	5.7	30	10926	2.8
10	80	0	3	9	1	48	18	1	1.55	4950	42419	8.6	29	18639	3.8
11	8	0	1	3	0	2	2	0	1.47	1150	2704	2.4	1	676	0.6
12	19	0	0	2	0	9	7	1	1.48	1830	10772	5.9	8	5386	2.9
13	7	0	0	2	0	2	3	0	16.00	290	306	1.1	2	122	0.4
14	8	0	1	1	0	2	4	0	2.00	1430	2996	2.1	2	999	0.7
15	8	2	0	4	0	0	2	0	0.93	880	2149	2.4	0	0	0.0
16	3	0	0	1	0	0	2	0	1.15	70	1738	24.8	0	0	0.0
17	10	0	1	0	1	0	8	0	1.03	110	7751	70.5	0	0	0.0
18	4	0	0	1	0	0	2	1	0.62	1090	3207	2.9	0	0	0.0
19	17	0	1	5	0	7	4	0	2.74	2460	4006	1.6	7	2549	1.0
20	3	0	0	0	0	0	2	1	0.48	2270	4103	1.8	0	0	0.0

The following elements should also be noted in the table.

There is a high proportion, in the order of 50%, of false failures: non-compliance with dossier, poor software configuration, physical data. Amongst the genuine failures found, another 50% are "component" failures and 30% are manufacturing faults. As we will verify later, this is because we are studying a set of boards at the start of their operational life.

Boards 15, 16, 17 and 18 present an abnormal number of faults compared with forecasts. They are not however "component" faults. The observed failure rate, although relatively insignificant because of the low number of hours of operation, has led to analysis and enabled the highlighting of manufacturing faults, and particularly brazing faults. This information made it possible to intervene at manufacturing plants.

Boards, like those mentioned, with a low cumulative number of hours of operation (<100,000 hours), present practically no component faults. This is for example the case of boards 4, 5, 15, 16, 17, 18 and 20. For these equipments, reliability can only be estimated by an upper limit with a certain confidence interval. For board 20, for example, the application of the Chi-2 law gives us a failure rate below 2000 Fit (2000 10^{-9}, 1 Fit = 10^{-9}) with 60% confidence. We are here in a case in which faulty elements are replaced as failures occur. If T is the cumulative observation time, which is equivalent to the cumulative duration of a so-called truncated test, and r is the number of failures observed, so the lower limit of the confidence interval for a unilateral confidence interval of $1-\alpha$, is (See Chapter 5):

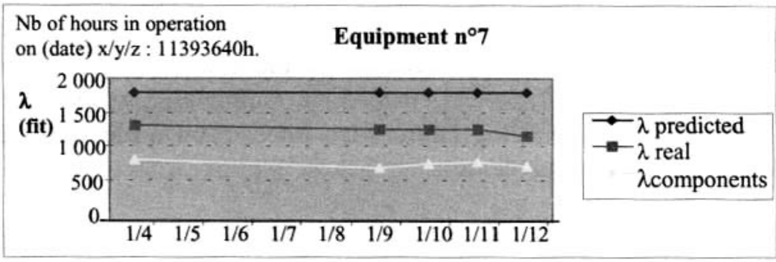
$$\theta_i = \frac{2T}{\chi^2_{2r+2;1-\alpha}}, \qquad (10\text{-}23)$$

which, it should be noted, means that MTBF is $\theta > \theta_i$ with a probability of $1-\alpha$.

In the case we are considering, we have: T = 487392, $\chi^2_{2;0.6} = 1.83$ (see tables in Appendix 2) and thus $\theta_i = 532\ 668$ that is λ_i = 1 877 10^{-9}.

Let us now consider equipments with a large number of hours of operation. The failure rates presented are very satisfactory when compared with the forecasts, and are even better than the expected values. This is in particular the case of boards 2 (ratio = 0.7), 6 (ratio = 0.5), 7 (ratio = 0.4), 11 (ratio = 0.6), 3 (ratio = 0.9), 13 (ratio = 0.4), 14 (ratio = 0.7).

In this respect, it is important to consider the trend in failure rates as a function of time. We show here, as examples, the cases of two boards: board 2 and board 7. Monitoring gives the following curves in Figure 10.12.



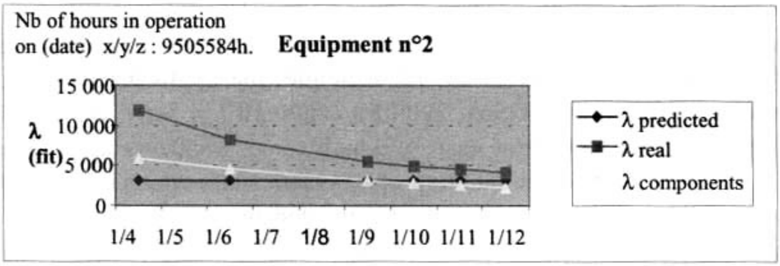


Figure 10.12. *Failure rate evolution as a function of time*

These graphs show that equipments, such as equipment no. 7, have reached their useful life period, and so their failure rate can be evaluated with a good degree of confidence. For example, the aggregate observation time of this equipment is $T = 1.1\ 10^7$ hours, and 8 genuine failures have been observed. We deduce from this by applying the Chi-2 law for a centred interval (see Chapter 5):

$$\frac{2T}{\chi^2_{2r+2;\alpha/2}} < \theta < \frac{2T}{\chi^2_{2r;1-\alpha/2}} \tag{10-24}$$

where $1 - \alpha = 0.8$, we obtain (See tables in Appendix 2):

$$\chi^2_{16;0.9} = 23.5 \text{ and } \chi^2_{18;0.1} = 10.9$$

Which gives: $9.36\times10^5 < \theta < 2.018\times10^6$, or $4.95\ 10^{-7} < \lambda < 1.06\ 10^{-6}$, with 80% confidence for a forecast λ of $1.59\ 10^{-6}$.

On the other hand, equipment no. 2 seems to be at the end of its "youth period". Its failure rate is tending towards the forecast value, and a precise estimate can only be made at the next observation period. As in simulation and in traffic, the importance of observing the evolution of the phenomenon studied over time should be stressed.

Finally, we would note that some equipments, although they have a large number of hours of operation, have a high failure rate compared with the forecast value. The boards concerned are 8, 9, 10 and 12. The analysis has enabled the identification of faults on specific components, and also soldering faults associated with these components, which obviously call for corrective action.

Many other lessons should be taken from these examples, but we will end our study at this point, and recommend that the reader should explore this subject more deeply in cases based on his/her own observations.

This page intentionally left blank

Conclusion

Our incursion into the domain of performance evaluation comes here to its end. Throughout these ten chapters, the reader has been able to discover, or rediscover, the foundations of disciplines which form the basis of operation of telecommunication networks. These include telecommunication principles, standards related to quality of service in telecommunication, probability theory, statistics, reliability, queueing theory, and the methodology that permits the efficient use of these techniques.

It is now for the engineer to practise, and to acquire his/her own expertise, by continuously exercising and learning from real problems.

But this is nothing but the starting point: the field of telecommunications is perpetually changing, raising new problems of performance. The expert is thus continuously dealing with new challenges. Practice, and mastering the whole set of knowledge presented in this book, will give him/her the tools needed to make the synthesis between past problems and issues to come, between the beauty of the mathematical construct and the field constraints.

If ever a profession was to be driven by modesty, constant learning and progress, it is this. And the reader is prompted to bear in mind this quotation: "The past cannot be revised, but present is just like materials at the builder's disposal, it is up to you to forge your future" (Antoine de Saint Exupéry, *Citadelle*).

We hope that this book, among others, will remain a worthy companion to the reader in his/her fascinating journeys ahead.

Appendix 1

Mathematics

We recall in this appendix several definitions and basic theorems concerning the functions of the complex variable. We give also the elementary results for power series, which are of constant usage in performance studies.

Functions of the complex variable: theorems and definitions

Definition of the function of the complex variable

Let Ω be an open set of the field C of complex numbers. A function of the complex variable on Ω is an application from Ω to C, which for every z in Ω yields a complex number denoted as:

$$Z = f(z).$$

It can always be written as:

$$Z = X(x, y) + iY(x, y).$$

The set Ω is the *domain* of the function.

Uniform function

The function $f(z)$ is said uniform (or single-valued) if every z in Ω has a single image $Z = f(z)$.

Continuity

Definitions about the limits of complex functions are identical to the ones concerning scalar functions.

Especially, z being a continuous function in the complex plane, every polynomial in z is a continuous function on the whole plane. Similarly, every rational fraction of z is continuous on the plane, except in these points where the denominator vanishes (these points are the *poles* of the fraction).

Analytical function

A function of the complex variable is derivable at a point of the complex plan if its derivative exists in that point. It is said analytic in a domain if it is derivable at every point of this domain.

A necessary and sufficient condition for the derivability of the function $f(z)$ at point $z = x + iy$ is that its real and imaginary parts $X(x, y)$ and $Y(x, y)$ are differentiable in $m(x, y)$, image of z, and verify the Cauchy-Riemann equations:

$$\frac{\partial X}{\partial x} = \frac{\partial Y}{\partial y}, \text{ and } \frac{\partial X}{\partial y} = -\frac{\partial Y}{\partial x}.$$

Holomorphic function

A function $f(z)$ is said holomorphic at a point z_0 if $f(z)$ is derivable in z_0 and if it exists a circle of centre z_0 in which $f(z)$ is continuous and uniform.

A function $f(z)$ is said holomorphic within a domain D if it has a derivative at each point of D, and if it is uniform in D.

Singular points and poles

Every point z_0 where the function $f(z)$ is not holomorphic is said to be a singular point (or singularity) of the function. The critical points of a multiple-valued function, and the poles of a fraction, are singularities. More generally, $z = z_0$ is a pole of $f(z)$ if $1/f(z)$ is holomorphic in z_0.

Integral of a continuous function

Let AB be a smooth arc in the complex plane:

$$\int_{AB} f(z)dz = \int_{AB} (X + iY)(dx + idy) = \int_{AB} Xdx - Ydy + i\int_{AB} Ydx + Xdy$$

The integral of a complex function is a linear combination of line integrals. It possesses all properties of the ordinary integral.

Example: a basic integral

$\int_C \frac{dz}{z-a}$, where C is the circle with centre $z = a$ and radius ρ.

Let us denote $|z-a| = \rho$ and $z - a = \rho e^{i\theta}$.

One has:

$$\int_C \frac{dz}{z-a} = \int_0^{2\pi} \frac{i\rho e^{i\theta}}{\rho e^{i\theta}} d\theta = \int_0^{2\pi} i d\theta = 2\pi i .$$

Cauchy's integral theorem

Given a function $f(z)$, holomorphic in the domain D, and C a smooth closed curve contained in D, then:

$$\int_C f(z) dz = 0$$

Cauchy integral

$$f(a) = \frac{1}{2\pi i} \int_{C^+} \frac{f(z)}{z-a} dz$$

C^+ meaning that the integral is evaluated along the positive direction (trigonometric, or counterclockwise sense) .

Paul Lévy's theorem

If $\phi(z)$ is an indefinitely derivable function, and if the series $\sum_{n=0}^{\infty} \frac{\phi^{(n)}(0)}{n!}(iu)^n$ is convergent, u being a real number, in a circle of radius a, then $\phi(z)$ is an analytic function in the band $-a < R(z) < +a$, $R(z)$ being the real part of z.

Corollaries

a) if $\phi(x)$ exists for x real, $\alpha < x < \beta$, $\phi(z)$ is analytic in the band $\alpha < R(z) < \beta$;
b) the closest to the origin singular point of $\phi(z)$, being on the convergence circle and outside the band of analyticity, is necessarily real.

Rouché's theorem

If $f(z)$ and $g(z)$ are analytical inside and on a closed contour C, and if $|g(z)| < |f(z)|$ on C, then $f(z)$ and $f(z)+g(z)$ have the same number of zeroes inside C.

Series expansions (real functions)

Taylor's formula

Let f be a continuous function in the interval $[a, b]$, having derivatives up to order n, continuous on the interval, and a derivative of order $n + 1$ on the interval $]a, b[$, then there exists c on $]a, b[$ such that:

$$f(b) - f(a) = \frac{(b-a)}{1!} f'(a) + \frac{(b-a)^2}{2!} f''(a) + .. \frac{(b-a)^n}{n!} f^{(n)}(a)$$
$$+ \frac{(b-a)^{n+1}}{(n+1)!} f^{(n+1)}(c)$$

A practical form for this result is:

$$f(x) = f(a) + \frac{(x-a)}{1!} f'(a) + \frac{(x-a)^2}{2!} f''(a) + ..$$

One gets the Mac-Laurin expansion if $a = 0$:

$$f(x) = f(0) + \frac{x}{1!} f'(0) + \frac{x^2}{2!} f''(0) + ..$$

Here are the expansions of several functions of constant use in the domain of performance studies.

Binomial theorem

Let us apply the Taylor formula to the function $(a+b)^n$, with n integer. One has:

$$(a+b)^n = a^n + \frac{b}{1!} n a^{n-1} + \ldots + \frac{b^p}{p!} n(n-1)\ldots(n-p+1) a^{n-p} + \ldots + b^{n-1} n a + b^n$$

The binomial coefficient is:

$$\binom{n}{p} = \frac{n(n-1)\ldots(n-p+1)}{p!} = \frac{n!}{p!(n-p)!} \text{ for } p > 0 \text{, and } \binom{n}{0} = 1.$$

From which the binomial theorem is written:

$$(a+b)^n = \binom{n}{0} a^n + \binom{n}{1} a^{n-1} b + \cdots + \binom{n}{p} a^{n-p} b^p + \cdots + \binom{n}{n} b^n.$$

Note that

$$\binom{n}{p} = \binom{n}{n-p}.$$

One verifies easily that:

$$\binom{n}{p} = \binom{n-1}{p-1} + \binom{n-1}{p}$$ (relation which allows building the Pascal triangle).

Using the binomial theorem, the expansion of the function $(1+x)^n$ is:

$$(1+x)^n = 1 + \binom{n}{1} x + \cdots + \binom{n}{p} x^p + \binom{n}{n} x^n.$$

Exponential function

$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \ldots \frac{x^n}{n!} + x^n \varepsilon(x)$, with ε such that $\varepsilon(x) \to 0$ as $x \to 0$, for every value of x.

Power function

$$a^x = 1 + \frac{x \ln a}{1!} + \frac{(x \ln a)^2}{2!} + \ldots, \text{ for every value of } x$$

Polynomial

$$(1+x)^m = 1 + mx + \frac{m(m-1)}{2!}x^2 + \ldots \frac{m(m-1)..(m-p+1)}{p!}x^p + \ldots., \text{ for arbitrary}$$

m positive or negative, integer or fractional.

If m is integer, one finds again the binomial development:

$$(1+x)^m = 1 + \binom{m}{1}x + \binom{m}{2}x^2 + \ldots \binom{m}{p}x^p + \ldots.$$

The following is an important special case:

$$\frac{1}{1+x} = 1 - x + x^2 - x^3 + \ldots. + (-1)^n x^n + x^n \varepsilon(x), \text{ if } |x| < 1$$

and also:

$$\frac{1}{1-x} = 1 + x + x^2 + x^3 + \ldots. + x^n + \ldots,$$

taking the derivative, one has:

$$\frac{1}{(1-x)^2} = x + 2x + 3x^2 + \ldots$$

Logarithm

$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \ldots, \text{ for } -1 < x \leq +1.$$

$$\ln(x) = 2\left[\frac{x-1}{x+1} + \frac{1}{3}\left(\frac{(x-1)}{(x+1)}\right)^3 + \frac{1}{5}\left(\frac{(x-1)}{(x+1)}\right)^5 + \ldots\right], \text{ for } x > 0.$$

Series expansion of a function of the complex variable

Taylor series

A function $f(z)$, holomorphic inside a circle C of centre a and on the circle itself, is expandable in the powers of $(z - a)$ and the series is convergent as long as z remains inside the circle. The expansion is unique, this is the Taylor expansion about $z = a$:

$$f(z) = f(a) + \frac{(z-a)}{1!} f'(a) + \frac{(z-a)^2}{2!} f''(a) + .. \frac{(z-a)^n}{n!} f^{(n)}(a) + ...$$

which is of the same form as in the real domain.

The expansions of the usual functions are identical to those obtained for the real variable, inside their *radius of convergence*. For instance,

$$e^z = 1 + \frac{z}{1!} + \frac{z^2}{2!} + ... \frac{z^n}{n!} + .., \text{ with } R, \text{ radius of convergence infinite.}$$

$$\frac{1}{1+z} = 1 - z + z^2 - z^3 + ... + (-1)^n z^n + .., \text{ with } R = 1$$

Laurent series expansion

A function $f(z)$ holomorphic inside an annulus of centre a and on its limits formed by the circles C and γ (γ inside C), is the sum of two convergent series, the first in the positive powers of de $(z - a)$, the other one in the powers of $\frac{1}{z-a}$.

One has:

$$f(z) = ... + \frac{A_{-n}}{(z-a)^n} + ... + \frac{A_{-1}}{(z-a)} + A_0 + A_1(z-a) + ... + A_n(z-a)^n + ...$$

which is the Laurent series of $f(z)$ about $z = a$, with:

$$A_n = \frac{1}{2\pi i} \int_{C^+} \frac{f(u)}{(u-a)^{n+1}} du \, ,$$

and $A_{-n} = \dfrac{1}{2\pi i} \displaystyle\int_{\gamma^+} (u-a)^{n-1} f(u)du$.

If a is the only singular point inside γ, then the Laurent series provides the expansion of $f(z)$ around the singularity $z = a$ (the result holds for z as close at possible of a but not for $z = a$). A_{-1} is the residue of $f(z)$ in a. This is, too, the coefficient of $(z-a)^{n-1}$ in the Taylor expansion of $(z-a)^n f(z)$.

Appendix 2

Tables

This appendix provides a brief set of data concerning the most commonly used distribution functions. More complete results are nowadays easy to obtain, through specific software packages or more comprehensive tables available e.g. on CD ROMs or on the Web. The goal of the following tables and graphs is to allow quickly visualising and quantifying the behaviour of important laws.

Erlang distribution

Erlang loss formula

$$E(N, A) = B = \frac{\frac{A^N}{N!}}{\sum_{j=0}^{N} \frac{A^j}{j!}}$$

Erlang tables

A (offered traffic) as a function of B (loss probability) and N (number of servers).

Table 1. *Erlang distribution*

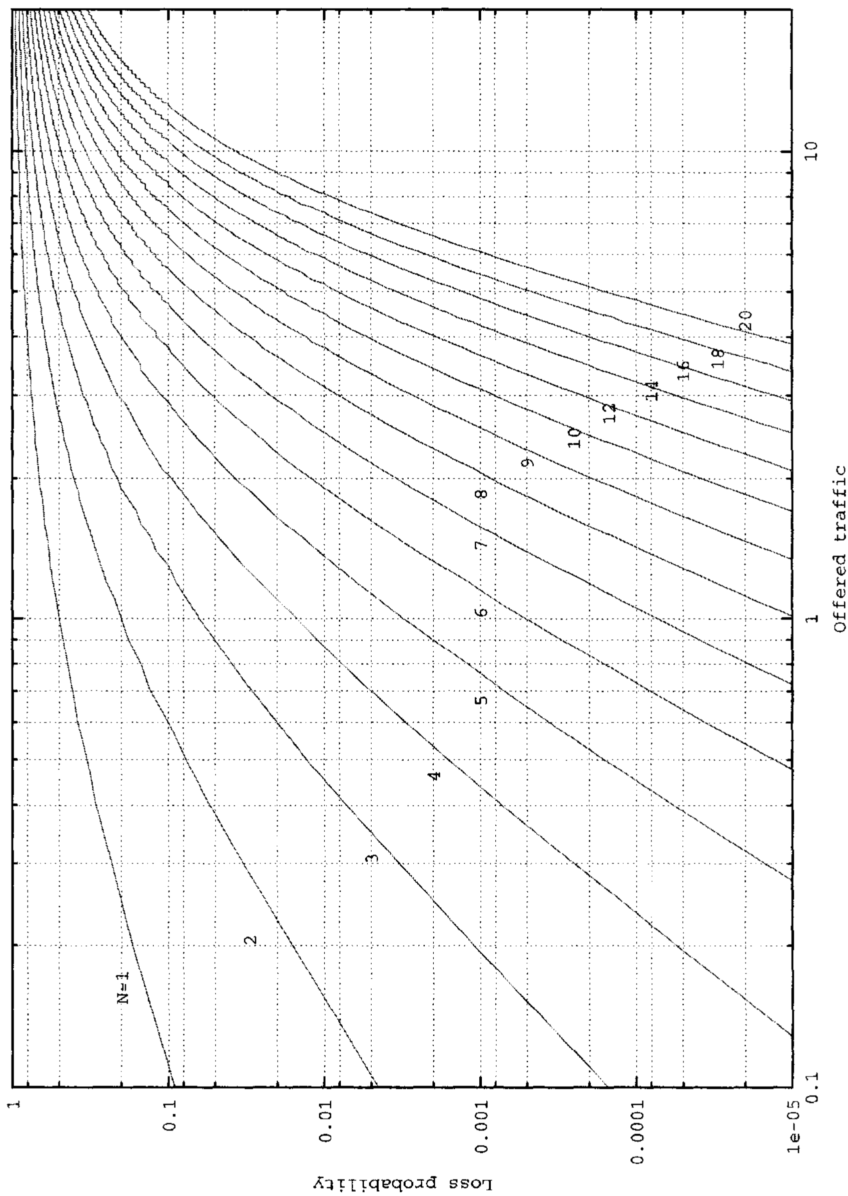
B% → N	0. 01	0.03	0.05	0.1	0.2	0.3	0.5
1	0.0001	0.0003	0.0005	0.001	0.002	0.003	0.005
3	0.0868	0.127	0.152	0.194	0.249	0.289	0.349
5	0.452	0.577	0.649	0.762	0.900	0.994	1.13
7	1.05	1.27	1.39	1.58	1.80	1.95	2.16
10	2.26	2.61	2.80	3.09	3.43	3.65	3.96
15	4.78	5.34	5.63	6.08	6.58	6.91	7.38
20	7.70	8.44	8.83	9.41	10.1	10.5	11.1
25	10.9	11.8	12.3	13	13.8	14.3	15
30	14.2	15.3	15.9	16.7	17.6	18.2	19

35	17.8	19	19.6	20.5	21.6	22.2	23.2
40	21.4	22.7	23.4	24.4	25.6	26.3	27.4
50	28.9	30.5	31.3	32.5	33.9	34.8	36
60	36.6	38.5	39.4	40.8	42.4	43.4	44.8
70	44.6	46.6	47.7	49.2	51	52.1	53.7
80	52.7	54.9	56.1	57.8	59.7	61	62.7
90	60.9	63.4	64.6	66.5	68.6	69.9	71.8
100	69.3	71.9	73.2	75.2	77.5	78.9	80.9
120	86.2	89.2	90.7	93	95.5	97.1	99.4
130	94.8	97.9	99.5	101.9	104.6	106.3	108.7
140	103.4	106.7	108.4	110.9	113.7	115.5	118
150	112.1	115.6	117.3	119.9	122.9	124.8	127.4
160	120.8	124.4	126.3	129	132.1	134	136.8
170	129.6	133.4	135.3	138.1	141.3	143.4	146.2
180	138.4	142.3	144.3	147.3	150.6	152.7	155.7
190	147.3	151.3	153.4	156.4	159.8	162.1	165.2
200	156.2	160.3	162.5	165.6	169.2	171.4	174.6
210	165.1	169.4	171.6	174.8	178.5	180.9	184.2
220	174	178.5	180.7	184.1	187.8	190.3	193.7
230	183	187.6	189.9	193.3	197.2	199.7	203.2
240	192	196.7	199.1	202.6	206.6	209.2	212.8
250	201	205.8	208.3	211.9	216	218.7	222.4
300	246.4	251.8	254.6	258.6	263.2	266.2	270.4
350	292.3	298.2	301.2	305.7	310.8	314.1	318.7
400	338.4	344.8	348.1	353	358.5	362.1	367.2
500	431.4	438.8	442.5	448.2	454.5	458.7	464.5
600	525.2	533.4	537.6	543.9	551	555.7	562.3
700	619.5	628.5	633.2	640.1	647.9	653.1	660.4
800	714.3	724	729.1	736.6	745.1	750.7	758.7
900	809.4	819.9	825.3	833.3	842.5	848.6	857.2
1000	904.8	916	921.7	930.3	940.1	946.6	955.9

B% → N	1	2	5	10	15	20
1	0.0101	0.0204	0.0526	0.111	0.176	0.250
3	0.455	0.602	0.899	1.27	1.60	1.93
5	1.36	1.66	2.22	2.88	3.45	4.01

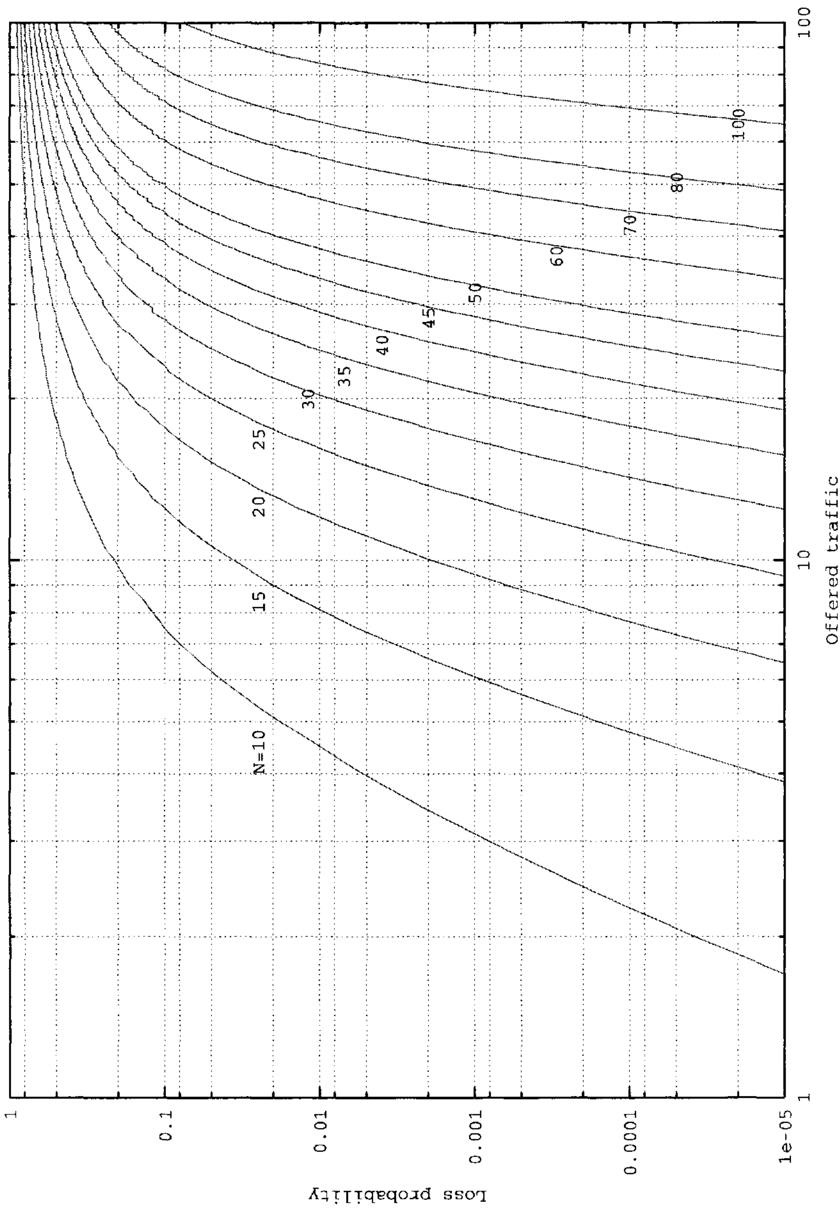
7	2.50	2.94	3.74	4.67	5.46	6.23
10	4.46	5.08	6.22	7.51	8.62	9.68
15	8.11	9.01	10.6	12.5	14.1	15.6
20	12	13.2	15.2	17.6	19.6	21.6
25	16.1	17.5	20	22.8	25.3	27.7
30	20.3	21.9	24.8	28.1	31	33.8
35	24.6	26.4	29.7	33.4	36.7	40
40	29	31	34.6	38.8	42.5	46.1
50	37.9	40.3	44.5	49.6	54	58.5
60	46.9	49.6	54.6	60.4	65.6	70.9
70	56.1	59.1	64.7	71.3	77.3	83.3
80	65.4	68.7	74.8	82.2	88.9	95.7
90	74.7	78.3	85	93.1	100.6	108.2
100	84.1	88	95.2	104.1	112.3	120.6
120	103	107.4	115.8	126.1	135.7	145.6
130	112.5	117.2	126.1	137.1	147.4	158
140	122	127	136.4	148.1	159.1	170.5
150	131.6	136.8	146.7	159.1	170.8	183
160	141.2	146.6	157	170.2	182.5	195.5
170	150.8	156.5	167.4	181.2	194.2	207.9
180	160.4	166.4	177.8	192.2	206	220.4
190	170.1	176.3	188.1	203.3	217.7	232.9
200	179.7	186.2	198.5	214.3	229.4	245.4
210	189.4	196.1	208.9	225.4	241.2	257.9
220	199.1	206	219.3	236.4	252.9	270.4
230	208.8	215.9	229.7	247.5	264.7	282.8
240	218.6	225.9	240.1	258.6	276.4	295.3
250	228.3	235.8	250.5	269.6	288.1	307.8
300	277.1	285.7	302.6	325	346.9	370.3
350	326.2	335.7	354.8	380.4	405.6	432.7
400	375.3	385.9	407.1	435.8	464.4	495.2
500	474	486.4	511.8	546.7	582	620.2
600	573.1	587.2	616.5	657.7	699.6	745.1
700	672.4	688.2	721.4	768.7	817.2	870.1
800	771.8	789.3	826.4	879.7	934.8	995.1
900	871.5	890.6	931.4	990.8	1052	1120
1000	971.2	991.9	1036	1102	1170	1245

Erlang curves $B = f(A, N)$



Erlang curves $A = 0.1–10$

Erlang curves B = f(A, N)

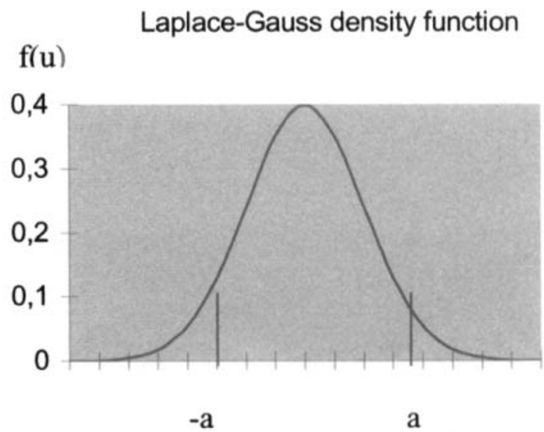


Erlang curves A = 10–100

Normal distribution (Gauss distribution)

Laplace-Gauss density function (standard Normal density)

$$f(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}}$$



Laplace-Gauss distribution function

$$1 - P(-a \leq u \leq a) = 2\frac{1}{\sqrt{2\pi}} \int_a^{\infty} e^{-\frac{u^2}{2}} du$$

Values of a as a function of P (e.g. $P = 10^{-3}$: $a = 3.29$ and $P = 0.11$: $a = 1.598$)

Table 2. *Laplace-Gauss distribution (standard normal distribution)*

P	10-3	10-4	10-5	10-6	10-7	10-8	10-9
	3.290	3.890	4.417	4.892	5.327	5.731	6.110

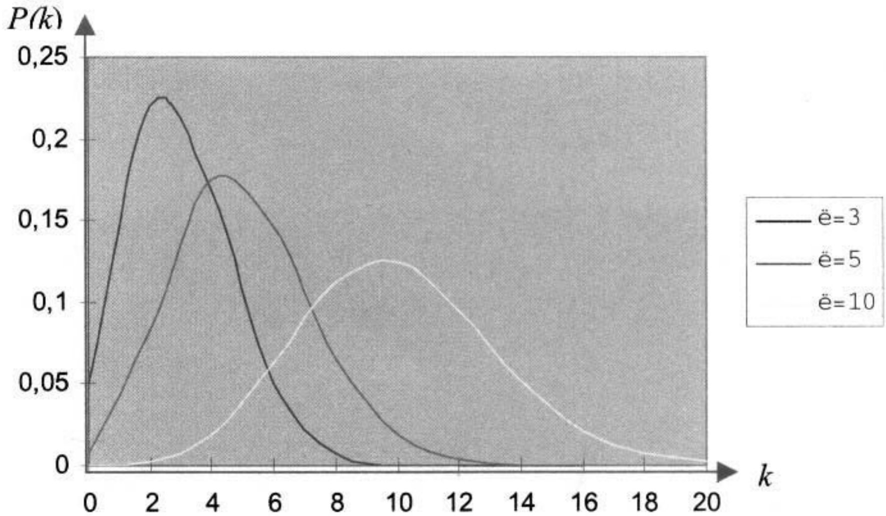
P	0	0.01	0.02	0.03	0.04
0		2.576	2.326	2.170	2.054
0.1	1.645	1.598	1.555	1.514	1.476
0.2	1.281	1.253	1.226	1.200	1.175
0.3	1.036	1.015	0.994	0.974	0.954
0.4	0.842	0.824	0.806	0.789	0.772
0.5	0.674	0.659	0. 643	0.628	0.613
0.6	0.524	0.510	0.482	0.482	0.468
0.7	0.385	0.372	0.345	0.345	0.332
0.8	0.253	0.240	0.215	0.215	0.202
0.9	0.126	0.113	0.088	0.088	0.075

P	0.05	0.06	0.07	0.08	0.09
0	1.960	1.880	1.812	1.751	1.695
0.1	1.440	1.405	1.372	1.341	1.310
0.2	1.150	1.126	1.103	1.080	1.058
0.3	0.934	0.915	0.896	0.878	0.859
0.4	0.755	0.739	0.722	0.706	0.609
0.5	0.598	0.583	0.568	0.553	0.539
0.6	0.454	0.440	0.426	0.412	0.399
0.7	0.318	0.305	0.292	0.279	0.266
0.8	0.189	0.176	0.164	0.151	0.138
0.9	0.063	0.050	0.038	0.025	0.012

Poisson distribution

Poisson density function

$$P(k) = e^{-\lambda} \frac{\lambda^k}{k!}$$



Poisson table: density function

$$P(x = k) = e^{-\lambda} \frac{\lambda^k}{k!}$$

Table 3. *Poisson density function*

λ → k	0.1	0.3	0.5	1	2	3	4
0	9.0e-1	7.4e-1	6.1e-1	3.7e-1	1.3e-1	5.0e-2	1.8e-2
1	9.0e-2	2.2e-1	3.0e-1	3.7e-1	2.7e-1	1.5e-1	7.3e-2
2	4.5e-3	3.3e-2	7.6e-2	1.8e-1	2.7e-1	2.2e-1	1.5e-1
3	1.5e-4	3.3e-3	1.3e-2	6.1e-2	1.8e-1	2.2e-1	1.9e-1
4	4.0e-6	2.5e-4	1.6e-3	1.5e-2	9.0e-2	1.7e-1	1.9e-1
5		1.5e-4	1.6e-4	3.0e-3	3.6e-2	1.0e-1	1.6e-1

6		1e-6	1.3e-5	5.1e-4	1.2e-2	5.0e-2	1.0e-1
7			1e-6	7.3e-5	3.4e-3	2.2e-2	5.9e-2
8				9e-6	8.6e-4	8.1e-3	3.0e-2
9				1e-6	1.9e-4	2.7e-3	1.3e-2
10					3.8e-5	8.1e-4	5.3e-3
12					1e-6	5.5e-5	6.4e-4
15						1e-6	1.5e-5
18							
20							
22							
25							
28							
30							

$\lambda \rightarrow$ k	5	6	7	8	9	10
0	6.7e-3	2.5e-3	9.1e-4	3.3e-4	1.2e-4	1.7e-5
1	3.4e-2	1.5e-2	6.4e-3	2.7e-3	1.1e-3	1.8e-4
2	8.4e-2	4.5e-2	2.2e-2	1.1e-2	5.0e-3	1.0e-3
3	1.4e-1	8.9e-2	5.2e-2	2.9e-2	1.5e-2	3.7e-3
4	1.7e-1	1.3e-1	9.1e-2	5.7e-2	3.4e-2	1.0e-2
5	1.7e-1	1.6e-1	1.3e-1	9.2e-2	6.1e-2	2.2e-2
6	1.5e-1	1.6e-1	1.5e-1	1.2e-1	9.1e-2	4.1e-2
7	1.0e-1	1.4e-1	1.5e-1	1.4e-1	1.2e-1	6.4e-2
8	6.5e-2	1.0e-1	1.3e-1	1.4e-1	1.3e-1	8.9e-2
9	3.6e-2	6.9e-2	1.0e-1	1.2e-1	1.3e-1	1.1e-1
10	1.8e-2	4.1e-2	7.1e-2	9.9e-2	1.2e-1	1.2e-1
12	3.4e-3	1.1e-2	2.6e-2	4.8e-2	7.3e-2	1.1e-1
15	1.6e-4	8.9e-4	3.3e-3	9.0e-3	1.9e-2	5.3e-2

18	4e-6	3.9e-5	2.3e-4	9.4e-4	2.9e-3	1.4e-2
20		4e-6	3.0e-5	1.6e-4	6.2e-4	4.6e-3
22			3e-6	2.2e-5	1.1e-4	1.2e-3
25				1e-6	6e-6	1.2e-4
28					1e-6	8e-6
30						1e-6

Poisson table: distribution function

$$P(\leq k) = \sum_{0}^{k} e^{-\lambda} \frac{\lambda^x}{x!}, \ x \text{ integer}$$

Table 4. *Poisson distribution function*

k → λ	0	1	2	3	4	5	6
0.1	0.90	0.99	1				
0.2	0.82	0.98	1				
0.3	0.74	0.96	1	1			
0.4	0.67	0.94	0.99	1			
0.5	0.61	0.91	0.99	1			
0.6	0.55	0.88	0.98	1	1		
0.7	0.50	0.84	0.97	0.99	1		
0.8	0.45	0.81	0.95	0.99	1		
0.9	0.41	0.77	0.94	0.99	1	1	
1	0.37	0.74	0.92	0.98	1	1	1
2	0.13	0.41	0.68	0.86	0.95	0.98	1
3	0.50	0.20	0.42	0.65	0.81	0.92	0.97
4	0.02	0.09	0.24	0.43	0.63	0.78	0.89
5	0.01	0.04	0.12	0.26	0.44	0.62	0.76
6	0.00	0.02	0.06	0.15	0.28	0.45	0.61
7		0.01	0.03	0.08	0.17	0.30	0.45
8		0.00	0.01	0.04	0.10	0.19	0.31
9			0.00	0.02	0.05	0.12	0.21
10				0.01	0.03	0.07	0.13
11				0.00	0.01	0.04	0.08
12					0.01	0.02	0.05
13					0.00	0.01	0.03
14						0.01	0.01
15						0.00	0.00

k → λ	7	8	9	10	15	20	25
0.1							
0.2							
0.3							
0.4							
0.5							
0.6							
0.7							
0.8							
0.9							
1	1						
2	1	1	1				
3	0.99	1	1	1			
4	0.95	0.98	0.99	1			
5	0.87	0.93	0.97	0.99	1		
6	0.74	0.85	0.92	0.98	1		
7	0.60	0.73	0.83	0.90	1		
8	0.45	0.60	0.72	0.82	0.99	1	
9	0.32	0.46	0.59	0.71	0.98	1	
10	0.22	0.33	0.46	0.58	0.95	1	
11	0.14	0.23	0.34	0.46	0.91	0.99	
12	0.09	0.15	0.24	0.35	0.88	0.99	
13	0.05	0.10	0.17	0.25	0.76	0.97	1
14	0.03	0.06	0.11	0.18	0.67	0.95	1
15	0.02	0.04	0.07	0.12	0.57	0.92	0.99

Bernoulli distribution

Bernoulli density function

$$\Pr(x = k) = \frac{N!}{k!(N-k)!} p^k (1-p)^{N-k}$$

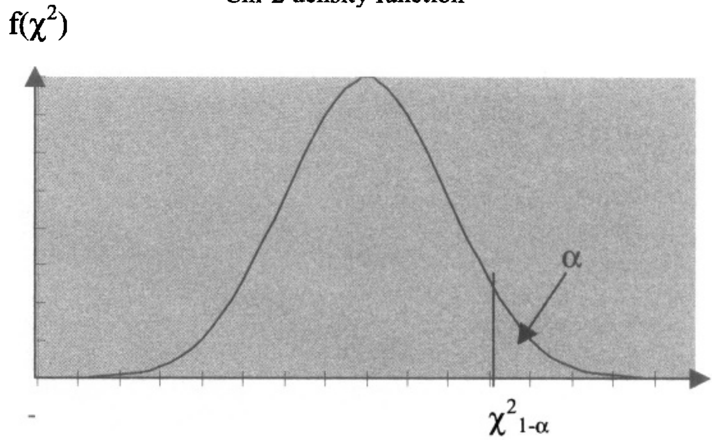
Table 5. *Bernoulli density function*

N	p%→ k	5	10	20
2	0	0.9025	0.81	0.64
	1	0.0950	0.18	0.32
	2	0.0025	0.01	0.04
3	0	0.8574	0.729	0.512
	1	0.1354	0.243	0.384
	2	0.0071	0.027	0.096
	3	0.0001	0.001	0.008
5	0	0.7738	0.5905	0.3277
	1	0.2036	0.328	0.4096
	2	0.0214	0.0729	0.2048
	3	0.0011	0.0081	0.0512
	4		0.0005	0.0064
	5			0.0003
8	0	0.6634	0.4305	0.1678
	1	0.2793	0.3826	0.3355
	2	0.0515	0.1488	0.2936
	3	0.0054	0.0331	0.1468
	4	0.0004	0.0046	0.0459
	5		0.0004	0.0092
	6			0.0011
	7			0.0001
10	0	0.5987	0.3487	0.1074
	1	0.3151	0.3874	0.2684
	2	0.0746	0.1937	0.3020
	3	0.0105	0.0574	0.2013
	4	0.0010	0.0112	0.0881
	5	0.0001	0.0015	0.0264
	6		0.0001	0.0055
	7			0.0008
	8			0.0001

N	p%→ k	30	40	50
2	0	0.49	0.36	0.25
	1	0.42	0.48	0.50
	2	0.09	0.16	0.25
3	0	0.343	0.216	0.125
	1	0.441	0n432	0.375
	2	0.189	0.288	0.375
	3	0.027	0.064	0.125
5	0	0.1681	0.0778	0.0312
	1	0.3602	0.2592	0.1562
	2	0.3087	0.3456	0.3125
	3	0.1323	0.2304	0.3125
	4	0.0284	0.0768	0.1562
	5	0.0024	0.0102	0.0312
8	0	0.0576	0.0168	0.0039
	1	0.1977	0.0896	0.0312
	2	0.2965	0.2090	0.1094
	3	0.2541	0.2787	0.2188
	4	0.1361	0.2322	0.2734
	5	0.0467	0.1239	0.2188
	6	0.0100	0.0413	0.1094
	7	0.0012	0.0079	0.0312
	8	0.0001	0.0007	0.0039
10	0	0.0282	0.0060	0.0010
	1	0.1211	0.0403	0.0098
	2	0.2335	0.1209	0.0439
	3	0.2668	0.2150	0.1172
	4	0.2001	0.2508	0.2051
	5	0.1029	0.2007	0.2461
	6	0.0368	0.1115	0.2051
	7	0.0090	0.0425	0.1172
	8	0.0014	0.0106	0.0439
	9	0.0001	0.0016	0.0098
	10		0.0001	0.0010

Chi-2 distribution

Chi-2 density function



Chi-2 table

$\chi^2_{R;\beta}$. Chi-2 distribution with R degrees of freedom, at confidence level $\beta = 1 - \alpha$

$\Pr\{\chi 2 > \chi 2(R)\} = \alpha$, example: for $R = 30$. $\Pr\{\chi 2 > 40{,}26)\} = 0{,}10$

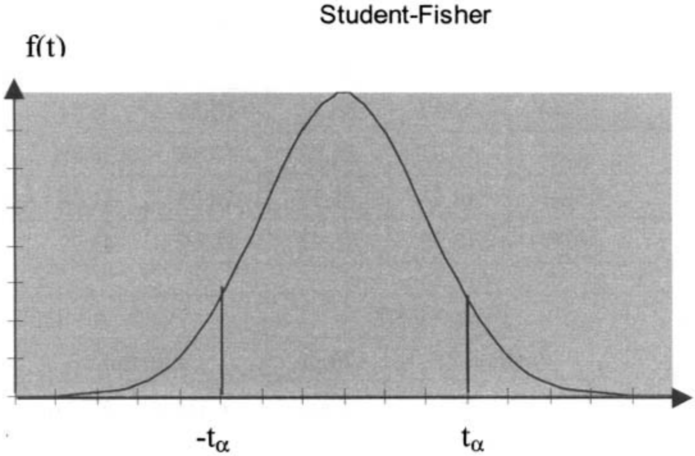
Table 6. *χ^2 distribution (confidence interval)*

$\chi^2_{R;\beta}$	$1-\alpha = 0.95$			$1-\alpha = 0.90$		
	Interval		limit	Interval		limit
R	$\chi^2_{0.025}$	$\chi^2_{0.975}$	$\chi^2_{0.95}$	$\chi^2_{0.05}$	$\chi^2_{0.95}$	$\chi^2_{0.90}$
2	0.05	7.38	5.99	0.103	5.99	4.61
4	0.48	11.1	9.49	0.711	9.49	7.78
6	1.24	14.5	12.6	1.64	12.6	10.6
8	2.18	17.5	15.5	8.73	15.5	13.4
10	3.25	20.5	18.3	3.94	18.3	16.0
12	4.40	23.3	21.0	5.23	21.0	18.5
14	5.63	26.1	23.7	6.57	23.7	21.1

16	6.91	28.8	26.3	7.96	26.3	23.5
18	8.23	31.5	28.9	9.39	28.9	26.0
20	9.59	34.2	31.4	10.9	31.4	28.4
22	10.98	36.78	33.92	12.34	33.92	30.81
24	12.40	31.37	36.41	13.85	36.41	33.20
26	13.84	41.92	38.88	15.38	38.88	35.56
28	15.31	44.46	41.34	16.93	41.34	37.92
30	16.79	46.98	43.77	18.49	43.77	40.26

$\chi^2_{R;\beta}$	$1-\alpha=0.80$			$1-\alpha=0.60$		
	Interval		limit	Interval		limit
R	$\chi^2_{0.10}$	$\chi^2_{0.90}$	$\chi^2_{0.80}$	$\chi^2_{0.20}$	$\chi^2_{0.80}$	$\chi^2_{0.60}$
2	0.21	4.61	3.22	0.45	3.22	1.83
4	1.06	7.78	5.99	1.65	5.99	4.04
6	2.20	10.6	8.56	3.07	8.56	6.21
8	3.49	13.4	11.1	4.59	11.1	8.35
10	4.87	16.0	13.4	6.18	13.4	10.5
12	6.30	18.5	15.8	7.81	15.8	12.6
14	7.79	21.1	18.1	9.47	18.1	14.7
16	9.31	23.5	20.5	11.15	20.5	16.8
18	10.9	26.0	22.8	12.85	22.8	18.9
20	12.4	28.4	25.0	14.57	25.0	21.0
22	14.04	30.81	27.3	16.31	27.3	23.1
24	15.66	33.20	29.5	18.06	29.5	25.2
26	17.29	35.56	31.8	19.82	31.8	27.3
28	18.94	37.92	34.0	21.59	34.0	29.4
30	20.60	40.26	36.2	23.36	36.2	31.5

Student-Fisher distribution



Student-Fisher table

$1 - P(-t_\alpha \le t \le t_\alpha) = \alpha$ as a function of *r* (degree of freedom)

Table 7. *Student-Fisher distribution (confidence interval)*

α → r	0.9	0.8	0.7	0.6	0.5	0.4	0.3
1	0.158	0.325	0.510	0.727	1	1.376	1.963
2	0.142	0.289	0.445	0.617	0.816	1.061	1.386
3	0.137	0.277	0.424	0.584	0.765	0.978	1.250
4	0.134	0.271	0.414	0.569	0.741	0.941	1.190
5	0.132	0.267	0.408	0.559	0.727	0.920	1.156
6	0.131	0.265	0.404	0.553	0.718	0.906	1.134
7	0.130	0.263	0.402	0.549	0.711	0.896	1.119
8	0.130	0.262	0.399	0.546	0.706	0.889	1.108
9	0.129	0.261	0.398	0.543	0.703	0.883	1.100
10	0.129	0.260	0.397	0.542	0.700	0.879	1.093
12	0.128	0.259	0.395	0.539	0.695	0.873	1.083
15	0.128	0.258	0.393	0.536	0.691	0.866	1.074
18	0.127	0.257	0.392	0.534	0.688	0.862	1.067
20	0.127	0.257	0.391	0.533	0.687	0.860	1.064
22	0.127	0.256	0.390	0.532	0.686	0.858	1.061
25	0.127	0.256	0.390	0.531	0.684	0.856	1.058

28	0.127	0.256	0.389	0.530	0.683	0.855	1.056
30	0.127	0.256	0.389	0.530	0.683	0.854	1.055

$\alpha \rightarrow$ r	0.2	0.1	0.05	0.02	0.01	0.001
1	3.078	6.314	12.706	31.821	63.657	636.62
2	1.886	2.920	4.303	6.965	9.925	31.598
3	1.638	2.353	3.182	4.541	5.481	12.924
4	1.533	2.132	2.776	3.747	4.604	8.610
5	1.476	2.015	2.571	3.365	4.032	6.869
6	1.440	1.943	2.447	3.143	3.707	5.959
7	1.415	1.895	2.365	2.998	3.499	5.408
8	1.397	1.860	2.306	2.896	3.355	5.041
9	1.383	1.833	2.262	2.821	3.250	4.781
10	1.372	1.812	2.228	2.764	3.169	4.587
12	1.356	1.782	2.179	2.681	3.055	4.318
15	1.341	1.753	2.131	2.602	2.947	4.073
18	1.330	1.734	2.101	2.552	2.878	3.922
20	1.325	1.725	2.086	2.528	2.845	3.850
22	1.321	1.717	2.074	2.508	2.819	3.792
25	1.316	1.708	2.060	2.485	2.787	3.725
28	1.313	1.701	2.048	2.467	2.763	3.674
30	1.310	1.697	2.042	2.457	2.750	3.646

This page intentionally left blank

Bibliography

IETF: Internet Engineering Task Force http://www.ietf.org/

ETSI: European Telecommunications Standards Institute http://www.etsi.org/

UIT: Union Internationale des Télécommunications http://www.itu.int/

ITU-T Recommendations series:

A. Organization of the work of ITU-T
B. Means of expression: definitions, symbols, classification
C. General Telecommunications statistics
D. General Tariff principles
E. Overall network operation, telephone service, service operation and human factors
F. Non telephone telecommunication services
G. Transmission system and media, digital systems and networks
H. Audiovisual and multimedia systems
I. Integrated services digital network
J. Transmission of television, sound programme and other multimedia signals
K. Protection against interference
L. Construction, installation ans protection of cables and other elements of outside plant
M. TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
N. Maintenance: international sound programme and television transmission circuits
O. Specifications of measuring equipment
P. Telephone transmission quality, telephone installations, local line networks
Q. Switching and signalling
R. Telegraph transmission
S. Telegraph services terminal equipment
T. Terminals for telematic services
U. Telegraph switching
V. Data communication over the telephone network
X. Data networks and open system communication
Y. Global information infrastructure and internet protocol aspects
Z. Languages and general software aspects for telecommunication systems

[BAN 00] Banks J., Carson J., Nelson B., *Discrete-Event System Simulation*, Prentice Hall, 2000.

[BAS 90] Bastos Martini M.R., Kanoun K., Moreira de Souza J., *Software reliability evaluation of the Tropico-R switching system*, IEEE Transaction Reliability, Vol 39, N°3, p.369–379, August 1990.

[BAY 00] Baynat B., *Théorie des files d'attente*, Hermès, Collection réseaux et télécommunications, Paris, 2000.

[BAZ 66] Bazovsky, *Fiabilité: théorie et Pratique de la Sûreté de Fonctionnement*, Dunod, 1966.

[BER 73] Berman S., Bézard R., *Statistique pour papa et quelques autres aussi selon toutes probabilités*, 2 tomes, Chiron, 1973.

[BOX 79] Boxma O.J., *On a tandem queueing model with identical service times at both counters*, Adv. Appl. Prob., Vol. 11, 1979, pp 616–659.

[BRA 83] Bradley P., Fox B.L., Schrage L.E., *A Guide to Simulation,* Springer-Verlag, New York, 1983.

[BUR 56] Burke P.J., *The output of a queueing system*, J. Op. Res. Soc. Amer.4, p. 699–704, 1956.

[BUX 81] W. Bux, *Local area subnetworks: a performance comparison*, IEEE Transaction on Communication, vol Com 29, n° 10, Oct. 1981.

[CIN 75] Cinlar E., *Introduction to stochastic processes*, Prentice Hall, 1975.

[COF 70] Coffman JR E.G., Muntz R., Trotter H., *Waiting time for processor sharing systems*, JACM, vol. 17, n° 1, January 1970.

[COH 82] Cohen J.W., *The single server queue*, North Holland, 1982.

[COM 99] Comer D.E., *Computer Networks and Internets*, Prentice Hall, 1999.

[CRA 77] Crane, Lemoine, *An Introduction to the Regenerative Method for Simulation Output Analysis,* Springer Verlag, LNCI Sciences, 1977.

[CRO 97] Crovella M.E., Bestavros A., *Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes*, IEEE/ACM Transactions on Networking, vol. 5, n° 6, p. 835–846, December 1997.

[FEL 00] Feldman P., *Discrete-event Simulation for Performance Evaluation With Algorithms and Examples in C and C++,* John Wiley & Sons, 2000.

[FEL 71] Feller, W. *An Introduction to Probability Theory and its Applications*, John Wiley and Sons, 1971.

[FIC 01] Fiche G., Veillard Y., *The tunneling technique and the tandem queue effect*, IFIP workshop on IP and ATM traffic management, Paris, 2001

[FIC 85] Fiche G., Le Gall P., Recupero S., *Study of blocking for multislot connection in digital link systems*, ITC 11, 1985.

[FIC 90] Fiche G., Le Corre F., *A method for evaluating CCITT performance parameters, application to a switching system*, ISS 13, 1990.

[FIC 93] Fiche G., Sabourin T., Le Gac M.L., Thomazeau F., *Assessing the performance of the Alcatel 1000 system*, Commutation & Transmission, n° 4, 1993.

[FIC 96] Fiche G., Le Palud C., Rouillard S., *Study of the performance of an ATM Clos switching network based on the composite technique*, IFIP, ATM Networks performance modelling and analysis, Chapman &Hall, 1996.

[FIS 78] Fishman G., *Principles of Discrete-Event Simulation,* John Wiley, 1978.

[FIS 96] Fishman G., *Monte Carlo, Concepts, Algorithms and Applications*, Springer Verlag, 1996.

[GEL 78] Gelenbe E. Mitrani I., *Analysis and synthesis of computer systems*, Academic Press, 1978

[GIM 65] Gimpelson L.A., *Analysis of mixtures of wide and narrowband traffic*, IEEE trans on Comm., Com 13, n° 3, 1965.

[GNE 72] Gnedenko B., Beliaev Y., Soloviev A., *Méthodes Mathématiques en Théorie de la Fiabilité*, Editions MIR, 1972.

[GRO 85] Gross D., Harris C., *Fundamentals of Queueing Theory*, 2nd edition, John Wiley and Sons, 1985.

[IETF] IETF. RFCs.701,1349,2210 to2216,2381,2474,2475,2597,2598.

[KLE 72] Kleinrock L., *Communication Nets, Stochastic Message Flow and Delay*, Dover, 1972.

[KLE 75] Kleinrock L., *Queueing Systems vol.1: Theory*, John Wiley & Sons, 1975.

[KLE 76] Kleinrock L., *Queueing Systems vol. 2: Computer Applications*, John Wiley and Sons, 1976.

[KNU 69] Knuth D., *The Art of Computer Programming* (vol. 2), Addison-Wesley, Mass., 1969.

[KRI 96] Krickeberg K., *Petit Cours de Statistique*, Springer, 1996.

[KUE 79] Kuehn P.J., *Multi queue systems with non exhaustive cyclic service* , Bell System Technical Journal, vol.58, n°3, pp 671-698, mar.1979.

[LAB 92] Labourdette J.F., Hart G.W., *Blocking probabilities in multitraffic Loss Systems: insensivity, asymptotic behavior, and approximations*, IEEE transactions on communications, vol. 40, n° 8, 1992.

[LAM 80] Lam S.S., *A CSMA protocol for Local Networks*, Computer networks, vol. 4 n° 1, 1980.

[LAW 91] Law A.M., David Kelton W., *Simulation Modelling and Analysis.* McGraw-Hill, Industrial Engineering series, New York, 1991.

[LEC 97] P. L'Ecuyer, T.H. Andres, *A Random Number Generator Based on the Combination of Four LCGs*, Mathematics and Computers in Simulation, 44, 99, 1997.

[LEE 55] Lee C.Y., *Analysis of switching networks*, Bell System Technical Journal, n° 34, 1955.

[LEG 62] Le Gall P., *Les systèmes avec ou sans attente et les processus stochastiques*, Dunod, Paris, 1962.

[LEG 93] Le Gall P., *The application of Pollaczek's method to single server queuing networks*, Archiv für Elektronik und Ubertragungstechnik, vol. 47, S.Hirzel Verlag, Stuttgart, 1993.

[LEG 94a] Le Gall P., *The overall sojourn time in tandem queues with identical successive service times and renewal input*, Stochastic Processes and their Applications, vol. 52, p. 165–178, Elsiever, 1994.

[LEG 94b] Le Gall P., *Traffic modeling in packet switched networks for single links,* Ann. Telecommun., vol. 49, n° 3–4, 1994.

[MAN 88] Manuel du CCITT: *Données de planification et méthodes de prévision. Manuel du GAS 10.* Volumes I and II, UIT, Genève, 1988.

[MAS 02] Massoni M., *Initiation aux statistiques descriptives avec Excel*, Vuibert, 2nd edition, 2002.

[MON 97] Monfort A., *Cours de statistique mathématique*, Economica, Collection Economie et Statistiques Avancées, 3rd edition, 1997.

[PAR 60] Parzen E., *Modern Probability Theory and its Application*, Wiley, New York, 1960.

[PAR 62] Parzen E., *Sochastic Processes*, Holden Day, 1962.

[POL 57] Pollaczek F., *Problèmes stochastiques posés par le phénomène de formation d'une queue d'attente à un guichet et par des phénomènes apparentés*, Paris, Gauthier Villars, 1957.

[POL 61] Pollaczek F., *Théorie analytique des problèmes stochastiques relatifs à un groupe de lignes téléphoniques avec attente*, Paris, Gauthier Villars, 1961.

[PUJ 02] Pujolle G., *Les réseaux*, 4th edition, Eyrolles, 2002.

[ROB 91] Roberts J., Virtamo J.T., *The superposition of periodic cell arrival processes in an ATM multiplexer*, IEEE Trans. on Comm., vol. 39, n° 2, 1991.

[ROB 96] Roberts J., Mocci U., Virtamo J. (dir.), *COST 242 Final report of Action*, Springer, 1996.

[ROB 94] Robertazzi T.G., *Computer Networks and Systems: Queueing Theory and Performance Evaluation*, Springer Verlag, 1994.

[ROS 97] Ross S.M., *Simulation*, Statistical modelling and decision sciences series, Academic Press, 2nd edition, 1997.

[SAB 91] Sabourin T., Fiche G., Ligeour M., *Overload control in a distributed system*, ITC 13, 1991,

[SAI 79] Saito T., Inose H., Hayashi S., *Evaluation of traffic carrying capability in one stage and two stage time division networks handling data with a variety of speed classes*, ITC 9, 1979.

[SCH 69] Schwob et Peyrache, *Traité de Fiabilité*, Masson, Paris, 1969.

[SHO 68] Shooman M.L., *Probabilistic reliability: an engineering approach*, McGraw-Hill, 1968.

[SYS 86] Syski R., *Introduction to Congestion Theory in Telephone Systems*, (2nd edition), Elsevier Ltd., 1986.

[SYS 93] Syski R., *Pollaczek's method in queueing theory, Queueing Theory*, R.Cruon, 1967 reedited in *Archiv für Elektronik und Ubertragungstechnik*, vol 47, S.Hirzel Verlag, Stuttgart, 1993.

[TAK 91] Takagi H., *Queueing Analysis. Vol. 1, vacations and priority systems*, North Holland, 1991.

[TAS 89] Tassi P., *Méthodes Statistiques*, Economica, Collection Economie et Statistiques Avancées, 2nd edition, Paris, 1989.

[VIN 84] Vinogradov O.P., *On the distribution of sojourn time in the tandem queues system with identical services times*, Proc 3rd ITC seminar, p. 449–450, Moscow, 1984.

[VIL 91] Villén M., Villén J., *RESTART: A method for accelerating rare event simulation*, ITC 13, Copenhagen, June 1991.

This page intentionally left blank

Index