ITU–D

Study Group 2
Question 16/2

Handbook
“TELETRAFFIC ENGINEERING”

Geneva, January 2005

PREFACE

This first edition of the *Teletraffic Engineering Handbook* has been worked out as a joint venture between the

- *ITU – International Telecommunication Union*
  `<http://www.itu.int>`, and the:
- *ITC – International Teletraffic Congress*
  `<http://www.i-teletraffic.org>`.

The handbook covers the basic theory of teletraffic engineering. The mathematical background required is elementary probability theory. The purpose of the handbook is to enable engineers to understand ITU–T recommendations on traffic engineering, evaluate tools and methods, and keep up-to-date with new practices. The book includes the following parts:

- Introduction: Chapter 1 – 2,
- Mathematical background: Chapter 3 – 6,
- Telecommunication loss models: Chapter 7 – 11,
- Data communication delay models: Chapter 12 – 14,
- Measurements: Chapter 15.

The purpose of the book is twofold: to serve both as a handbook and as a textbook. Thus the reader should, for example, be able to study chapters on loss models without studying the chapters on the mathematical background first.
The handbook is based on many years of experience in teaching the subject at the Technical University of Denmark and from ITU training courses in developing countries by the editor Villy B. Iversen. ITU-T Study Group 2 (Working Party 3/2) has reviewed Recommendations on traffic engineering. Many engineers from the international teletraffic community and students have contributed with ideas to the presentation. Supporting material, such as software, exercises, advanced material, and case studies, is available at `<http://www.com.dtu.dk/teletraffic>`, where comments and ideas will also be appreciated.

The handbook was initiated by the International Teletraffic Congress (ITC), Committee 3 (Developing countries and ITU matters), reviewed and adopted by ITU-D Study Group 2 in 2001. The Telecommunication Development Bureau thanks the International Teletraffic Congress, all Member States, Sector Members and experts, who contributed to this publication.

Hamadoun I. Touré
Director
Telecommunication Development Bureau
International Telecommunication Union

Notations

a	Carried traffic per source or per channel
A	Offered traffic $= A_o$
A_c	Carried traffic $= Y$
A_ℓ	Lost traffic
B	Call congestion
$\mathcal{B}$	Burstiness
c	Constant
C	Traffic congestion = load congestion
$\mathcal{C}_n$	Catalan's number
d	Slot size in multi-rate traffic
D	Probability of delay or
	Deterministic arrival or service process
E	Time congestion
$E_{1,n}(A) = E_1$	Erlang's B–formula = Erlang's 1. formula
$E_{2,n}(A) = E_2$	Erlang's C–formula = Erlang's 2. formula
F	Improvement function
g	Number of groups
h	Constant time interval or service time
$H(k)$	Palm–Jacobæus' formula
I	Inverse time congestion $I = 1/E$
$J_\nu(z)$	Modified Bessel function of order ν
k	Accessibility = hunting capacity
	Maximum number of customers in a queueing system
K	Number of links in a telecommuncation network or
	number of nodes in a queueing network
L	Mean queue length
$L_{kø}$	Mean queue length when the queue is greater than zero
$\mathcal{L}$	Random variable for queue length
m	Mean value (average) $= m_1$
m_i	i'th (non-central) moment
m'_i	i'th centrale moment
m_r	Mean residual life time
M	Poisson arrival process
n	Number of servers (channels)
N	Number of traffic streams or traffic types
$p(i)$	State probabilities, time averages
$p\{i, t \mid j, t_0\}$	Probability for state i at time t given state j at time t_0

$P(i)$	Cumulated state probabilities $P(i) = \sum_{x=-\infty}^{i} p(x)$
$q(i)$	Relative (non normalised) state probabilities
$Q(i)$	Cumulated values of $q(i)$: $Q(i) = \sum_{x=-\infty}^{i} q(x)$
Q	Normalisation constant
r	Reservation parameter (trunk reservation)
R	Mean response time
s	Mean service time
S	Number of traffic sources
t	Time instant
T	Random variable for time instant
U	Load function
v	Variance
V	Virtual waiting time
w	Mean waiting time for delayed customers
W	Mean waiting time for all customers
$\mathcal{W}$	Random variable for waiting time
x	Variable
X	Random variable
y	Arrival rate. Poisson process: $y = \lambda$
Y	Carried traffic
Z	Peakedness

α	Offered traffic per source
β	Offered traffic per idle source
γ	Arrival rate for an idle source
ε	Palm's form factor
ϑ	Lagrange-multiplicator
κ_i	i'th cumulant
λ	Arrival rate of a Poisson process
Λ	Total arrival rate to a system
μ	Service rate, inverse mean service time
$\pi(i)$	State probabilities, arriving customer mean values
$\psi(i)$	State probabilities, departing customer mean values
ϱ	Service ratio
σ^2	Variance, σ = standard deviation
τ	Time-out constant or constant time-interval

Contents

Chapter 1

Introduction to Teletraffic Engineering

Teletraffic theory is defined as *the application of probability theory to the solution of problems concerning planning, performance evaluation, operation, and maintenance of telecommunication systems.* More generally, teletraffic theory can be viewed as a discipline of planning where the tools (stochastic processes, queueing theory and numerical simulation) are taken from the disciplines of operations research.

The term *teletraffic* covers all kinds of *data communication traffic* and *telecommunication traffic*. The theory will primarily be illustrated by examples from telephone and data communication systems. The tools developed are, however, independent of the technology and applicable within other areas such as road traffic, air traffic, manufacturing and assembly belts, distribution, workshop and storage management, and all kinds of service systems.

The objective of teletraffic theory can be formulated as follows:

> *to make the traffic measurable in well defined units through mathematical models and to derive the relationship between grade-of-service and system capacity in such a way that the theory becomes a tool by which investments can be planned.*

The task of teletraffic theory is to design systems as cost effectively as possible with a predefined grade of service when we know the future traffic demand and the capacity of system elements. Furthermore, it is the task of teletraffic engineering to specify methods for controlling that the actual grade of service is fulfilling the requirements, and also to specify emergency actions when systems are overloaded or technical faults occur. This requires methods for forecasting the demand (for instance based on traffic measurements), methods for calculating the capacity of the systems, and specification of quantitative measures for the grade of service.

When applying the theory in practice, a series of decision problems concerning both short term as well as long term arrangements occur.

Short term decisions include a.o. the determination of the number of circuits in a trunk group, the number of operators at switching boards, the number of open lanes in the supermarket, and the allocation of priorities to jobs in a computer system.
Long term decisions include for example decisions concerning the development and extension of data- and telecommunication networks, the purchase of cable equipment, transmission systems etc.

The application of the theory in connection with design of new systems can help in comparing different solutions and thus eliminate non-optimal solutions at an early stage without having to build up prototypes.

1.1 Modelling of telecommunication systems

For the analysis of a telecommunication system, a model must be set up to describe the whole (or parts of) the system. This modelling process is fundamental especially for new applications of the teletraffic theory; it requires knowledge of *both* the technical system as well as the mathematical tools and the implementation of the model on a computer. Such a model contains three main elements (Fig. 1.1):

- the system structure,
- the operational strategy, and
- the statistical properties of the traffic.

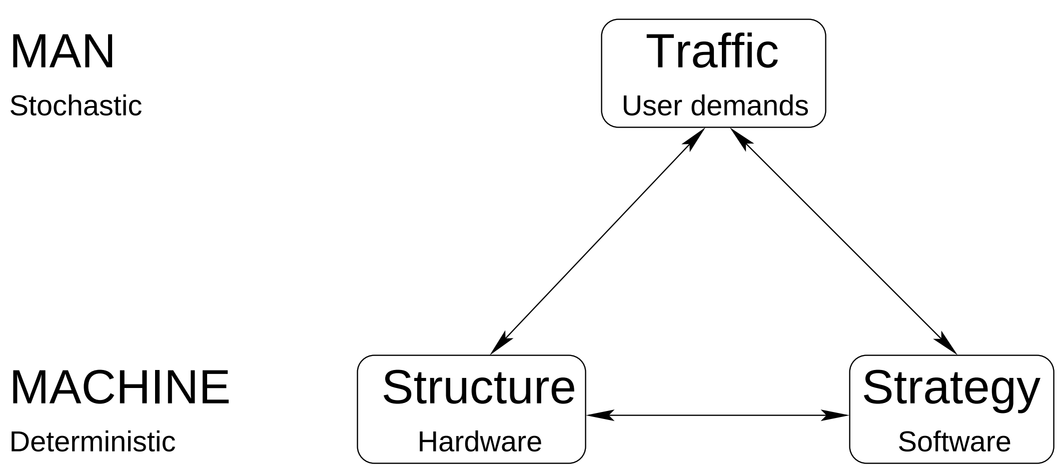


Figure 1.1: *Telecommunication systems are complex man/machine systems. The task of teletraffic theory is to configure optimal systems from knowledge of user requirements and habits.*

1.1.1 System structure

This part is technically determined and it is in principle possible to obtain any level of details in the description, e.g. at component level. Reliability aspects are stochastic as errors occur at random, and they will be dealt with as traffic with a high priority. The system structure is given by the physical or logical system which is described in manuals in every detail. In road traffic systems, roads, traffic signals, roundabouts, etc. make up the structure.

1.1.2 The operational strategy

A given physical system (for instance a roundabout in a road traffic system) can be used in different ways in order to adapt the traffic system to the demand. In road traffic, it is implemented with traffic rules and strategies which might be different for the morning and the evening traffic.

In a computer, this adaption takes place by means of the operation system and by operator interference. In a telecommunication system, strategies are applied in order to give priority to call attempts and in order to route the traffic to the destination. In Stored Program Controlled (*SPC*) telephone exchanges, the tasks assigned to the central processor are divided into classes with different priorities. The highest priority is given to accepted calls followed by new call attempts whereas routine control of equipment has lower priority. The classical telephone systems used *wired logic* in order to introduce strategies while in modern systems it is done by software, enabling more flexible and adaptive strategies.

1.1.3 Statistical properties of traffic

User demands are modelled by statistical properties of the traffic. Only by measurements on real systems is it possible to validate that the theoretical modelling is in agreement with reality. This process must necessarily be of an iterative nature (Fig. 1.2). A mathematical model is build up from a thorough knowledge of the traffic. Properties are then derived from the model and compared to measured data. If they are not in satisfactory accordance with each other, a new iteration of the process must take place.

It appears natural to split the description of the traffic properties into stochastic processes for arrival of call attempts and processes describing service (holding) times. These two processes are usually assumed to be mutually independent, meaning that the duration of a call is independent of the time the call arrived. Models also exists for describing the behaviour of users (subscribers) experiencing blocking, i.e. they are refused service and may make a new call attempt a little later (repeated call attempts). Fig. 1.3 illustrates the terminology usually applied in the teletraffic theory.

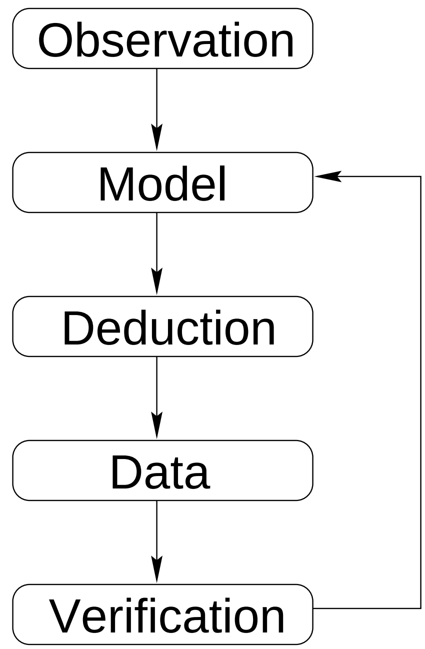


Figure 1.2: *Teletraffic theory is an inductive discipline. From observations of real systems we establish theoretical models, from which we derive parameters, which can be compared with corresponding observations from the real system. If there is agreement, the model has been validated. If not, then we have to elaborate the model further. This scientific way of working is called the research spiral.*

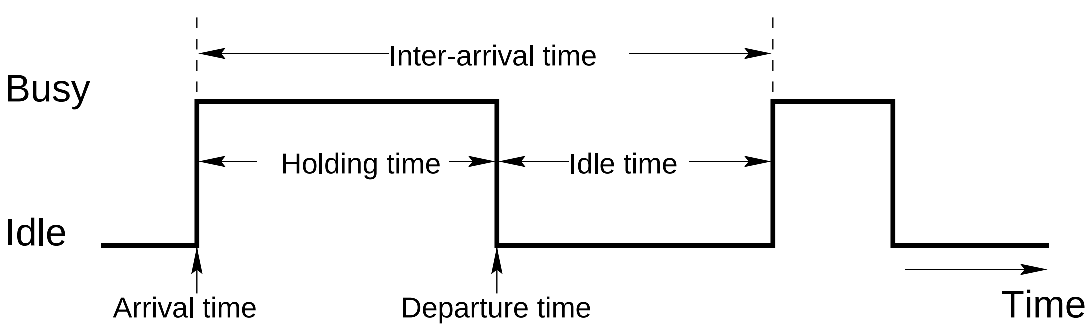


Figure 1.3: *Illustration of the terminology applied for a traffic process. Notice the difference between time intervals and instants of time. We use the terms arrival and call synonymously. The inter-arrival time, respectively the inter-departure time, are the time intervals between arrivals, respectively departures.*

1.1.4 Models

General requirements to a model are:

1. It must without major difficulty be possible to verify the model and it must be possible to determine the model parameters from observed data.

2. It must be feasible to apply the model for practical dimensioning.

We are looking for a description of for example the variations observed in the number of ongoing established calls in a telephone exchange, which vary incessantly due to calls being established and terminated. Even though common habits of subscribers imply that daily variations follows a predictable pattern, it is impossible to predict individual call attempts or duration of individual calls. In the description, it is therefore necessary to use statistical methods. We say that call attempt events take place according to a *stochastic process*, and the inter arrival time between call attempts is described by those probability distributions which characterise the stochastic process.

An alternative to a mathematical model is a simulation model or a physical model (prototype). In a computer *simulation model* it is common to use either collected data directly or to use artificial data from statistical distributions. It is however, more resource demanding to work with simulation since the simulation model is not general. Every individual case must be simulated. The development of a physical prototype is even more time and resource consuming than a simulation model.

In general mathematical models are therefore preferred but often it is necessary to apply simulation to develop the mathematical model. Sometimes prototypes are developed for ultimate testing.

1.2 Conventional telephone systems

This section gives a short description on what happens when a call arrives to a traditional telephone central. We divide the description into three parts: *structure, strategy and traffic*. It is common practice to distinguish between subscriber exchanges (access switches, local exchanges, *LEX*) and transit exchanges (*TEX*) due to the hierarchical structure according to which most national telephone networks are designed. Subscribers are connected to local exchanges or to access switches (concentrators), which are connected to local exchanges. Finally, transit switches are used to interconnect local exchanges or to increase the availability and reliability.

1.2.1 System structure

Here we consider a telephone exchange of the crossbar type. Even though this type is being taken out of service these years, a description of its functionality gives a good illustration on the tasks which need to be solved in a digital exchange. The equipment in a conventional telephone exchange consists of *voice paths* and *control paths.* (Fig. 1.4).

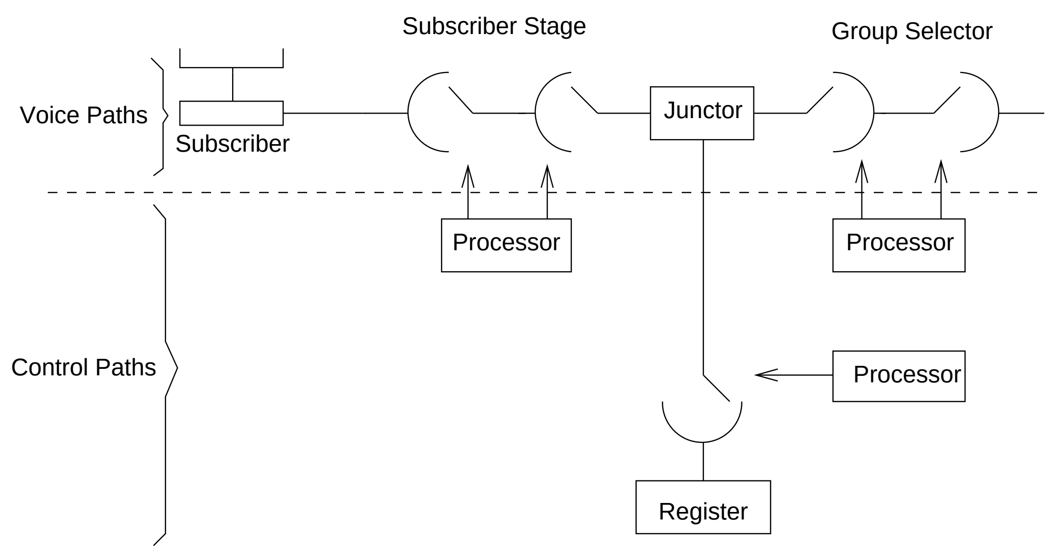


Figure 1.4: *Fundamental structure of a switching system.*

The voice paths are occupied during the whole duration of the call (in average three minutes) while the control paths only are occupied during the call establishment phase (in the range 0.1 to 1 s). The number of voice paths is therefore considerable larger than the number of control paths. The voice path is a connection from a given inlet (subscriber) to a given outlet. In a *space divided system* the voice paths consists of passive component (like relays, diodes or *VLSI* circuits). In a *time division system* the voice paths consist of specific time-slots within a frame. The control paths are responsible for establishing the connection. Normally, this happens in a number of stages where each stage is performed by a control device: a *microprocessor*, or a *register*.

Tasks of the control device are:

- Identification of the originating subscriber (who wants a connection (inlet)).
- Reception of the digit information (address, outlet).
- Search after an idle connection between inlet and outlet.
- Establishment of the connection.
- Release of the connection (performed sometimes by the voice path itself).

In addition the charging of the calls must be taken care of. In conventional exchanges the control path is build up on relays and/or electronic devices and the logical operations are done by *wired logic*. Changes in the functions require physical changes and they are difficult and expensive
In digital exchanges the control devices are processors. The logical functions are carried out by software, and changes are considerable more easy to implement. The restrictions are far less constraining, as well as the complexity of the logical operations compared to the wired logic. *Software controlled exchanges* are also called *SPC-systems* (Stored Program Controlled systems).

1.2.2 User behaviour

We consider a conventional telephone system. When an *A-subscriber* initiates a call, the hook is taken off and the wired pair to the subscriber is short-circuited. This triggers a relay at the exchange. The relay identifies the subscriber and a micro processor in the subscriber stage choose an idle *cord*. The subscriber and the cord is connected through a switching stage. This terminology originates from a the time when a manual operator by means of the cord was connected to the subscriber. A manual operator corresponds to a register. The cord has three outlets.

A *register* is through another switching stage coupled to the cord. Thereby the subscriber is connected to a register (register selector) via the cord. This phase takes less than one second.

The register sends the dial tone to the subscriber who dials the desired telephone number of the *B-subscriber*, which is received and maintained by the register. The duration of this phase depends on the subscriber.

A microprocessor analyses the digit information and by means of a group selector establishes a connection through to the desired subscriber. It can be a subscriber at same exchange, at a neighbour exchange or a remote exchange. It is common to distinguish between exchanges to which a direct link exists, and exchanges for which this is not the case. In the latter case a connection must go through an exchange at a higher level in the hierarchy. The digit information is delivered by means of a code transmitter to the code receiver of the desired exchange which then transmits the information to the registers of the exchange.

The register has now fulfilled its obligation and is released so it is idle for the service of other call attempts. The microprocessors work very fast (around 1–10 ms) and independently of the subscribers. The cord is occupied during the whole duration of the call and takes control of the call when the register is released. It takes care of different types of signals (busy, reference etc), pulses for charging, and release of the connection when the call is put down, etc.

It happens that a call does not pass on as planned. The subscriber may make an error, suddenly hang up, etc. Furthermore, the system has a limited capacity. This will be dealt with in Chap. 2. Call attempts towards a subscriber take place in approximately the same way. A code receiver at the exchange of the B-subscriber receives the digits and a connection is set up through the group switching stage and the local switch stage through the B-subscriber with use of the registers of the receiving exchange.

1.2.3 Operation strategy

The voice path normally works as loss systems while the control path works as delay systems (Chap. 2).

If there is not both an idle cord as well as an idle register then the subscriber will get no dial tone no matter how long he/she waits. If there is no idle outlet from the exchange to the desired B-subscriber a busy tone will be sent to the calling A-subscriber. Independently of any additional waiting there will not be established any connection.

If a microprocessor (or all microprocessors of a specific type when there are several) is busy, then the call will wait until the microprocessor becomes idle. Due to the very short holding time then waiting time will often be so short that the subscribers do not notice anything. If several subscribers are waiting for the same microprocessor, they will normally get service in random order independent of the time of arrival.

The way by which control devices of the same type and the cords share the work is often *cyclic*, such that they get approximately the same number of call attempts. This is an advantage since this ensures the same amount of wear and since a subscriber only rarely will get a defect cord or control path again if the call attempt is repeated.

If a control path is occupied more than a given time, a forced disconnection of the call will take place. This makes it impossible for a single call to block vital parts of the exchange, e.g. a register. It is also only possible to generate the ringing tone for a limited duration of time towards a B-subscriber and thus block this telephone a limited time at each call attempt. An exchange must be able to operate and function independently of subscriber behaviour.

The cooperation between the different parts takes place in accordance to strictly and well defined rules, called protocols, which in conventional systems is determined by the wired logic and in software control systems by software logic.

The digital systems (e.g. *ISDN* = Integrated Services Digital Network, where the whole telephone system is digital from subscriber to subscriber ($2 \cdot B + D = 2 \times 64 + 16$ *Kbps* per subscriber), *ISDN* = *N-ISDN* = Narrowband *ISDN*) of course operates in a way different from the conventional systems described above. However, the fundamental teletraffic tools for evaluation are the same in both systems. The same also covers the future broadband systems *B–ISDN* which will be based on *ATM* = Asynchronous Transfer Mode.

1.3 Communication networks

There exists different kinds of communications networks:, telephone networks, telex networks, data networks, Internet, etc. Today the telephone network is dominating and physically other networks will often be integrated in the telephone network. In future digital networks it is the plan to integrate a large number of services into the same network (*ISDN*, *B-ISDN*).

1.3.1 The telephone network

The telephone network has traditionally been build up as a hierarchical system. The individual subscribers are connected to a subscriber switch or sometimes a local exchange (*LEX*). This part of the network is called the access network. The subscriber switch is connected to a specific main local exchange which again is connected to a transit exchange (*TEX*) of which there usually is at least one for each area code. The transit exchanges are normally connected into a mesh structure. (Fig. 1.5). These connections between the transit exchanges are called the *hierarchical transit network*. There exists furthermore connections between two local exchanges (or subscriber switches) belonging to different transit exchanges (local exchanges) if the traffic demand is sufficient to justify it.

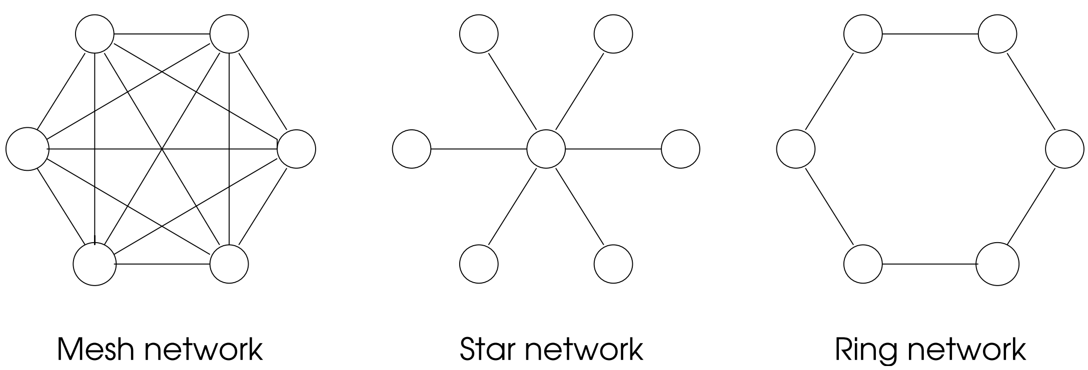


Figure 1.5: *There are three basic structures of networks: mesh, star and ring. Mesh networks are applicable when there are few large exchanges (upper part of the hierarchy, also named polygon network), whereas star networks are proper when there are many small exchanges (lower part of the hierarchy). Ring networks are applied for example in fibre optical systems.*

A connection between two subscribers in different transit areas will normally pass the following exchanges:

$$USER \rightarrow LEX \rightarrow TEX \rightarrow TEX \rightarrow LEX \rightarrow USER$$

The individual transit trunk groups are based on either analogue or digital transmission systems, and multiplexing equipment is often used.

Twelve analogue channels of 3 kHz each make up one first order *bearer frequency system* (frequency multiplex), while 32 digital channels of 64 Kbps each make up a first order *PCM-system* of 2.048 Mbps (pulse-code-multiplexing, time multiplexing).

The 64 Kbps are obtained from a sampling of the analogue signal at a rate of 8 kHz and an amplitude accuracy of 8 bit. Two of the 32 channels in a *PCM system* are used for signalling and control.

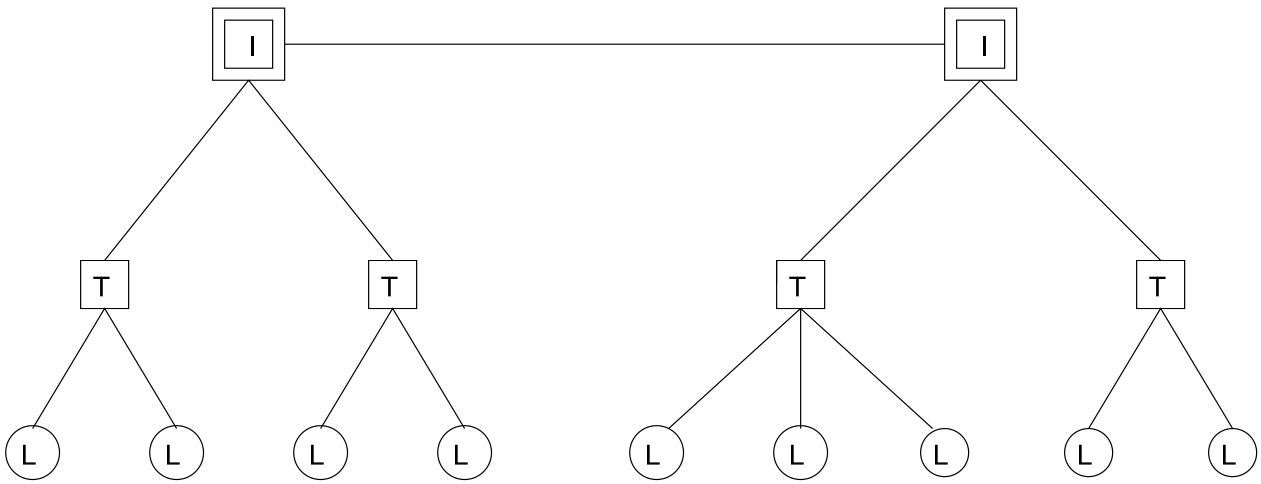


Figure 1.6: *In a telecommunication network all exchanges are typically arranged in a three-level hierarchy. Local-exchanges or subscriber-exchanges (L), to which the subscribers are connected, are connected to main exchanges (T), which again are connected to inter-urban exchanges (I). An inter-urban area thus makes up a star network. The inter-urban exchanges are interconnected in a mesh network. In practice the two network structures are mixed, because direct trunk groups are established between any two exchanges, when there is sufficient traffic. In the future Danish network there will only be two levels, as T and I will be merged.*

Due to reliability and security there will almost always exist at least two disjoint paths between any two exchanges and the strategy will be to use the cheapest connections first. The hierarchy in the Danish digital network is reduced to two levels only. The upper level with transit exchanges consists of a fully connected meshed network while the local exchanges and subscriber switches are connected to two or three different transit exchanges due to security and reliability.

The telephone network is characterised by the fact that before any two subscribers can communicate a full two-way (duplex) *connection* must be created, and the connection exists during the whole duration of the communication. This property is referred to as the telephone network being *connection oriented* as distinct from for example the Internet which is connection-less. Any network applying for example *line–switching* or *circuit–switching* is connection oriented. A packet switching network may be either connection oriented (for example virtual connections in *ATM*) or connection-less. In the discipline of network planning, the objective is to optimise network structures and traffic routing under the consideration of traffic demands, service and reliability requirement etc.

Example 1.3.1: VSAT-networks
VSAT-networks (Maral, 1995 [76]) are for instance used by multi-national organisations for transmission of speech and data between different divisions of news-broadcasting, in catastrophic situations, etc. It can be both point-to point connections and point to multi-point connections (distribution and broadcast). The acronym *VSAT* stands for Very Small Aperture Terminal (Earth station) which is an antenna with a diameter of 1.6–1.8 meter. The terminal is cheap and mobile. It is thus possible to bypass the public telephone network. The signals are transmitted from a *VSAT* terminal via a satellite towards another *VSAT* terminal. The satellite is in a fixed position 35 786 km above equator and the signals therefore experiences a propagation delay of around 125 ms per hop. The available bandwidth is typically partitioned into channels of 64 Kbps, and the connections can be one-way or two-ways.

In the simplest version, all terminals transmit directly to all others, and a *full mesh network* is the result. The available bandwidth can either be assigned in advance (*fixed assignment*) or dynamically assigned (*demand assignment*). Dynamical assignment gives better utilisation but requires more control.

Due to the small parabola (antenna) and an attenuation of typically 200 dB in each direction, it is practically impossible to avoid transmission error, and error correcting codes and possible retransmission schemes are used. A more reliable system is obtained by introducing a main terminal (a *hub*) with an antenna of 4 to 11 meters in diameter. A communication takes place through the hub. Then both hops (*VSAT* → *hub* and *hub* → *VSAT*) become more reliable since the hub is able to receive the weak signals and amplify them such that the receiving *VSAT* gets a stronger signal. The price to be paid is that the propagation delay now is 500 ms. The hub solution also enables centralised control and monitoring of the system. Since all communication is going through the hub, the network structure constitutes a star topology. □

1.3.2 Data networks

Data network are sometimes engineered according to the same principle as the telephone network except that the duration of the connection establishment phase is much shorter. Another kind of data network is given in the so-called *packet distribution network*, which works according to the *store-and-forward* principle (see Fig. 1.7). The data to be transmitted are not sent directly from transmitter to receiver in one step but in steps from exchange to exchange. This may create delays since the exchanges which are computers work as delay systems (connection-less transmission).

If the packet has a maximum fixed length, the network is denoted *packet switching* (e.g. *X.25* protocol). In *X.25* a message is segmented into a number of packets which do not necessarily follow the same path through the network. The protocol header of the packet contains a sequence number such that the packets can be arranged in correct order at the receiver. Furthermore error correction codes are used and the correctness of each packet is checked at the receiver. If the packet is correct an acknowledgement is sent back to the preceding node which now can delete its copy of the packet. If the preceding node does not receive

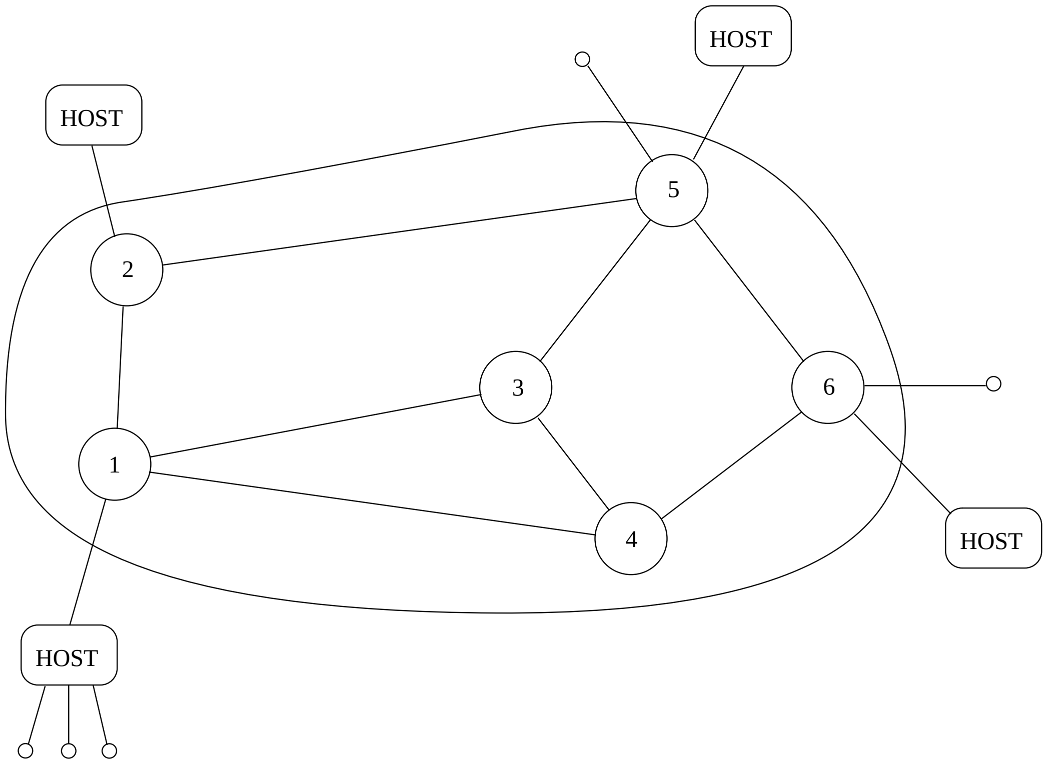


Figure 1.7: *Datagram network: Store- and forward principle for a packet switching data network.*

any acknowledgement within some given time interval a new copy of the packet (or a whole frame of packets) are retransmitted. Finally, there is a control of the whole message from transmitter to receiver. In this way a very reliable transmission is obtained. If the whole message is sent in a single packet, it is denoted *message–switching*.

Since the exchanges in a data network are computers, it is feasible to apply advanced strategies for traffic routing.

1.3.3 Local Area Networks (LAN)

Local area networks are a very specific but also very important type of data network where all users through a computer are attached to the same digital transmission system, e.g. a coaxial cable. Normally, only one user at a time can use the transmission medium and get some data transmitted to another user. Since the transmission system has a large capacity compared to the demand of the individual users, a user experiences the system as if he is the only user. There exist several types of local area networks. Applying adequate strategies for the medium access control (*MAC*) principle, the assignment of capacity in case of many users competing for transmission is taken care of. There exist two main types of Local Area Networks: *CSMA/CD* (Ethernet) and *token networks*. The *CSMA/CD* (Carrier Sense

Multiple Access/Collision Detection) is the one most widely used. All terminals are all the time listening to the transmission medium and know when it is idle and when it is occupied. At the same time a terminal can see which packets are addressed to the terminal itself and therefore needs to be stored. A terminal wanting to transmit a packet transmit it if the medium is idle. If the medium is occupied the terminal wait a random amount of time before trying again. Due to the finite propagation speed, it is possible that two (or even more) terminals starts transmission within such a short time interval so that two or more messages collide on the medium. This is denoted as a *collision*. Since all terminals are listening all the time, they can immediately detect that the transmitted information is different from what they receive and conclude that a collision has taken place (*CD* = Collision Detection). The terminals involved immediately stops transmission and try again a random amount of time later (back-off).

In local area network of the token type, it is only the terminal presently possessing the token which can transmit information. The token is rotating between the terminals according to predefined rules.

Local area networks based on the *ATM* technique are also in operation. Furthermore, wireless *LAN*s are becoming common. The propagation is negligible in local area networks due to small geographical distance between the users. In for example a satellite data network the propagation delay is large compared to the length of the messages and in these applications other strategies than those used in local area networks are used.

1.4 Mobile communication systems

A tremendous expansion is seen these years in mobile communication systems where the transmission medium is either analogue or digital radio channels (wireless) in contrast to the convention cable systems. The electro magnetic frequency spectrum is divided into different bands reserved for specific purposes. For mobile communications a subset of these bands are reserved. Each band corresponds to a limited number of radio telephone channels, and it is here the limited resource is located in mobile communication systems. The optimal utilisation of this resource is a main issue in the cellular technology. In the following subsection a representative system is described.

1.4.1 Cellular systems

Structure. When a certain geographical area is to be supplied with mobile telephony, a suitable number of base stations must be put into operation in the area. A base station is an antenna with transmission/receiving equipment or a radio link to a mobile telephone exchange (*MTX*) which are part of the traditional telephone network. A mobile telephone exchange is common to all the base stations in a given traffic area. Radio waves are damped when they propagate in the atmosphere and a base station is therefore only able to cover a limited geographical area which is called a cell (not to be confused with *ATM*–cells). By transmitting the radio waves at adequate power it is possible to adapt the coverage area such that all base stations covers exactly the planned traffic area without too much overlapping between neighbour stations. It is not possible to use the same radio frequency in two neighbour base stations but in two base stations without a common border the same frequency can be used thereby allowing the channels to be reused.

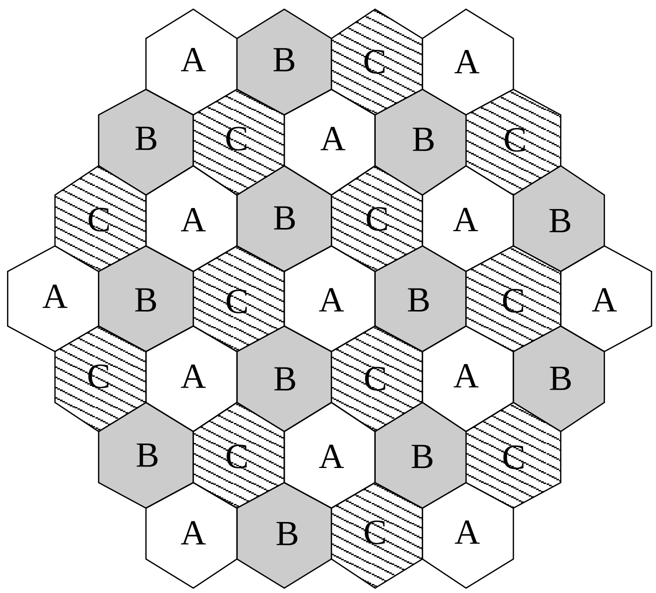


Figure 1.8: *Cellular mobile communication system. By dividing the frequencies into 3 groups (A, B and C) they can be reused as shown.*

In Fig. 1.8 an example is shown. A certain number of channels per cell corresponding to a given traffic volume is thereby made available. The size of the cell will depend on the traffic volume. In densely populated areas as major cities the cells will be small while in sparsely populated areas the cells will be large.
Channel allocation is a very complex problem. In addition to the restrictions given above, a number of other also exist. For example, there has to be a certain distance (number of channels) between two channels on the same base station (neighbour channel restriction) and to avoid interference also other restrictions exist.

Strategy. In mobile telephone systems a database with information about all the subscriber has to exist. Any subscriber is either active or passive corresponding to whether the radio telephone is switched on or off. When the subscriber turns on the phone, it is automatically assigned to a so-called *control channel* and an identification of the subscriber takes place. The control channel is a radio channel used by the base station for control. The remaining channels are *traffic channels*

A call request towards a mobile subscriber (B-subscriber) takes place the following way. The mobile telephone exchange receives the call from the other subscriber (A-subscriber, fixed or mobile). If the B-subscriber is passive (handset switched off) the A-subscriber is informed that the B-subscriber is not available. Is the B-subscriber active, then the number is put out on all control channels in the traffic area. The B-subscriber recognises his own number and informs through the control channel in which cell (base station) he is in. If an idle traffic channel exists it is allocated and the *MTX* puts up the call.

A call request from a mobile subscriber (A-subscriber) is initiated by the subscriber shifting from the control channel to a traffic channel where the call is established. The first phase with reading in the digits and testing the availability of the B-subscriber is in some cases performed by the control channel (common channel signalling)

A subscriber is able to move freely within his own traffic area. When moving away from the base station this is detected by the *MTX* which constantly monitor the signal to noise ratio and the *MTX* moves the call to another base station and to another traffic channel with better quality when this is required. This takes place automatically by cooperation between the *MTX* and the subscriber equipment normally without being noticed by the subscriber. This operation is called *hand over*, and of course requires the existence of an idle traffic channel in the new cell. Since it is improper to interrupt an existing call, hand-over calls are given higher priorities than new calls. This strategy can be implemented by reserving one or two idle channels for hand-over calls.

When a subscriber is leaving its traffic area, so-called *roaming* will take place. The *MTX* in the new area is from the identity of the subscriber able to locate the home *MTX* of the subscriber. A message to the home *MTX* is forwarded with information on the new position. Incoming calls to the subscriber will always go to the home *MTX* which will then route the call to the new *MTX*. Outgoing calls will be taken care of the usual way.

A widespread digital wireless system is *GSM*, which can be used throughout Western Europe. *The International Telecommunication Union* is working towards a global mobile system *UPC* (*Universal Personal Communication*), where subscribers can be reached worldwide (*IMT2000*).

Paging systems are primitive one-way systems. *DECT*, Digital European Cord-less Telephone, is a standard for wireless telephones. They can be applied locally in companies, business centres etc. In the future equipment which can be applied both for *DECT* and *GSM* will come up. Here *DECT* corresponds to a system with very small cells while *GSM* is a system with larger cells.

Satellite communication systems are also being planned in which the satellite station corresponds to a base station. The first such system *Iridium*, consisted of 66 satellites such that more than one satellite always were available at any given location within the geographical range of the system. The satellites have orbits only a few hundred kilometres above the Earth. Iridium was unsuccessful, but newer systems such as the *Inmarsat* system is now in use.

1.5 ITU recommendations on traffic engineering

The following section is based on *ITU–T* draft Recommendation E.490.1: *Overview of Recommendations on traffic engineering*. See also (Villen, 2002 [100]). *The International Telecommunication Union* (*ITU*) is an organisation sponsored by the *United Nations* for promoting international telecommunications. It has three sectors:

- the Telecommunication Standardisation Sector (*ITU–T*),
- the Radio communication Sector (*ITU–R*), and
- the Telecommunication Development Sector (*ITU–D*).

The primary function of the *ITU–T* is to produce international standards for telecommunications. The standards are known as recommendations. Although the original task of *ITU–T* was restricted to facilitate international inter-working, its scope has been extended to cover national networks, and the *ITU–T* recommendations are nowadays widely used as de facto national standards and as references.

The aim of most recommendations is to ensure compatible inter-working of telecommunication equipment in a multi-vendor and multi-operator environment. But there are also recommendations that advice on best practices for operating networks. Included in this group are the recommendations on traffic engineering.

The *ITU–T* is divided into Study Groups. *Study Group 2* (*SG2*) is responsible for *Operational Aspects of Service Provision Networks and Performance*. Each Study Group is divided into Working Parties. *Working Party 3* of *Study Group 2* (*WP 3/2*) is responsible for *Traffic Engineering*.

1.5.1 Traffic engineering in the ITU

Although Working Party 3/2 has the overall responsibility for traffic engineering, some recommendations on traffic engineering or related to it have been (or are being) produced by other Groups. Study Group 7 deals in the *X* Series with traffic engineering for data communication networks, Study Group 11 has produced some recommendations (*Q* Series) on traffic aspects related to system design of digital switches and signalling, and some recommendations of the *I* Series, prepared by Study Group 13, deal with traffic aspects related to network architecture of *N*- and *B-ISDN* and *IP*– based networks. Within Study Group 2, Working Party 1 is responsible for the recommendations on routing and Working Party 2 for the Recommendations on network traffic management.

This section will focus on the recommendations produced by Working Party 3/2. They are in the *E* Series (numbered between E.490 and E.799) and constitute the main body of *ITU–T* recommendations on traffic engineering.

The Recommendations on traffic engineering can be classified according to the four major traffic engineering tasks:

- Traffic demand characterisation;
- Grade of Service (GoS) objectives;
- Traffic controls and dimensioning;
- Performance monitoring.

The interrelation between these four tasks is illustrated in Fig. 1. The initial tasks in traffic engineering are to characterise the traffic demand and to specify the *GoS* (or performance) objectives. The results of these two tasks are input for dimensioning network resources and for establishing appropriate traffic controls. Finally, performance monitoring is required to check if the *GoS* objectives have been achieved and is used as a feedback for the overall process.

Secs. 1.5.2, 1.5.3, 1.5.4, 1.5.5 describe each of the above four tasks. Each section provides an overall view of the respective task and summarises the related recommendations. Sec. 1.5.6 summarises a few additional Recommendations as their scope do not match the items considered in the classification Sec. 1.5.7 describes the current work program and Sec. 1.5.8 states some conclusions.

1.5.2 Traffic demand characterisation

Traffic characterisation is done by means of models that approximate the statistical behaviour of network traffic in large population of users. Traffic models adopt simplifying assumptions concerning the complicated traffic processes. Using these models, traffic demand is characterised by a limited set of parameters (mean, variance, index of dispersion of counts, etc). Traffic modelling basically involves the identification of what simplifying assumptions can be made and what parameters are relevant from viewpoint of of the impact of traffic demand on network performance.

Traffic measurements are conducted to validate these models, with modifications being made when needed. Nevertheless, as the models do not need to be modified often, the purpose of traffic measurements is usually to estimate the values that the parameters defined in the traffic models take at each network segment during each time period.

As a complement to traffic modelling and traffic measurements, traffic forecasting is also required given that, for planning and dimensioning purposes, it is not enough to characterise

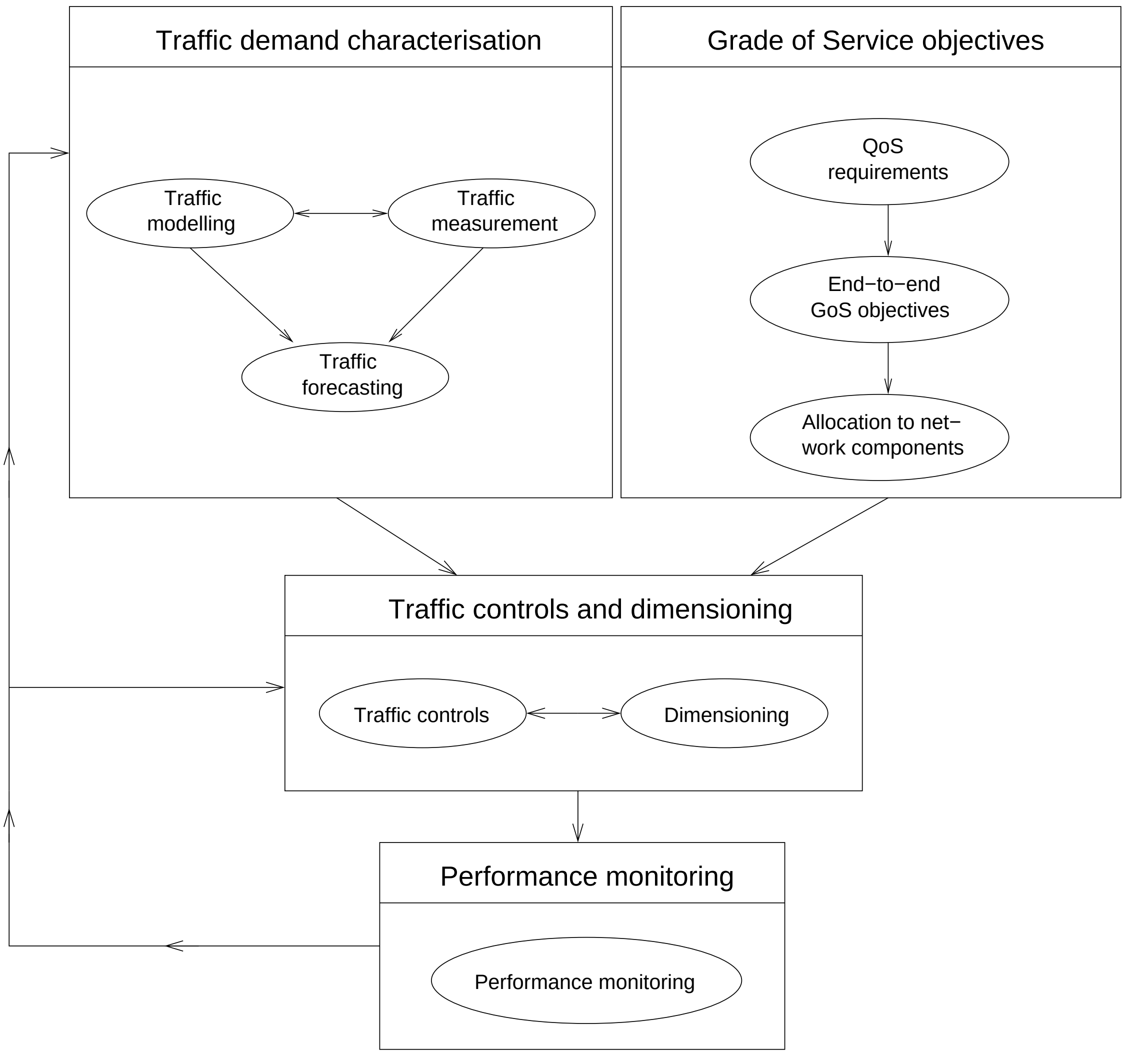


Figure 1.9: *Traffic engineering tasks.*

present traffic demand, but it is necessary to forecast traffic demands for the time period foreseen in the planning process.

Thus the *ITU* recommendations cover these three aspects of traffic characterisation: traffic modelling, traffic measurements, and traffic forecasting.

Traffic modelling

Recommendations on traffic modelling are listed in Tab. 1.1. There are no specific recommendations on traffic modelling for the classical circuit-switched telephone network. The only service provided by this network is telephony given other services, as fax, do not have a significant impact on the total traffic demand. Every call is based on a single 64 Kbps point-to-point bi-directional symmetric connection. Traffic is characterised by call rate and mean holding time at each origin-destination pair. Poissonian call arrival process (for first-choice routes) and negative exponential distribution of the call duration are the only assumptions needed. These assumptions are directly explained in the recommendations on dimensioning.

Rec.	Date	Title
E.711	10/92	User demand modelling
E.712	10/92	User plane traffic modelling
E.713	10/92	Control plane traffic modelling
E.716	10/96	User demand modelling in Broadband-ISDN
E.760	03/00	Terminal mobility traffic modelling

Table 1.1: *Recommendations on traffic modelling.*

The problem is much more complex in *N*- and *B-ISDN* and in *IP*–based network. There are more variety of services, each with different characteristics, different call patterns and different QoS requirements. **Recommendations E.711 and E.716** explain how a call, in *N–ISDN* and *B–ISDN* respectively, must be characterised by a set of connection characteristics (or call attributes) and by a call pattern.

Some examples of connection characteristics are the following: information transfer mode (circuit-switched or packet switched), communication configuration (point-to-point, multi-point or broadcast), transfer rate, symmetry (uni-directional, bi-directional symmetric or bi-directional asymmetric), QoS requirements, etc.

The call pattern is defined in terms of the sequence of events occurred along the call and of the times between these events. It is described by a set of traffic variables, which are expressed as statistical variables, that is, as moments or percentiles of distributions of random variables indicating number of events or times between events. The traffic variables can be classified into call-level (or connection-level) and packet-level (or transaction-level, in *ATM* cell-level) traffic variables.

The call-level traffic variables are related to events occurring during the call set-up and release phases. Examples are the mean number of re-attempts in case of non-completion and mean call-holding time.

The packet-level traffic variables are related to events occurring during the information transfer phase and describe the packet arrival process and the packet length. Recommendation E.716 describes a number of different approaches for defining packet-level traffic variables.

Once each type of call has been modelled, the user demand is characterised, according to E.711 and E.716, by the arrival process of calls of each type. Based on the user demand characterisation made in Recommendations E.711 and E.716, **Recommendations E.712 and E.713** explain how to model the traffic offered to a group of resources in the user plane and the control plane, respectively.

Finally, **Recommendation E.760** deals with the problem of traffic modelling in mobile networks where not only the traffic demand per user is random but also the number of users being served at each moment by a base station or by a local exchange. The recommendation provides methods to estimate traffic demand in the coverage area of each base station and mobility models to estimate hand-over and location updating rates.

Traffic measurements

Recommendations on traffic measurements are listed in Tab. 1.2. As indicated in the table, many of them cover both traffic and performance measurements. These recommendations can be classified into those on general and operational aspects (E.490, E.491, E.502 and E.503), those on technical aspects (E.500 and E.501) and those specifying measurement requirements for specific networks (E.502, E.505 and E.745). Recommendation E.743 is related to the last ones, in particular to Recommendation E.505.

Let us start with the recommendations on general and operational aspects. **Recommendation E.490** is an introduction to the series on traffic and performance measurements. It contains a survey of all these recommendations and explains the use of measurements for short term (network traffic management actions), medium term (maintenance and reconfiguration) and long term (network extensions).

Recommendation E.491 points out the usefulness of traffic measurements by destination for network planning purposes and outlines two complementary approaches to obtain them: call detailed records and direct measurements.

Recommendations E.504 describes the operational procedures needed to perform measurements: tasks to be made by the operator (for example to define output routing and scheduling of measured results) and functions to be provided by the system supporting the man-machine interface.

Once the measurements have been performed, they have to be analysed. **Recommendation E.503** gives an overview of the potential application of the measurements and describes the operational procedures needed for the analysis.

Let us now describe Recommendations E.500 and E.501 on general technical aspects. **Recommendation E.500** states the principles for traffic intensity measurements. The traditional concept of busy hour, which was used in telephone networks, cannot be extended to modern multi-service networks. Thus Recommendation E.500 provides the criteria to choose the length of the read-out period for each application. These criteria can be summarised as

Rec.	Date	Title
E.490*	06/92	Traffic measurement and evaluation - general survey
E.491	05/97	Traffic measurement by destination
E.500	11/98	Traffic intensity measurement principles
E.501	05/97	Estimation of traffic offered in the network
E.502*	02/01	Traffic measurement requirements for digital telecommunication exchanges
E.503*	06/92	Traffic measurement data analysis
E.504*	11/88	Traffic measurement administration
E.505*	06/92	Measurements of the performance of common channel signalling network
E.743	04/95	Traffic measurements for SS No. 7 dimensioning and planning
E.745*	03/00	Cell level measurement requirements for the *B-ISDN*

Table 1.2: *Recommendations on traffic measurements. Recommendations marked * cover both traffic and performance measurements.*

follows:

a) To be large enough to obtain confident measurements: the average traffic intensity in a period (t_1, t_2) can be considered a random variable with expected value A. The measured traffic intensity $A(t_1, t_2)$ is a sample of this random variable. As $t_2 - t_1$ increases, $A(t_1, t_2)$ converges to A. Thus the read-out period length $t_2 - t_1$ must be large enough such that $A(t_1, t_2)$ lies within a narrow confidence interval about A.
An additional reason to choose large read-out periods is that it may not be worth the effort to dimension resources for very short peak traffic intervals.

b) To be short enough so that the traffic intensity process is approximately stationary during the period, i.e. that the actual traffic intensity process can be approximated by a stationary traffic intensity model. Note that in the case of bursty traffic, if a simple traffic model (e.g. Poisson) is being used, criterion (b) may lead to an excessively short read-out period incompatible with criterion (a). In these cases alternative models should be used to obtain longer read-out period.

Recommendation E.500 also advises on how to obtain the daily peak traffic intensity over the measured read-out periods. It provides the method to derive the *normal load* and *high load* traffic intensities for each month and, based on them, the *yearly representative values* (YRV) for *normal and high loads*.

As offered traffic is required for dimensioning while only carried traffic is obtained from measurements, **Recommendation E.501** provides methods to estimate the traffic offered to a circuit group and the origin-destination traffic demand based on circuit group measurements. For the traffic offered to a circuit group, the recommendation considers both circuit groups with only-path arrangement, and circuit groups belonging to a high-usage/final circuit group arrangement. The repeated call attempts phenomenon is taken into account in the estimation. Although the recommendation only refers to circuit-switched networks with single-rate connections, some of the methods provided can be extended to other types of networks. Also, even though the problem may be much more complex in multi-service networks, advanced exchanges typically provide, in addition to circuit group traffic measurements, other measurements such as the number of total, blocked, completed and successful call attempts per service and per origin-destination pair, which may help to estimate offered traffic.

The third group of recommendations on measurements includes **Recommendations E.502, E.505 and E.745** which specify traffic and performance measurement requirements in *PSTN* and *N-ISDN* exchanges (E.502), *B-ISDN* exchanges (E.745) and nodes of *SS No. 7 Common Channel Signalling Networks* (E.505).

Finally, Recommendation E.743 is complementary to E.505. It identifies the subset of the measurements specified in Recommendation E.505 that are useful for SS No. 7 dimensioning and planning, and explains how to derive the input required for these purposes from the performed measurements.

Traffic forecasting

Traffic forecasting is necessary both for strategic studies, such as to decide on the introduction of a new service, and for network planning, that is, for the planning of equipment plant investments and circuit provisioning. The Recommendations on traffic forecasting are listed in Tab. 1.3. Although the title of the first two refers to international traffic, they also apply to the traffic within a country.

Recommendations E.506 and E.507 deal with the forecasting of traditional services for which there are historical data. **Recommendation E.506** gives guidance on the prerequisites for the forecasting: base data, including not only traffic and call data but also economic, social and demographic data are of vital importance. As the data series may be incomplete, strategies are recommended for dealing with missing data. Different forecasting approaches are presented: direct methods, based on measured traffic in the reference period, versus composite method based on accounting minutes, and top-down versus bottom-up procedures.

Recommendation E.507 provides an overview of the existing mathematical techniques for forecasting: curve-fitting models, autoregressive models, autoregressive integrated moving average (*ARIMA*) models, state space models with Kalman filtering, regression models and econometric models. It also describes methods for the evaluation of the forecasting models

Rec.	Date	Title
E.506	06/92	Forecasting international traffic
E.507	11/88	Models for forecasting international traffic
E.508	10/92	Forecasting new telecommunication services

Table 1.3: *Recommendations on traffic forecasting.*

and for the choice of the most appropriate one in each case, depending on the available data, length of the forecast period, etc.

Recommendation E.508 deals with the forecasting of new telecommunication services for which there are no historical data. Techniques such as market research, expert opinion and sectorial econometrics are described. It also advises on how to combine the forecasts obtained from different techniques, how to test the forecasts and how to adjust them when the service implementation starts and the first measurements are taken.

1.5.3 Grade of Service objectives

Grade of Service (*GoS*) is defined in Recommendations E.600 and E.720 as a number of traffic engineering parameters to provide a measure of adequacy of plant under specified conditions; these *GoS* parameters may be expressed as probability of blocking, probability of delay, etc. Blocking and delay are caused by the fact that the traffic handling capacity of a network or of a network component is finite and the demand traffic is stochastic by nature.

GoS is the traffic related part of network performance (*NP*), defined as the ability of a network or network portion to provide the functions related to communications between users. Network performance does not only cover *GoS* (also called trafficability performance), but also other non-traffic related aspects as dependability, transmission and charging performance.

NP objectives and in particular *GoS* objectives are derived from *Quality of Service* (*QoS*) requirements, as indicated in Fig. 1.9. *QoS* is a collective of service performances that determine the degree of satisfaction of a user of a service. *QoS* parameters are user oriented and are described in network independent terms. *NP* parameters, while being derived from them, are network oriented, i.e. usable in specifying performance requirements for particular networks. Although they ultimately determine the (user observed) *QoS*, they do not necessarily describe that quality in a way that is meaningful to users.

QoS requirements determine end-to-end *GoS* objectives. From the end-to-end objectives, a partition yields the *GoS* objectives for each network stage or network component. This partition depends on the network operator strategy. Thus *ITU* recommendations only specify the partition and allocation of GoS objectives to the different networks that may have to cooperate to establish a call (for example originating national network, international network and terminating national network in an international call).

In order to obtain an overview of the network under consideration and to facilitate the partitioning of the *GoS*, *ITU* Recommendations provide the so-called reference connections. A reference connection consists of one or more simplified drawings of the path a call (or connection) can take in the network, including appropriate reference points where the interfaces between entities are defined. In some cases a reference point define an interface between two operators. Recommendations devoted to provide reference connections are listed in Tab. 1.4. **Recommendation E.701** provides reference connection for *N-ISDN* networks, **Recommendation E.751** for land mobile networks, **Recommendation E.752** for maritime and aeronautical systems, **Recommendation E.755** for *UPT* services, and **Recommendation E.651** for *IP*–based networks. In the latter, general reference connections are provided for the end-to-end connections and more detailed ones for the access network in case of *HFC* (Hybrid Fiber Coax) systems. As an example, Fig. 1.10 (taken from Fig. 6.2 of Recommendation E.651) presents the reference connection for an *IP–to–PSTN/ISDN* or *PSTN/ISDN–to–IP* call.

Rec.	Date	Title
E.701	10/92	Reference connections for traffic engineering
E.751	02/96	Reference connections for traffic engineering of land mobile networks
E.752	10/96	Reference connections for traffic engineering of maritime and aeronautical systems
E.755	02/96	Reference connections for *UPT* traffic performance and GoS
E.651	03/00	Reference connections for traffic engineering of *IP* access networks

Table 1.4: *Recommendations on reference connections.*

We now apply the philosophy explained above for defining *GoS* objectives, starting with the elaboration of Recommendation E.720, devoted to *N-ISDN*. The recommendations on *GoS* objectives for *PSTN*, which are generally older, follow a different philosophy and can now be considered an exception within the set of GoS recommendations. Let us start this overview with the new recommendations. They are listed in Tab. 1.5. **Recommendations E.720 and E.721** are devoted to *N-ISDN* circuit-switched services. Recommendation E.720 provides general guidelines and Recommendation E.721 provides *GoS* parameters and target values. The recommended end-to-end GoS parameters are:

- Pre-selection delay
- Post-selection delay

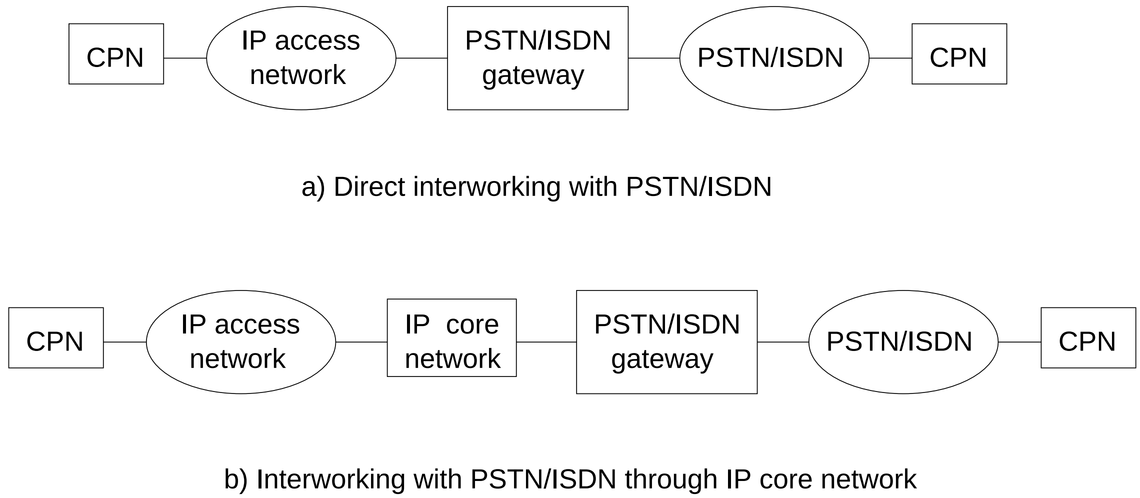


Figure 1.10: *IP–to–PSTN/ISDN or PSTN/ISDN–to–IP reference connection. CPN = Customer Premises Network.*

- Answer signal delay
- Call release delay
- Probability of end-to-end blocking

After defining these parameters, Recommendation E.721 provides target values for normal and high load as defined in Recommendation E.500. For the delay parameters, target values are given for the mean delay and for the 95 % quantile. For those parameters that are dependent on the length of the connection, different sets of target values are recommended for local, toll, and international connections. The recommendation provides reference connections, characterised by a typical range of the number of switching nodes, for the three types of connections.

Based on the delay related *GoS* parameters and target values given in Recommendations E.721, **Recommendation E.723** identifies *GoS* parameters and target values for Signalling System # 7 networks. The identified parameters are the delays incurred by the initial address message (*IAM*) and by the answer message (*ANM*). Target values consistent with those of Recommendation E.721 are given for local, toll and international connections. The typical number of switching nodes of the reference connections provided in Recommendation E.721 are complemented in Recommendation E.723 with typical number of *STP*s (signal transfer points).

The target values provided in Recommendation E.721 refer to calls not invoking intelligent network (*IN*) services. **Recommendation E.724** specifies incremental delays that are allowed when they are invoked. Reference topologies are provided for the most relevant service classes, such as database query, call redirection, multiple set-up attempts, etc. Target values of the incremental delay for processing a single *IN* service are provided for some service

Rec.	Date	Title
E.720	11/98	*ISDN* grade of service concept
E.721	05/99	Network grade of service parameters and target values for circuit-switched services in the evolving *ISDN*
E.723	06/92	Grade-of-service parameters for Signalling System No. 7 networks
E.724	02/96	GoS parameters and target GoS objectives for IN Services
E.726	03/00	Network grade of service parameters and target values for *B-ISDN*
E.728	03/98	Grade of service parameters for *B-ISDN* signalling
E.770	03/93	Land mobile and fixed network interconnection traffic grade of service concept
E.771	10/96	Network grade of service parameters and target values for circuit-switched land mobile services
E.773	10/96	Maritime and aeronautical mobile grade of service concept
E.774	10/96	Network grade of service parameters and target values for maritime and aeronautical mobile services
E.775	02/96	*UPT* Grade of service concept
E.776	10/96	Network grade of service parameters for *UPT*
E.671	03/00	Post selection delay in *PSTN/ISDNs* using Internet telephony for a portion of the connection

Table 1.5: *Recommendations on GoS objectives (except for PSTN).*

classes as well as of the total incremental post-selection delay for processing all *IN services*.

Recommendation E.726 is the equivalent of Recommendation E.721 for *B-ISDN*. As *B-ISDN* is a packet-switched network, call-level and packet-level (in this case cell-level) *GoS* parameters are distinguished. Call-level *GoS* parameters are analogous to those defined in Recommendation E.721. The end-to-end cell-level *GoS* parameters are:

- Cell transfer delay
- Cell delay variation
- Severely errored cell block ratio
- Cell loss ratio
- Frame transmission delay
- Frame discard ratio

While the call-level *QoS* requirements may be similar for all the services (perhaps with the exception of emergency services), the cell-level *QoS* requirements may be very different depending on the type of service: delay requirements for voice and video services are much more stringent than those for data services. Thus target values for the cell-level must be service dependent. These target values are left for further study in the current issue while target values are provided for the call-level *GoS* parameters for local, toll and international connections.

Recommendation E.728, for *B-ISDN* signalling, is based on the delay related call-level parameters of Recommendation E.726. Recommendation E.728 in its relation to Recommendation E.726, is analogous to the corresponding relationship between Recommendation E.723 and E.721.

In the mobile network series, there are three pairs of recommendations analogous to the E.720/E.721 pair: **Recommendations E.770 and E.771** for land mobile networks, **Recommendations E.773 and E.774** for maritime and aeronautical systems and **Recommendations E.775 and E.776** for *UPT* services. All these are for circuit-switched services. They analyse the features of the corresponding services that make it necessary to specify less stringent target values for the *GoS* parameters than those defined in E.721, and define additional *GoS* parameters that are specific for these services. For example, in Recommendations E.770 and E.771 on land mobile networks, the reasons for less stringent parameters are: the limitations of the radio interface, the need for the authentication of terminals and of paging of the called user, and the need for interrogating the home and (in case of roaming) visited network databases to obtain the routing number. An additional *GoS* parameter in land mobile networks is the probability of unsuccessful hand-over. Target values are given for fixed-to-mobile, mobile-to-fixed and mobile-to-mobile calls considering local, toll and international connections.

The elaboration of recommendations on *GoS* parameters and target values for *IP*–based network has just started. **Recommendation E.671** only covers an aspect on which was urgent to give advice. It was to specify target values for the post-selection delay in *PSTN/ISDN* networks when a portion of the circuit-switched connection is replaced by *IP* telephony and the users are not aware of this fact. Recommendation E.671 states that the end-to-end delay must in this case be equal to that specified in Recommendation E.721.

Let us finish this overview on GoS recommendations with those devoted to the *PSTN*. They are listed in Tab. 1.6. Recommendations E.540, E.541 and E.543 can be considered the counterpart for *PSTN* of Recommendation E.721 but organised in a different manner, as pointed out previously. They are focused on international connections, as was usual in the old *ITU* recommendations. **Recommendation E.540** specifies the blocking probability of the international part of an international connection, **Recommendation E.541** the end-to-end blocking probability of an international connection, and **Recommendation E.543** the internal loss probability and delays of an international telephone exchange. A revision of these recommendations is needed to decide if they can be deleted, while extending the scope

Rec.	Date	Title
E.540	11/98	Overall grade of service of the international part of an international connection
E.541	11/88	Overall grade of service for international connections (subscriber-to-subscriber)
E.543	11/88	Grades of service in digital international telephone exchanges
E.550	03/93	Grade of service and new performance criteria under failure conditions in international telephone exchanges

Table 1.6: *Recommendations on GoS objectives in the PSTN.*

of Recommendation E.721 to cover *PSTN*.

The target values specified in all of the *GoS* recommendations assume that the network and its components are fully operational. On the other hand, the Recommendations on availability deal with the intensity of failures and duration of faults of network components, without considering the fraction of call attempts which is blocked due to the failure. **Recommendation E.550** combines the concepts from the fields of both availability and traffic congestion, and defines new performance parameters and target values that take into account their joint effects in a telephone exchange.

1.5.4 Traffic controls and dimensioning

Once the traffic demand has been characterised and the *GoS* objectives have been established, traffic engineering provides a cost efficient design and operation of the network while assuring that the traffic demand is carried and *GoS* objectives are satisfied.

The inputs of traffic engineering to the design and operation of networks are network dimensioning and traffic controls. Network dimensioning assures that the network has enough resources to support the traffic demand. It includes the dimensioning of the physical network elements and also of the logical network elements, such as the virtual paths of an *ATM* network. Traffic controls are also necessary to ensure that the *GoS* objectives are satisfied. Among the traffic controls we can distinguish:

- **Traffic routing:** routing patterns describe the route set choices and route selection rules for each origin-destination pair. They may be hierarchical or non-hierarchical, fixed or dynamic. Dynamic methods include time-dependent routing methods,in which the routing pattern is altered at a fixed time on a pre-planned basis, and state-dependent or event-dependent routing, in which the network automatically alters the routing pattern based on present network conditions. Recommendations E.170 to E.177 and E.350 to E.353 all deal with routing, are out of the scope of this section. Nevertheless, reference to routing is constantly made in the traffic engineering recommendations here presented. On one hand routing design is based on traffic engineering considerations: for example, alternative routing schemes are based on cost efficiency considerations, dynamic routing methods are based on considerations of robustness under focused overload or failure conditions or regarding traffic forecast errors. On the other hand, network dimensioning is done by taking into account routing methods and routing patterns.

- **Network traffic management controls:** these controls assure that network throughput is maintained under any overload or failure conditions. Traffic management controls may be *protective or expansive*. The protective controls such as code blocking or call gapping assure that the network does not waste resources in processing calls that will be unsuccessful or limit the flow of calls requiring many network resources (overflow calls). The expansive controls re-route the traffic towards those parts of the network that are not overloaded. Traffic management is usually carried out at traffic management centres where real-time monitoring of network performance is made through the collection and display of real-time traffic and performance data. Controls are usually triggered by an operator on a pre-planned basis (when a special event is foreseen) or in real-time. In the *ITU–T* organisation, network traffic management is under the responsibility of WP 2/2. Recommendations E.410 to E.417, dealing with this subject, are out of the scope of this section. Nevertheless, reference to traffic management is made in the traffic engineering recommendations. For example, measurement requirements specified in the traffic and performance measurement recommendations include the real-time measurements required for network traffic management.

- **Service protection methods:** they are call-level traffic controls that control the grade of service for certain streams of traffic by means of a discriminatory restriction of the access to circuit groups with little idle capacity. Service protection is used to provide stability in networks with non-hierarchical routing schemes by restricting overflow traffic to an alternative route that is shared with first-choice traffic. It is also used to balance *GoS* between traffic streams requesting different bandwidth or to give priority service to one type of traffic.

- **Packet-level traffic controls:** these controls assure that the packet-level *GoS* objectives of the accepted calls are satisfied under any network condition and that a cost-efficient grade of service differentiation is made between services with different packet-level *QoS* requirements.

- **Signalling and intelligent network (IN) controls:** given that these networks are the neural system of the whole network, a key objective in the design and operation of them is to maximise their robustness, that is, their ability to withstand both traffic overloads and failures of network elements. It is achieved both by means of redundancy of network elements and by means of a set of congestion and overload controls, as explained in Recommendations E.744 to be described below.

Let us classify the recommendations on dimensioning and traffic controls into those devoted to circuit-switched networks, to packet-switched networks, and to signalling and *IN*–structured networks.

Circuit-Switched networks

Recommendations on traffic controls and dimensioning of circuit-switched networks are listed in Tab. 1.7. These recommendations deal with dimensioning and service protection methods taking into account traffic routing methods.

Rec.	Date	Title
E.510	10/45	Determination of the number of circuits in manual operation
E.520	11/88	Number of circuits to be provided in automatic and/or semi-automatic operation, without overflow facilities
E.521	11/88	Calculation of the number of circuits in a group carrying overflow traffic
E.522	11/88	Number of circuits in a high-usage group
E.524	05/99	Overflow approximations for non-random inputs
E.525	06/92	Designing networks to control grade of service
E.526	03/93	Dimensioning a circuit group with multi-slot bearer services and no overflow inputs
E.527	03/00	Dimensioning at a circuit group with multi-slot bearer services and overflow traffic
E.528	02/96	Dimensioning of digital circuit multiplication equipment (DCME) systems
E.529	05/97	Network dimensioning using end-to-end GoS objectives
E.731	10/92	Methods for dimensioning resources operating in circuit switched mode

Table 1.7: *Recommendations on traffic controls and dimensioning of circuit–switched networks.*

Recommendations E.520, E.521, E.522 and E.524 deal with the dimensioning of circuit groups or high-usage/final group arrangements carrying single-rate (or single-slot) connections. Service protection methods are not considered in these recommendations:

- **Recommendation E.520** deals with methods for dimensioning of only-path circuit groups (Fig. 1.11a).
- **Recommendations E.521 and E.522** provide methods for the dimensioning of simple alternative routing arrangements as the one shown in Fig. 1.11(b), where there only

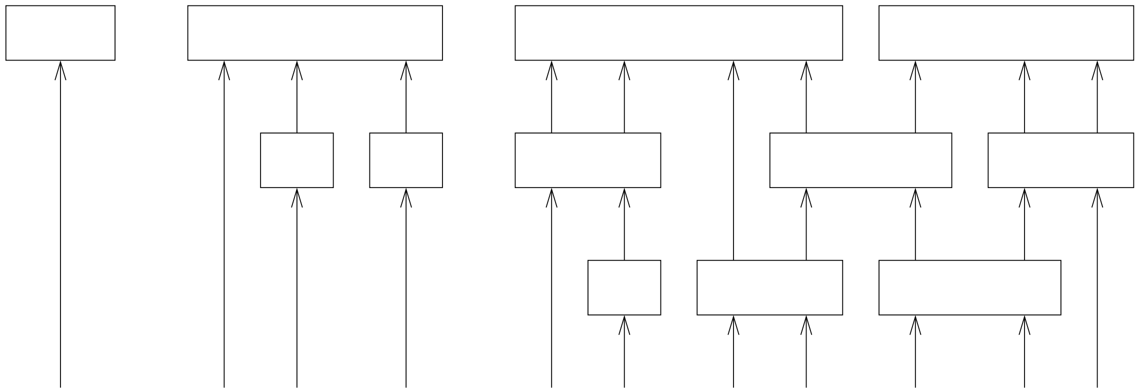


Figure 1.11: *Examples of circuit group arrangements.*

exist first- and second-choice routes, and where the whole traffic overflowing from a circuit group is offered to the same circuit group. Recommendation E.521 provides methods for dimensioning the final group satisfying GoS requirements for given sizes of the high-usage circuit groups, and Recommendation E.522 advises on how to dimension high-usage groups to minimise the cost of the whole arrangement.

- **Recommendation E.524** provides overflows approximations for non-random inputs which allows for the dimensioning of more complex arrangements (i.e. without the previous mentioned limitations) as that shown in Fig. 1.11 (c). Several approaches are described and compared from the point of view of accuracy and complexity.

Recommendation E.525 introduces service protection methods for networks carrying single-rate connections. It describes the applications and the available methods: split circuit groups, circuit reservation (also called trunk reservation or, in packet-switched networks, bandwidth reservation) and virtual circuits. The recommendations provides methods to evaluate the blocking probability of each traffic stream both for only-path circuit groups and for alternative routing arrangements, which allow for the dimensioning of the circuit groups and of the thresholds defining the protection methods. A comparison of the available service protection methods is made from the point of view of efficiency, overload protection, robustness and impact of peakedness.

Recommendations E.526 and E.527 deal with the dimensioning of circuit groups carrying multi-slot (or multi-rate) connections. Service protection methods are considered in both of them. Recommendation E.526 deals with only-path circuit groups while Recommendation E.527 deals with alternative routing schemes.

Tab. 1.8 summarises the items considered in each of the Recommendations mentioned above. **Recommendation E.528** deals with the dimensioning of a particular but very important type of circuit group, where Digital Circuit Multiplication Equipment (*DCME*) is used to achieve statistical multiplexing gain in communications via satellite. This is to save circuits by means of interpolating speech bursts of different channels by taking advantage of the silences

Recommendation	E.520	E.521	E.522	E.524	E.525	E.526	E.527
Alternative routing	No	Yes*	Yes*	Yes	Yes	No	Yes
Service protection	No	No	No	No	Yes	Yes	Yes
Multi-slot connections	No	No	No	No	No	Yes	Yes

Table 1.8: *Items considered in the circuit group dimensioning. Recommendations E.520 to E.527. * Only simple arrangements.*

existing in a conversation. Dimensioning methods for circuit groups providing integration of traffic containing voice, facsimile and voice band data are given.

Recommendation E.731 is also devoted to circuit group dimensioning and considers those special features of N-IDSN that may have an impact on traffic engineering. Apart from multi-slot connections and service protection methods, the recommendation studies the impact of attribute negotiation (of attributes affecting either the choice of circuit group or the required number of circuits), of service reservation (reservation of dedicated resources or of resources shared with on demand services) and of point-to-multi-point connections.

Recommendation E.529 collects all the dimensioning methods on circuit group or alternative routing arrangement described in previous Recommendations, with a view to giving guidelines for the dimensioning of the whole network using end-to-end *GoS* objectives. Dimensioning methods for networks with fixed, time-dependent, state-dependent or event-dependent traffic routing are described. Principles for the decomposition of the networks into blocks that may be considered statistically independent are given, and the iterative procedure required for network optimisation is described.

Packet-Switched networks

Recommendations on traffic controls and dimensioning of packet-switched networks are listed in Tab. 1.9. They deal with *B-ISDN* networks using *ATM* technology, but most of the methods described apply to other packet-switched networks, as for example *IP*–based networks, in which the admission of connections is controlled.

The connection admission control (*CAC*) establishes a division between the packet-level and the connection-level. When a user request the establishment of a new connection, the *CAC* decides if the connection can be admitted while satisfying packet-level *GoS* of both new and existing connections. This decision is usually made by means of allocating resources (typically bandwidth) to each connection and refusing new request when there are insufficient resources. Thus:

Rec.	Date	Title
E.735	05/97	Framework for traffic control and dimensioning in *B-ISDN*
E.736	05/97	Methods for cell level traffic control in *B-ISDN*
E.737	05/97	Dimensioning methods for *B-ISDN*

Table 1.9: *Recommendations on traffic controls and dimensioning of packet-switched networks.*

- From a packet-level perspective: as the *CAC* assures that packet-level *GoS* objectives are satisfied regardless of the rate of connections offered to the network, it makes the packet-level independent from the connection-level offered traffic and from the network dimensioning.

- From a connection-level perspective: as the *CAC*, in deciding on the acceptance of a connection, takes into account all the packet-level controls implemented, it summarises all the packet-level controls in an amount of resources required by a connection. It makes the connection-level of a packet-switched networks similar to that of a circuit-switched network: the amount of resources required by a connection, called effective or *equivalent bandwidth* (or, in *ATM*, *equivalent cell rate*) is equivalent to the number of slots required by a multi-slot connection in a circuit-switched network. Connection-level traffic controls and network dimensioning must assure that the connection-level *GoS* requirements, typically the specified connection blocking probabilities, are satisfied taking into account the effective bandwidth that has to be allocated to each connection.

In practice, this separation between packet-and connection-level is not so complete as described above: the effective bandwidth of a connection depends on the capacity of the physical or logical link in which it is carried (apart from the packet-level traffic characteristics of the connection) while, in its turn, the capacity of the links must be dimensioned by taking into account the effective bandwidth of the connections. Thus, an iterative process between connection- and packet-level for network dimensioning is necessary.

Recommendation E.735 is the framework for traffic control and dimensioning in *B-ISDN*. It introduces the concepts described above, defines what is a connection and what is a resource, and analyses strategies for logical network configuration.

Recommendation E.736 focuses on packet-level. It provides methods for packet-level performance evaluation, proposes possible multiplexing strategies (peak rate allocation, rate envelope multiplexing and statistical rate sharing) and analyses the implications and applications of each of them. Based on this analysis, the recommendation provides methods for packet-level controls. Emphasis is placed on methods for Connection Admission Control and for the integration (or segregation) of services with different *QoS* requirements either by using dedicated resources or by sharing the same resources and implementing loss and/or delay priorities. It also addresses adaptive resource management techniques to control the flow of packets of services with non-stringent delay requirements.

Recommendation E.737 provides methods for circuit group and network dimensioning and addresses connection-level traffic controls, in particular service protection methods. Traffic routing methods are also taken into account. As the effective bandwidth of a connection is modelled as a number of slots of a multi-slot connection, this recommendation is not very different from those on circuit-switched network dimensioning. Nevertheless the recommendation deals with some features that are particular of packet-switched networks: the above mentioned iteration between effective bandwidth and network dimensioning; the required bandwidth discretization into multiples of a bandwidth quantisation unit, given that the multi-slot models only deal with integer number of slots; and the implications on the dimensioning of services with different packet-level *QoS* requirements.

Signalling and IN–Structured Networks

The recommendations on traffic controls and dimensioning of signalling networks and intelligent networks (*IN*) are listed in Tab. 1.10. Recommendations E.733 and E.734 deal with dimensioning and Recommendation E.744 with traffic controls.

Rec.	Date	Title
E.733	11/98	Methods for dimensioning resources in Signalling System No. 7 networks
E.734	10/96	Methods for allocating and dimensioning Intelligent Network (IN) resources
E.744	10/96	Traffic and congestion control requirements for SS No. 7 and IN–structured networks

Table 1.10: *Recommendations on traffic controls and dimensioning of signalling and IN–structured networks.*

Recommendation E.733 provides a methodology for the planning and dimensioning of signalling system No. 7 networks. The methodology takes into account the fact that the efficiency of the signalling links should not be the primary consideration, but the performance of the network under failure and traffic overload has greater importance. The recommendation describes the reference traffic and reference period that, in agreement with Recommendations E.492 and E.500, must be used to dimension the number of signalling links and to ensure that the capacity of network switching elements is not exceeded. It describes the factors for determining a maximum design link utilisation, ϱ_{max}, which ensure that the end-to-end delay objectives described in Recommendation E.723 are met. Delays incurred when, due to failures, the link load is $2\,\varrho_{max}$ are also taken into account for determining ϱ_{max}. Initial values for ϱ_{max} being used are described and methods are given for determining the number of signalling links and the switching capacity required.

Recommendation E.734 deals with resource allocation and dimensioning methods for Intelligent Networks. It discusses the new traffic engineering factors to be considered: services with reference period out of the normal working hours, mass calling situations produced by some services, fast implementation of new services with uncertain forecast. The last factor makes it necessary to have the allocation and dimensioning procedures flexible enough to provide, as quickly as possible, the resources required as new services are implemented or the user demand changes. The recommendation provides criteria for resource allocation, both for the location of the *IN*–specific elements and for the partitioning of the Intelligent Network functionality (such as service logic) among these elements. It also provides methods for the dimensioning of the *IN* nodes and of the supporting signalling subnetwork, and discusses the impact on the circuit-switched network dimensioning.

Traffic and congestion control procedures for SS. No. 7 and *IN*–structured networks are specified in the Q and E.410-series Recommendations. These procedures generally leave key parameter values to be specified as part of the implementation. Given that robustness is a key requirement of signalling and *IN*–structured networks, a proper implementation of these controls is essential.

Recommendation E.744 provides guidelines for this implementation, indicating how the control parameters should be chosen in different types of networks. The recommendation also advises on requirements to be placed on signalling nodes and *IN* nodes on the needs for node-level overload controls and on how such controls must interrelate with network-level controls. Finally, the recommendation states basic principles to keep different systems and controls harmonised in order to allow for various vendor products and network implementations to be interconnected with a high confidence the control procedures will work properly.

1.5.5 Performance monitoring

Once the network is operational, continuous monitoring of the *GoS* is required. Although the network is correctly dimensioned, there are overload and failure situations not considered in the dimensioning where short term (minutes, hours) network traffic management actions have to be taken. In situations considered in the dimensioning, traffic forecast errors or approximations made in the dimensioning models may lead to a *GoS* different from the one expected. *GoS* monitoring is needed to detect these problems and to produce feedback for traffic characterisation and network design. Depending on the problems detected, network reconfigurations, changes of the routing patterns or adjustment of traffic control parameters can be made in medium term (weeks, months). The urgency of a long term planning of network extensions may also be assessed.

Recommendations E.490, E.491, E.502, E.503, E.504, E.505 and E.745, covering both traffic and performance measurements, have been described in Sec. 1.5.2, cover both traffic and performance measurements. We consider in this section two other Recommendations, E.492 and E.493, listed in Tab. 1.11 which are only related to performance measurements.

Rec.	Date	Title
E.492	02/96	Traffic reference period
E.493	02/96	Grade of Service (GoS) monitoring

Table 1.11: *Recommendations on performance measurements (for recommendations covering both traffic and performance measurements, see Tab. 1.2).*

Recommendation E.492 provides the definition of traffic reference periods for the purposes of collecting measurements for monitoring Grade-of-Service for networks and network components. This Recommendation is closely related to Recommendation E.500, which defines read-out periods for traffic intensity measurements required for network dimensioning. These read-out periods have to be consistent with those used for performance monitoring once the network is operative. Recommendation E.492 also defines the *normal* and *high load* periods that are representative of each month. The purpose of these definitions, also consistent with those of Recommendation E.500, is to identify which day and read-out period to use for comparing the monitored *GoS* to the *GoS* target values specified for *normal* and *high load*.

Recommendation E.493 addresses how to perform end-to-end *GoS* monitoring, taking into account practical limitations. Measurement of blocking or mishandling probabilities is straightforward. However, as direct measurements of end-to-end delays are not feasible in a continuous monitoring, the Recommendation proposes methods to approximate end-to-end delays (mean and 95 % quantile) by means of local measurements autonomously taken in each network element. The proposed methods do not require coordination between network elements to take the measurements. The Recommendation also explains how to apply the proposed methods to the monitoring of each of the connection-level *GoS* parameters defined in the recommendations on *GoS* objectives.

1.5.6 Other recommendations

There are a few other Recommendations for which their scope does not match any of the items considered in the classification made here. They are listed in Tab. 1.12.

Recommendations E.600 provides a list of traffic engineering terms and definitions used throughout the whole set of traffic engineering Recommendations.

Recommendations E.700 & E.750 are introductory Recommendations to the E.700/749 Series Recommendations on traffic engineering for *N*- and *B-ISDN*, and to the E.750/799

Rec.	Date	Title
E.523	11/88	Standard traffic profiles for international traffic streams
E.600	03/93	Terms and definitions of traffic engineering
E.700	10/92	Framework of the E.700-Series Recommendations
E.750	03/00	Introduction to the E.750-Series of Recommendations on traffic engineering aspects of networks supporting mobile and *UPT* services

Table 1.12: *Recommendations not matching under any of the items considered in the classification made here.*

Series Recommendations on traffic engineering for mobile networks, respectively.

Recommendation E.523 provides standardised 24-hour traffic profiles for traffic streams between countries in different relative time locations. This measurement-based information may be useful for those countries where no measurements are available. The profiles refer to telephone traffic and must not be used for data traffic for which the profiles may be very different.

1.5.7 Work program for the Study Period 2001–2004

The work in *ITU* is planned for periods of four years, called *study periods*. In the past, recommendations developed along a study period were approved and published at the end of the period. At present, working methods are more dynamic: recommendations can be approved and published at any moment, work program prepared for a study period can be updated along the period according to needs.

Work program for the 2001-2004 period makes emphasis on traffic engineering for Personal Communications, *IP* Networks and Signalling. Three Questions (i.e. subjects for study) have been defined, one for each topic. The titles of the Questions are:

- Traffic engineering for Personal Communications;
- Traffic engineering for SS7- and *IP*–based Signalling Networks;
- Traffic engineering for Networks Supporting *IP* Services.

An Expert Group has been formed for each Question. The Expert Group, co-ordinated by a rapporteur, is in charge of elaborating the recommendations related to the Question.

1.5.8 Conclusions

An overview of the *ITU* traffic engineering recommendations has been given. A high amount of work of worldwide specialists on traffic engineering is behind this extensive set of recommendations. The whole set intends to be a valuable help for engineers in charge of designing and operating telecommunication networks. Nevertheless, the set of traffic engineering recommendation can never be a complete set: new technologies, new services, new teletraffic methods are continuously appearing and need to be incorporated to the recommendations.

Teletraffic researchers are encouraged to contribute to the preparation of new recommendations and to the revision of the old ones. The *ITU* recommendations can be seen as a bridge between the teletraffic research activity and the daily traffic engineering practice carried out by the operators. An innovative method has a greater chance to be used in practice if it appears in an *ITU* recommendation. It is thus worth for the researcher to contribute to the *ITU* in order to extend his ideas. The daily operational practice will also obtain benefit from this contribution. Current working methods as for instance the extensive use of E-mail, facilitate informal cooperation of any researcher with the *ITU* work.

Chapter 2

Traffic concepts and grade of service

The costs of a telephone system can be divided into costs which are dependent upon the number of subscribers and costs that are dependent upon the amount of traffic in the system.

The goal when planning a telecommunication system is to adjust the amount of equipment so that variations in the subscriber demand for calls can be satisfied without noticeable inconvenience while the costs of the installations are as small as possible. The equipment must be used as efficiently as possible.

Teletraffic engineering deals with optimisation of the structure of the network and adjustment of the amount of equipment that depends upon the amount of traffic.

In the following some fundamental concepts are introduced and some examples are given to show how the traffic behaves in real systems. All examples are from the telecommunication area.

2.1 Concept of traffic and traffic unit [erlang]

In teletraffic theory we usually use the word *traffic* to denote the traffic intensity, i.e. traffic per time unit. The term traffic comes from Italian and means business. According to ITU–T (1993 [34]) we have the following definition:

Definition of Traffic Intensity: *The instantaneous traffic intensity in a pool of resources is the number of busy resources at a given instant of time.*

The pool of resources may be a group of servers, e.g. trunk lines. The statistical moments of the traffic intensity may be calculated for a given period of time T. For the mean traffic intensity we get:

$$Y(T) = \frac{1}{T} \cdot \int_0^T n(t)\, \mathrm{d}t. \tag{2.1}$$

where $n(t)$ denotes the number of occupied devices at the time t.

Carried traffic $Y = A_c$: This is called the traffic carried by the group of servers during the time interval T (Fig. 2.1). In applications, the term traffic intensity usually has the meaning of average traffic intensity.

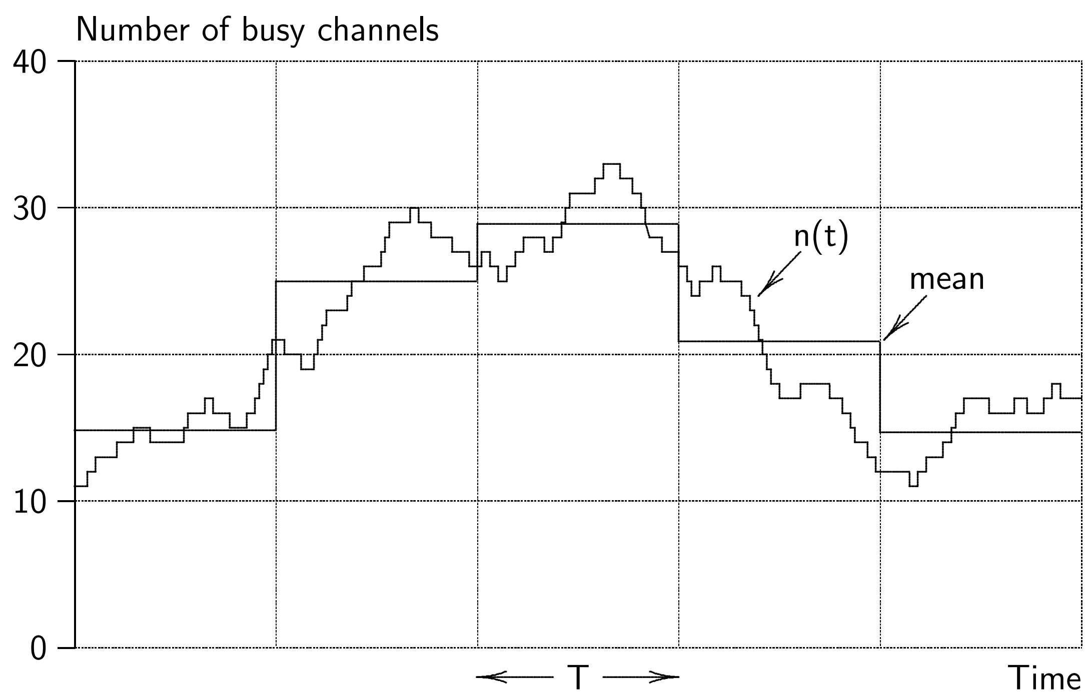


Figure 2.1: *The carried traffic (intensity) (= number of busy devices) as a function $n(t)$ of time. For dimensioning purposes we use the average traffic intensity during a period of time T (mean).*

The *ITU-T* recommendation also says that the unit usually used for traffic intensity is *erlang* (symbol *E*). This name was given to the traffic unit in 1946 by CCIF (predecessor to *CCITT* and to *ITU-T*), in honour of the Danish mathematician A. K. Erlang (1878-1929), who was the founder of traffic theory in telephony. The unit is dimensionless. The total traffic carried in a time period T is a *traffic volume*, and it is measured in erlang–hours (*Eh*). It is equal to the sum of all holding times inside the time period. According to the *ISO* standards the standardised unit should be *erlang-seconds*, but usually *erlang-hours* has a more natural order of size).

The carried traffic can never exceed the number of channels (lines). A channel can at most carry one erlang. The income is often proportional to the carried traffic.

Offered traffic A: In theoretical models the concept *offered traffic* is used; this is the traffic which would be carried if no calls were rejected due to lack of capacity, i.e. if the number of servers were unlimited. The offered traffic is a theoretical value and it cannot be measured. It is only possible to estimate the offered traffic from the carried traffic.

Theoretically we operate with two parameters:

1. call intensity λ, which is the mean number of calls offered per time unit, and

2. mean service time s.

The offered traffic is equal to:

$$A = \lambda \cdot s \,. \tag{2.2}$$

From this equation it is seen that the unit of traffic has no dimension. This definition assumes according to the above definition that there is an unlimited number of servers. If we use the definition for a system with limited capacity we get a definition which depends upon the capacity of the system. The latter definition has been used for many years, for example in the Engset case (Chap. 8), but it is not appropriate, because the offered traffic should be independent of the system.

Lost or Rejected traffic A_ℓ: The difference between offered traffic and carried traffic is equal to the rejected traffic. The value of this parameter can be reduced by increasing the capacity of the system.

Example 2.1.1: Definition of traffic
If the call intensity is 5 calls per minute, and the mean service time is 3 minutes then the offered traffic is equal to 15 erlang. The offered traffic-volume during a working day of 8 hours is then 120 erlang-hours. □

Example 2.1.2: Traffic units
Earlier other units of traffic have been used. The most common which may still be seen are:

SM	=	Speech-minutes 1 *SM*= 1/60 *Eh*.
CCS	=	Hundred call seconds: 1 *CCS* = 1/36 *Eh*. This unit is based on a mean holding time of 100 seconds and can still be found, e.g. in USA.
EBHC	=	Equated busy hour calls: 1 *EBHC* = 1/30 *Eh*. This unit is based on a mean holding time of 120 seconds.

We will soon realize, that *erlang* is the natural unit for traffic intensity because this unit is independent of the time unit chosen. □

The offered traffic is a theoretical parameter used in the theoretical dimensioning formulæ. However, the only measurable parameter in reality is the carried traffic, which often depends upon the actual system.

In data transmissions systems we do not talk about service times but about transmission needs. A job can for example be a data packet of s units (e.g. bits or bytes). The capacity of the system φ, the data signalling speed, is measured in units per second (e.g. bits/second). Then the service time for such a job, i.e. transmission time, is s/φ time units (e.g. seconds), i.e. depending on φ. If on the average λ jobs are served per time unit, then *the utilisation* ϱ of the system is:

$$\varrho = \frac{\lambda \cdot s}{\varphi} . \tag{2.3}$$

The observed utilisation will always be inside the interval $0 \leq \varrho \leq 1$, as it is the carried traffic.

Multi-rate traffic: If we have calls occupying more than one channel, and calls of type i occupy d_i channels, then the offered traffic expressed in number of busy channels becomes:

$$A = \sum_{i=0}^{N} \lambda_i \cdot s_i \cdot d_i \, , \tag{2.4}$$

where N is number of traffic types, and λ_i and s_i denotes the arrival rate and mean holding time of type i.

Potential traffic: In planning and demand models we use the term potential traffic, which would equal the offered traffic if there were no limitations in the use of the phone because of economics or availability (always a free phone available).

2.2 Traffic variations and the concept busy hour

The teletraffic varies according to the activity in the society. The traffic is generated by single sources, subscribers, who normally make telephone calls independently of each other.

A investigation of the traffic variations shows that it is partly of a stochastic nature partly of a deterministic nature. Fig. 2.2 shows the variation in the number of calls on a Monday morning. By comparing several days we can recognise a deterministic curve with superposed stochastic variations.

During a 24 hours period the traffic typically looks as shown in Fig. 2.3. The first peak is caused by business subscribers at the beginning of the working hours in the morning, possibly calls postponed from the day before. Around 12 o'clock it is lunch, and in the afternoon there is a certain activity again.

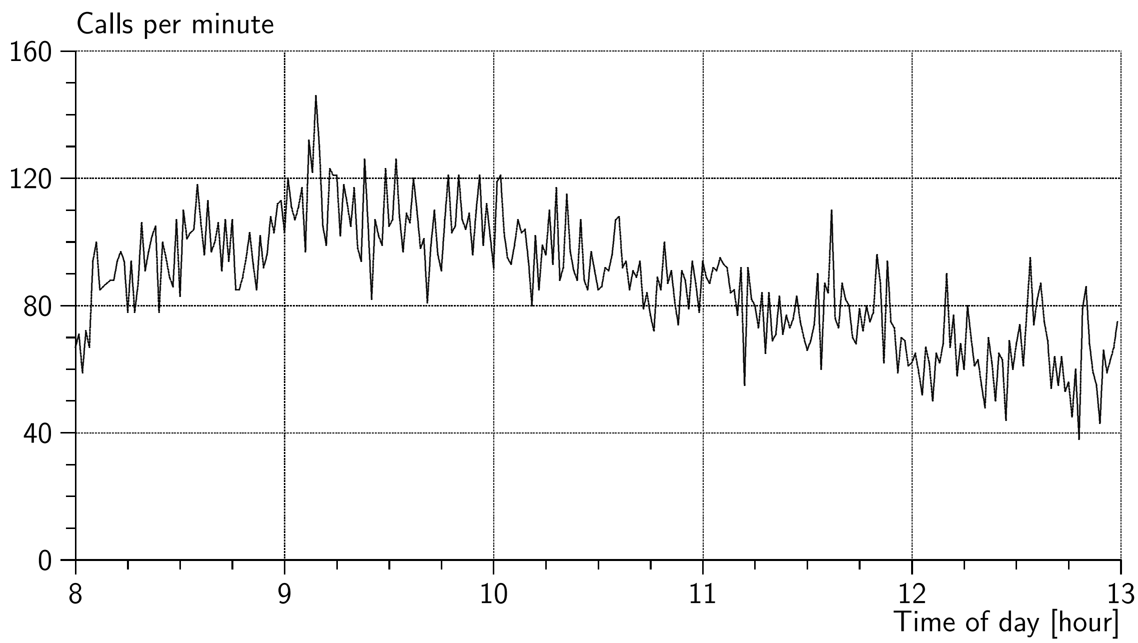


Figure 2.2: *Number of calls per minute to a switching centre a Monday morning. The regular 24-hour variations are superposed by stochastic variations.* (Iversen, 1973 [35]).

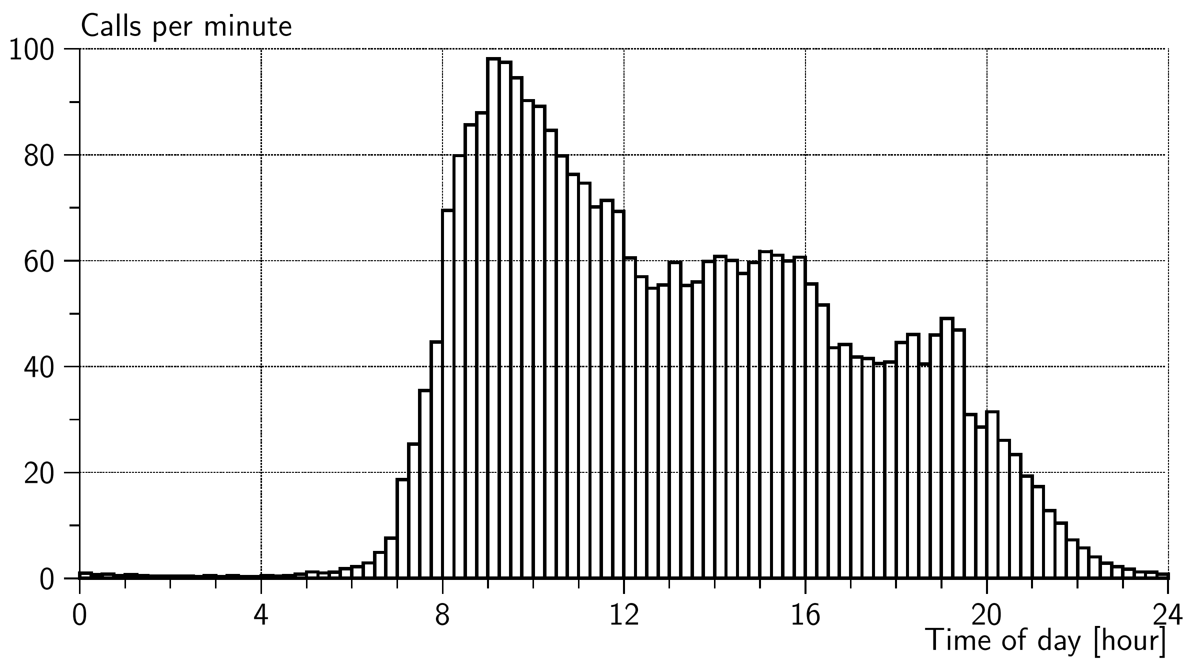


Figure 2.3: *The mean number of calls per minute to a switching centre taken as an average for periods of 15 minutes during 10 working days (Monday – Friday). At the time of the measurements there were no reduced rates outside working hours* (Iversen, 1973 [35]).

Around 19 o'clock there is a new peak caused by private calls and a possible reduction in rates after 19.30. The mutual size of the peaks depends among other thing upon whether the exchange is located in a typical residential area or in a business area. They also depend upon which type of traffic we look at. If we consider the traffic between Europa and for USA most calls takes place in the late afternoon because of the time difference.

The variations can further be split up into variation in call intensity and variation in service time. Fig. 2.4 shows variations in the mean service time for occupation times of trunk lines during 24 hours. During business hours it is constant, just below 3 minutes. In the evening it is more than 4 minutes and during the night very small, about one minute.

Busy Hour: The highest traffic does not occur at same time every day. We define the concept *time consistent busy hour, TCBH* as those 60 minutes (determined with an accuracy of 15 minutes) which during a long period on the average has the highest traffic.

It may therefore some days happen that the traffic during the *busiest hour* is larger than the time consistent busy hour, but on the average over several days, the busy hour traffic will be the largest.

We also distinguish between busy hour for the total telecommunication system, an exchange, and for a single group of servers, e.g. a trunk group. Certain trunk groups may have a busy hour outside the busy hour for the exchange (for example trunk groups for calls to the USA).

In practice, for measurements of traffic, dimensioning, and other aspects it is an advantage to have a predetermined well–defined busy hour.

The deterministic variations in teletraffic can be divided into:

- 24 hours variation (Fig. 2.3 and 2.4).

- Weekly variations (Fig. 2.5). Normally the highest traffic is on Monday, then Friday, Tuesday, Wednesday and Thursday. Saturday and especially Sunday has a very low traffic level. A good rule of thumb is that the 24 hour traffic is equal to 8 times the busy hour traffic (Fig. 2.5), i.e. only one third of capacity in the telephone system is utilised. This is the reason for the reduced rates outside the busy hours.

- Variation during a year. There is a high traffic in the beginning of a month, after a festival season, and after quarterly period begins. If Easter is around the 1st of April then we observe a very high traffic just after the holidays.

- The traffic increases year by year due to the development of technology and economics in the society.

Above we have considered traditional voice traffic. Other services and traffic types have other patterns of variation. In Fig. 2.6 we show the variation in the number of calls per 15 minutes

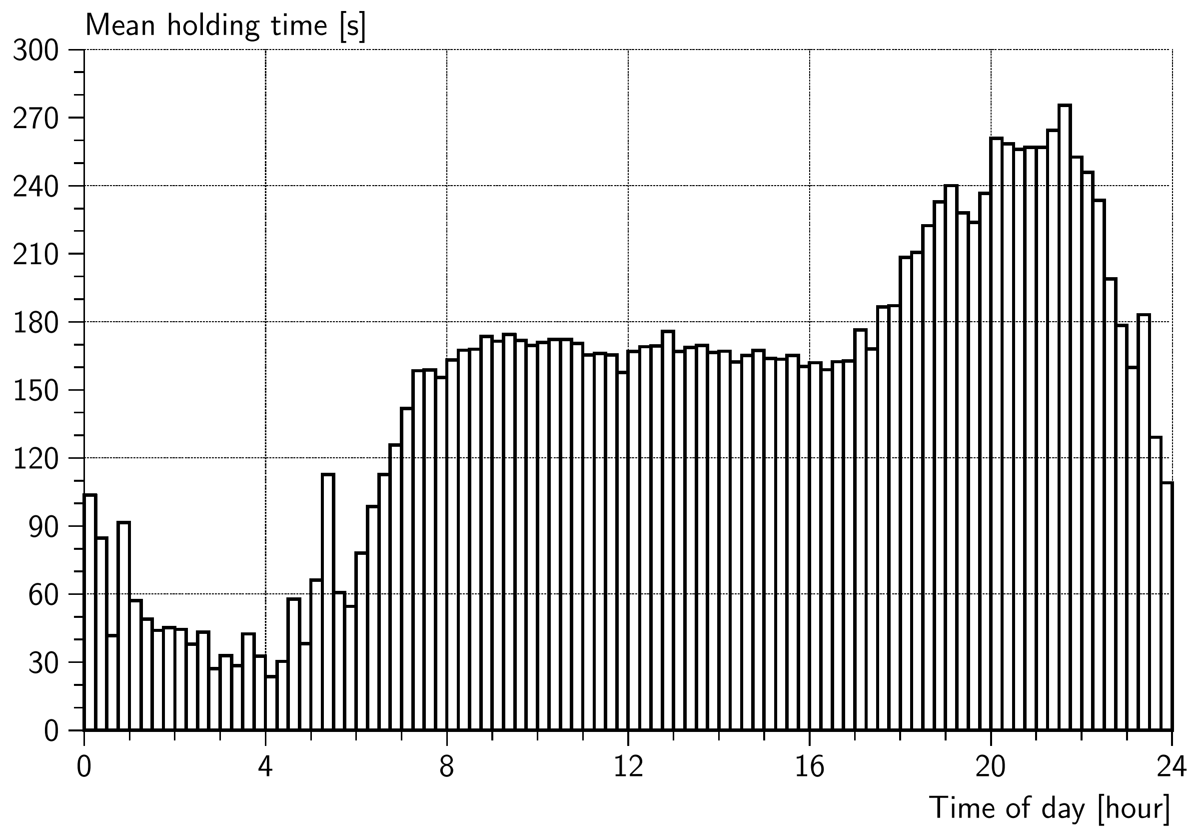


Figure 2.4: *Mean holding time for trunk lines as a function of time of day. (Iversen, 1973 [35]). The measurements exclude local calls.*

to a modem pool for dial-up Internet calls. The mean holding time as a function of the time of day is shown in Fig. 2.7.

Cellular mobile telephony has a different profile with maximum late in the afternoon, and the mean holding time is shorter than for wire-line calls. By integrating various forms of traffic in the same network we may therefore obtain a higher utilisation of the resources.

2.3 The blocking concept

The telephone system is not dimensioned so that all subscribers can be connected at the same time. Several subscribers are sharing the expensive equipment of the exchanges. The concentration takes place from the subscriber toward the exchange. The equipment which is separate for each subscriber should be made as cheap as possible.

In general we expect that about 5–8 % of the subscribers should be able to make calls at the same time in busy hour (each phone is used 10–16 % of the time). For international calls less than 1 % of the subscribers are making calls simultaneously. Thus we exploit *statistical*

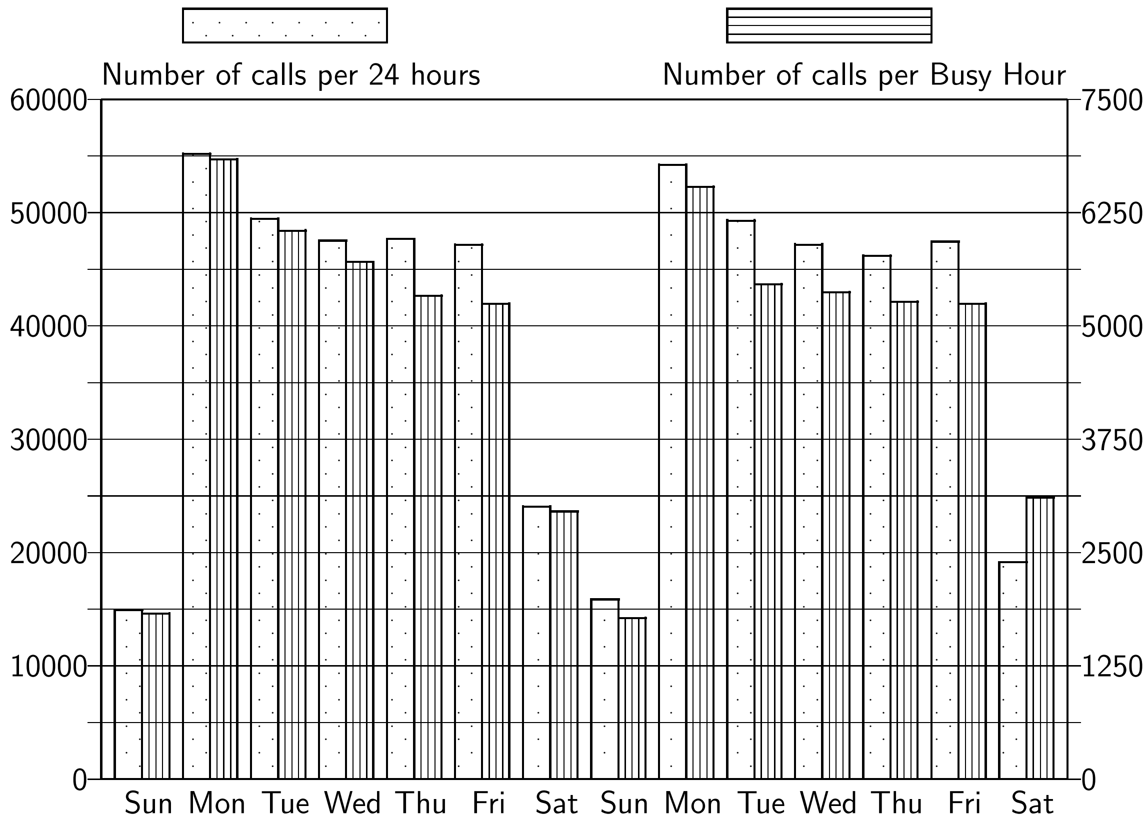


Figure 2.5: *Number of calls per 24 hours to a switching centre (left scale). The number of calls during busy hour is shown for comparison at the right scale. We notice that the 24–hour traffic is approximately 8 times the busy hour traffic. This factor is called the traffic concentration (Iversen, 1973 [35]).*

multiplexing advantages. Every subscriber should feel that he has unrestricted access to all resources of the telecommunication system even if he is sharing it with many others.

The amount of equipment is limited for economical reasons and it is therefore possible that a subscriber cannot establish a call, but has to *wait* or be *blocked* (the subscriber for example gets busy tone and has to make a new call attempt). Both are inconvenient to the subscriber. Depending on how the system operates we distinguish between *loss–systems* (e.g. trunk groups) and *waiting time systems* (e.g. common control units and computer systems) or a mixture of these if the number of waiting positions (buffer) is limited.

The inconvenience in *loss–systems* due to insufficient equipment can be expressed in three ways (network performance measures):

Call congestion B: The fraction of all call attempts which observes all servers busy (the user-perceived quality-of-service, the nuisance the subscriber experiences).

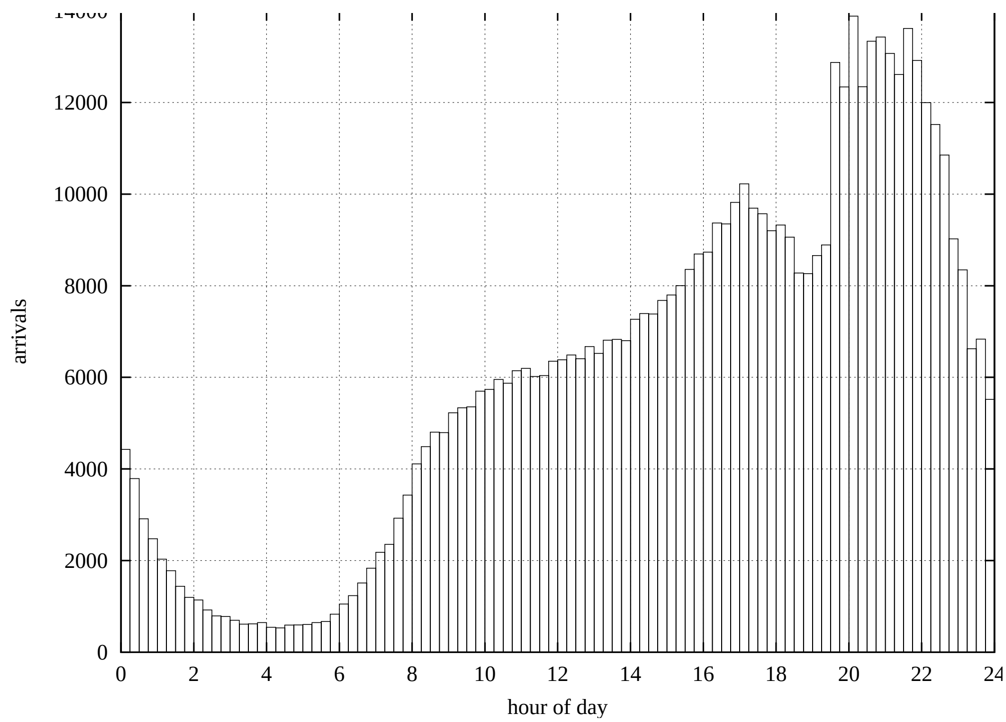


Figure 2.6: *Number of calls per 15 minutes to a modem pool of Tele Danmark Internet. Tuesday 1999.01.19.*

Time congestion E: The fraction of time when all servers are busy. Time congestion can for example be measured at the exchange (= *virtual congestion*).

Traffic congestion C: The fraction of the offered traffic that is not carried, possibly despite several attempts.

These quantitative measures can forexample be used to establish dimensioning standards for trunk groups.

At small congestion values it is possible with a good approximation to handle congestion in the different part of the system as mutually independent. The congestion for a certain route is then approximately equal to the sum of the congestion in each link of the route. During the busy hour we normally allow a congestion of a few percentage between two subscribers.

The systems cannot manage every situation without inconvenience for the subscribers. The purpose of teletraffic theory is to find relations between quality of service and cost of equipment. The existing equipment should be able to work at maximum capacity during abnormal traffic situations (e.g. a burst of phone calls), i.e. the equipment should keep working and make useful connections.

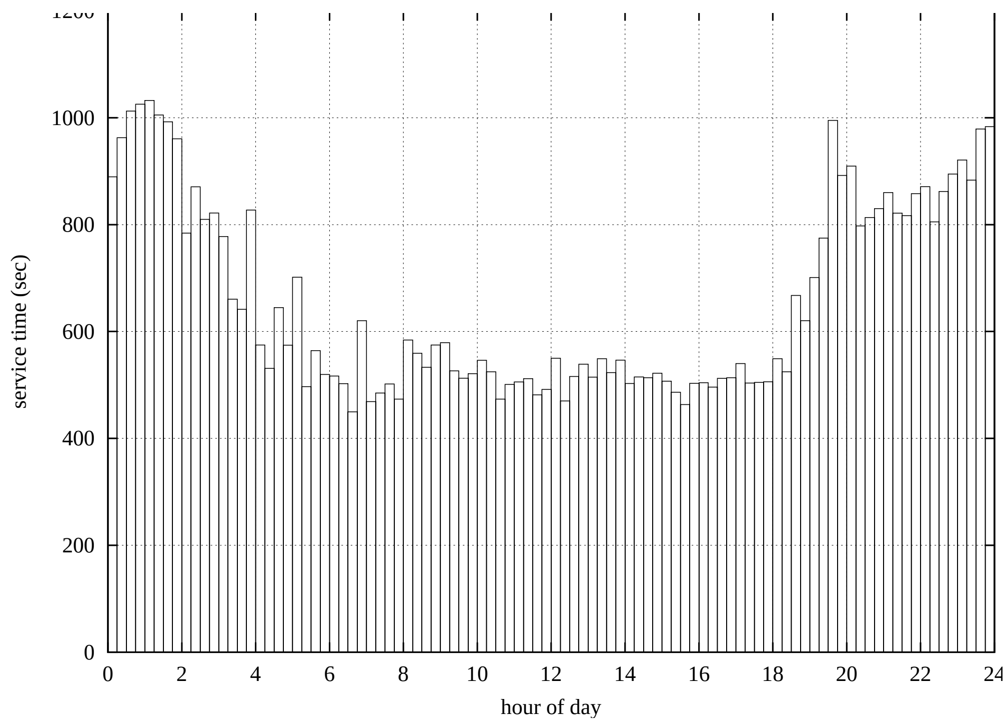


Figure 2.7: *Mean holding time in seconds as a function of time of day for calls arriving inside the period considered. Tele Danmark Internet. Tuesday 1999.01.19.*

The inconvenience in *delay–systems* (queueing systems) is measured as a waiting time. Not only the mean waiting time is of interest but also the distribution of the waiting time. It could be that a small delay do not mean any inconvenience, so there may not be a linear relation between inconvenience and waiting time.

In telephone systems we often define an upper limit for the acceptable waiting time. If this limit exceeded then a time-out of the connection will take place (enforced disconnection).

2.4 Traffic generation and subscribers reaction

If *Subscriber A* want to speak to *Subscriber B* this will either result in a successful call or a failed call–attempt. In the latter case *A* may repeat the call attempt later and thus initiate a series of several call–attempts which fail. Call statistics typically looks as shown in Table 2.1, where we have grouped the errors into a few typical classes. We notice that the only error which can be directly influenced by the operator is *technical errors and blocking*, and this class usually is small, a few percentages during the Busy Hour. Furthermore, we notice that the number of calls which experience *B–busy* depends on the number of *A-errors* and *technical errors & blocking*. Therefore, the statistics in Table 2.1 are misleading. To

Outcome	I–country	D–country
A-error:	15 %	20 %
Blocking and technical errors:	5 %	35 %
B no answer before A hangs up:	10 %	5 %
B-busy:	10 %	20 %
B-answer = conversation:	60 %	20 %
No conversation:	40 %	80 %

Table 2.1: *Typical outcome of a large number of call attempts during Busy Hour for Industrialised countries, respectively Developing countries.*

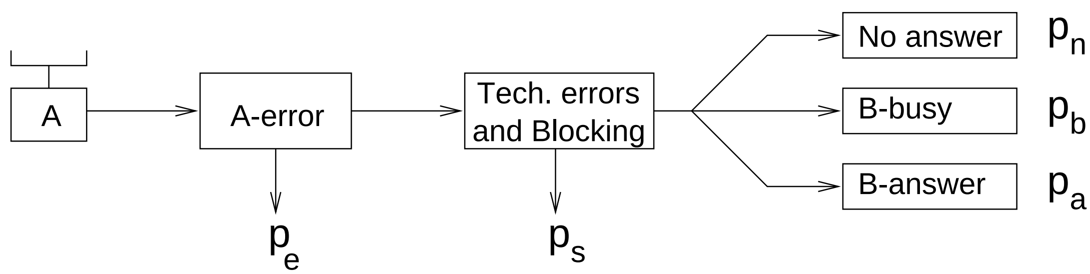


Figure 2.8: *When calculating the probabilities of events for a certain number of call attempts we have to consider the conditional probabilities.*

obtain the relevant probabilities, which are shown in Fig. 2.8, we shall only consider the calls arriving at the considered stage when calculating probabilities. Applying the notation in Fig. 2.8 we find the following probabilities for a call attempts (assuming independence):

$$p\{\text{A-error}\} = p_e \tag{2.5}$$

$$p\{\text{Congestion \& tech. errors}\} = (1-p_e)\cdot p_s \tag{2.6}$$

$$p\{\text{B–no answer}\} = (1-p_e)\cdot(1-p_s)\cdot p_n \tag{2.7}$$

$$p\{\text{B–busy}\} = (1-p_e)\cdot(1-p_s)\cdot p_b \tag{2.8}$$

$$p\{\text{B–answer}\} = (1-p_e)\cdot(1-p_s)\cdot p_a \tag{2.9}$$

Using the numbers from Table 2.1 we find the figures shown in Table 2.2. From this we notice that even if the A-subscriber behaves correctly and the telephone system is perfect, then only 75 %, respectively 45 % of the call attempts result in a conversation.

We distinguish between *the service time* which includes the time from the instant a server is occupied until the server becomes idle again (e.g. both call set-up, duration of the conversation, and termination of the call), and *conversation duration*, which is the time period where A talks with B. Because of failed call–attempts the mean service time is often less than the

I – country					D – country				
p_e	$=$	$\frac{15}{100}$	$=$	15%	p_e	$=$	$\frac{20}{100}$	$=$	20%
p_s	$=$	$\frac{5}{85}$	$=$	6%	p_s	$=$	$\frac{35}{80}$	$=$	44%
p_n	$=$	$\frac{10}{80}$	$=$	13%	p_n	$=$	$\frac{5}{45}$	$=$	11%
p_b	$=$	$\frac{10}{80}$	$=$	13%	p_b	$=$	$\frac{20}{45}$	$=$	44%
p_a	$=$	$\frac{60}{80}$	$=$	75%	p_a	$=$	$\frac{20}{45}$	$=$	44%

Table 2.2: *The relevant probabilities for the individual outcomes of the call attempts calculated for Table 2.1*

mean call duration if we include all call–attempts. Fig. 2.9 shows an example with observed holding times.

Example 2.4.1: Mean holding times
We assume that the mean holding time of calls which are interrupted before B-answer (A-error, congestion, technical errors) is 20 seconds and that the mean holding time for calls arriving at the called party (B-subscriber) (no answer, B-busy, B-answer) is 180 seconds. The mean holding time at the A-subscriber then becomes by using the figures in Table 2.1:

$$\text{I – country:} \qquad m_a = \frac{20}{100} \cdot 20 + \frac{80}{100} \cdot 180 = 148 \text{ seconds}$$

$$\text{D – country:} \qquad m_a = \frac{55}{100} \cdot 20 + \frac{45}{100} \cdot 180 = \ 92 \text{ seconds}$$

We thus notice that the mean holding time increases from 148s, respectively 92s, at the A-subscriber to 180s at the B-subscriber. If one call intent implies more repeated call attempts (cf. Example 2.4), then the carried traffic may become larger than the offered traffic. □

If we know the mean service time of the individual phases of a call attempt, then we can calculate the proportion of the call attempts which are lost during the individual phases. This can be exploited to analyse electro-mechanical systems by using *SPC-systems* to collect data.

Each call–attempt loads the controlling groups in the exchange (e.g. a computer or a control unit) with an almost constant load whereas the load of the network is proportional to the duration of the call. Because of this many failed call–attempts are able to overload the control devices while free capacity is still available in the network. Repeated call–attempts are not necessarily caused by errors in the telephone-system. They can also be caused by e.g. a busy B–subscriber. This problem were treated for the first time by Fr. Johannsen in *"Busy"* published in 1908 (Johannsen, 1908 [51]) . Fig. 2.10 and Fig. 2.11 show some examples from measurements of subscriber behaviour.

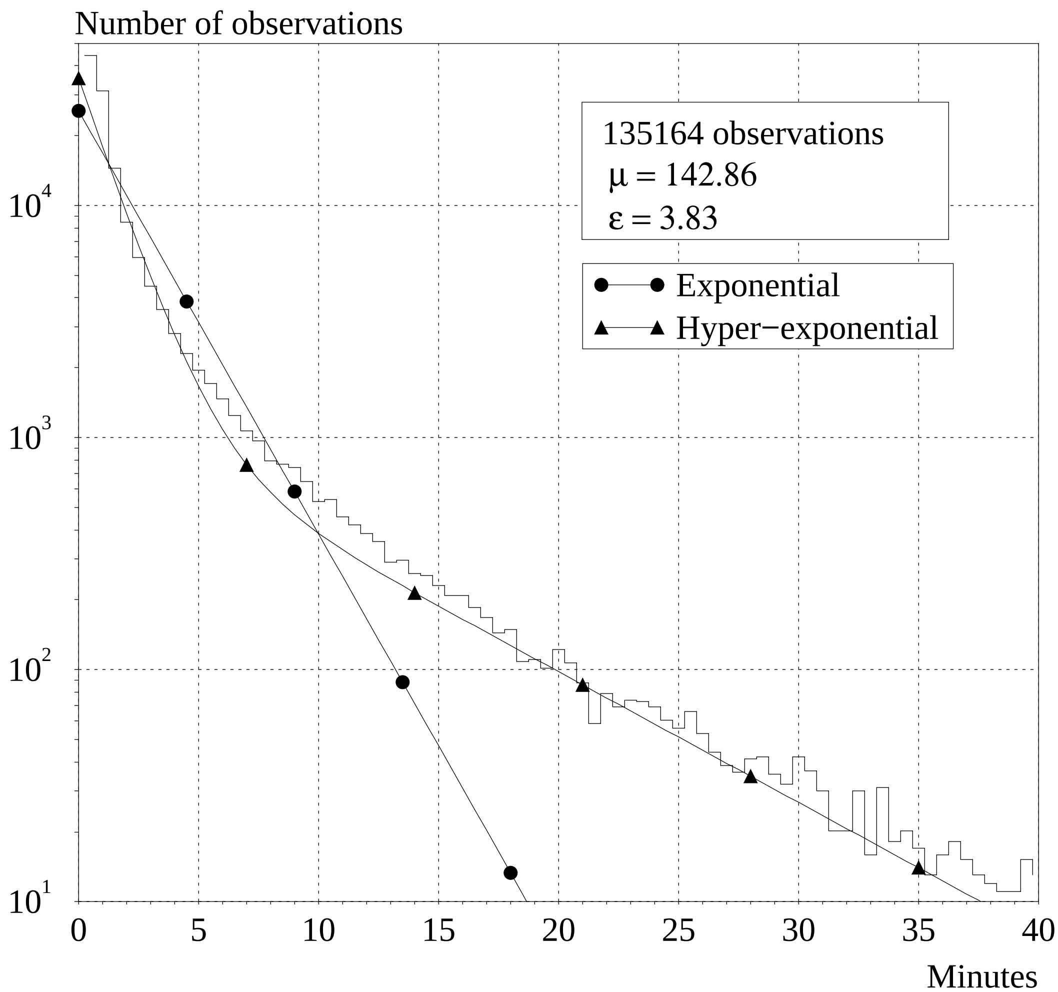


Figure 2.9: *Frequency function for holding times of trunks in a local switching centre.*

Studies of the subscribers response to for example busy tone is of vital importance for the dimensioning of telephone systems. In fact, *human–factors (= subscriber–behaviour)* is a part of the teletraffic theory which is of great interest.

During Busy Hour $\alpha = 10 - 16$ % of the subscribers are busy using the line for incoming or outgoing calls. Therefore, we would expect that α% of the call attempts experience B-busy. This is, however, wrong, because the subscribers have different traffic levels. Some subscribers receive no incoming call attempts, whereas others receive more than the average. In fact, it is so that the most busy subscribers on the average receive most call attempts. A-subscribers have an inclination to choose the most busy B-subscribers, and in practice we observe that the probability of B-busy is about $4 \cdot \alpha$, if we take no measures. For residential subscribers it is difficult to improve the situation. But for large business subscribers having a *PAX* (= *PABX*) (Private Automatic eXchange) with a group-number a sufficient number of lines will eliminate B-busy. Therefore, in industrialised countries the total probability of B-busy becomes of the same order of size as α (Table 2.1). For D–countries the traffic is more focused towards individual numbers and often the business subscribers don't benefit from group numbering, and therefore we observe a high probability of B-busy (40–50 %).

At the Ordrup measurements approximately 4% of the call were repeated call–attempts. If a subscriber experience blocking or B–busy there is 70% probability that the call is repeated within an hour. See Table 2.3.

	Number of observations				
Attempt no.	Success	Continue	Give up	$p\{\text{success}\}$	Persistence
		75.389			
1	56.935	7.512	10.942	0.76	0.41
2	3.252	2.378	1.882	0.43	0.56
3	925	951	502	0.39	0.66
4	293	476	182	0.31	0.72
5	139	248	89	0.29	0.74
> 5	134		114		
Total	61.678		13.711		

Table 2.3: *An observed sequence of repeated call–attempts (national calls, "Ordrup–measurements"). The probability of success decreases with the number of call–attempts, while the persistence increases. Here a repeated call–attempt is a call repeated to the same B–subscriber within one hour.*

A classical example of the importance of the subscribers reaction was seen when Valby gas-works (in Copenhagen) exploded in the mid sixties. The subscribers in Copenhagen generated a lot of call–attempts and occupied the controlling devices in the exchanges in the area of Copenhagen. Then subscribers from Esbjerg (western part of Denmark) phoning to Copenhagen had to wait because the dialled numbers could not be transferred to Copenhagen immediately. Therefore the equipment in Esbjerg was kept busy by waiting, and subscribers making local calls in Esbjerg could not complete the call attempts.

This is an example of how a overload situation spreads like a *chain reaction* throughout the network. The more tight a network has been dimensioned, the more likely it is that a chain reaction will occur. An exchange should always be constructed so that it keeps working with full capacity during overload situations.

In a modern exchange we have the possibility of giving priority to a group of subscribers in an emergency situation, e.g. doctors and police (*preferential traffic)*. In computer systems similar conditions will influence the performance. For example, if it is difficult to get a free entry to a terminal–system, the user will be disposed not to log off, but keep the terminal, i.e. increase the service time. If a system works as a waiting–time system, then the mean waiting time will increase with the third order of the mean service time (Chap. 13). Under these conditions the system will be saturated very fast, i.e. be overloaded. In countries with an overloaded telecommunication network (e.g. developing countries) a big percentage of the call–attempts will be repeated call–attempts.

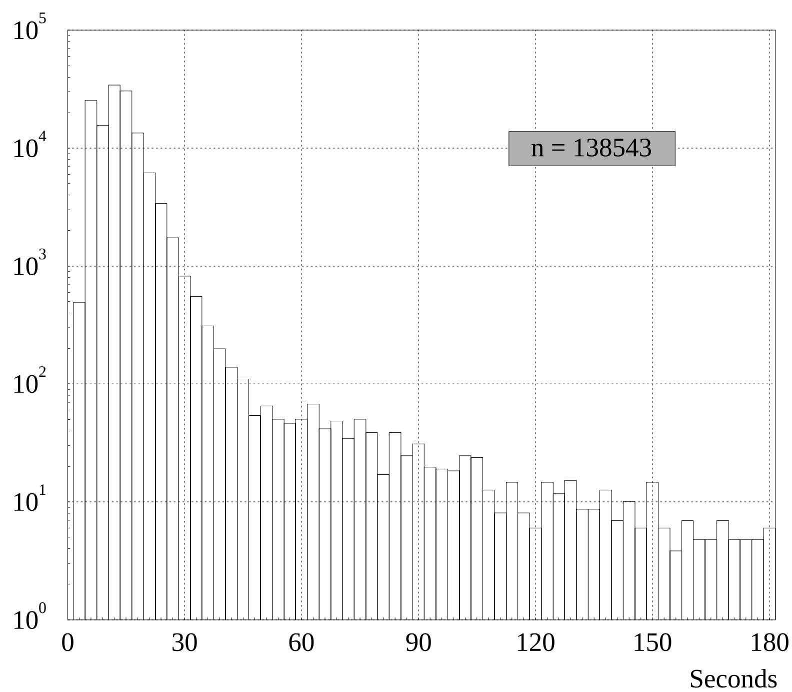


Figure 2.10: *Histogram for the time interval from occupation of register (dial tone) to B–answer for completed calls. The mean value is 13.60 s.*

Example 2.4.2: Repeated call attempt
This is an example of a simple model of repeated call attempts. Let us introduce the following notation:

$$b = \text{persistence} \tag{2.10}$$

$$B = \text{p}\{\text{non-completion}\} \tag{2.11}$$

The persistence b is the probability that an unsuccessful call attempt is repeated, and $p\{completion\}$ $= (1-B)$ is the probability that the B-subscriber (called party) answers. For one call intent we get the following history: We get the following probabilities for one call intent:

Attempt No.	p{B-answer}	p{Continue}	p{Give up}
0		1	
1	$(1-B)$	$B \cdot b$	$B \cdot (1-b)$
2	$(1-B) \cdot (B \cdot b)$	$(B \cdot b)^2$	$B \cdot (1-b) \cdot (B \cdot b)$
3	$(1-B) \cdot (B \cdot b)^2$	$(B \cdot b)^3$	$B \cdot (1-b) \cdot (B \cdot b)^2$
4	$(1-B) \cdot (B \cdot b)^3$	$(B \cdot b)^4$	$B \cdot (1-b) \cdot (B \cdot b)^3$
...	...	...	...
Total	$\dfrac{(1-B)}{(1-B \cdot b)}$	$\dfrac{1}{(1-B \cdot b)}$	$\dfrac{B \cdot (1-b)}{(1-B \cdot b)}$

Table 2.4: *A single call intent results in a series of call attempts. The distribution of the number of attempts is geometrically distributed.*

$$\text{p\{completion\}} = \frac{(1-B)}{(1-B \cdot b)} \tag{2.12}$$

$$\text{p\{non-completion\}} = \frac{B \cdot (1-b)}{(1-B \cdot b)} \tag{2.13}$$

$$\text{No. of call attempts per call intent} = \frac{1}{(1-B \cdot b)} \tag{2.14}$$

Let us assume the following mean holding times:

s_c = mean holding time of completed calls

$s_n = 0$ = mean holding time of non-completed calls

Then we get the following relations between the traffic carried Y and the traffic offered A:

$$Y = A \cdot \frac{1-B}{1-B \cdot b} \tag{2.15}$$

$$A = Y \cdot \frac{1-B \cdot b}{1-B} \tag{2.16}$$

This is similar to the result given in ITU–T Rec. E.502. □

In practice, the persistence b and the probability of completion $1-B$ will depend on the number of times the call has been repeated (cf. Table 2.3). If the unsuccessful calls have a positive mean holding time, then the carried traffic may become larger than the offered traffic.

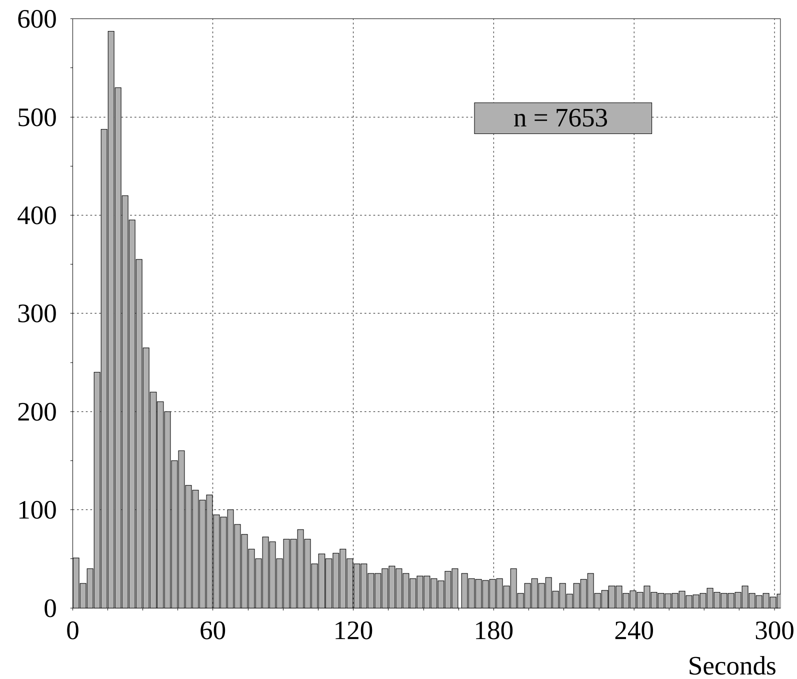


Figure 2.11: *Histogram for all call attempts repeated within 5 minutes, when the called party is busy.*

2.5 Introduction to Grade-of-Service = GoS

The following section is based on (Veirø, 2001 [99]). A network operator must decide what services the network should deliver to the end user and the level of service quality that the user should experience. This is true for any telecommunications network, whether it is circuit- or packet-switched, wired or wireless, optical or copper-based, and it is independent of the transmission technology applied. Further decisions to be made may include the type and layout of the network infrastructure for supporting the services, and the choice of techniques to be used for handling the information transport. These further decisions may be different, depending on whether the operator is already present in the market, or is starting service from a greenfield situation (i.e. a situation where there is no legacy network in place to consider).

As for the Quality of Service (QoS) concept, it is defined in the ITU-T Recommendation E.800 as: *The collective effect of service performance, which determine the degree of satisfaction of a user of the service.* The QoS consists of a set of parameters that pertain to the traffic performance of the network, but in addition to this, the QoS also includes a lot of other concepts. They can be summarised as:

- service support performance
- service operability performance
- serveability performance and
- service security performance

The detailed definitions of these terms are given in the E.800. The better service quality an operator chooses to offer to the end user, the better is the chance to win customers and to keep current customers. But a better service quality also means that the network will become more expensive to install and this, normally, also has a bearing to the price of the service. The choice of a particular service quality therefore depends on political decisions by the operator and will not be treated further here.

When the quality decision is in place the planning of the network proper can start. This includes the decision of a transport network technology and its topology as well as reliability aspects in case one or more network elements become malfunctioning. It is also at this stage where the routing strategy has to be determined.

This is the point in time where it is needed to consider the Grade of Service (GoS). This is defined in the ITU-T Recommendation E.600 as: *A number of traffic engineering variables to provide a measure of adequacy of a group of ressources under specified conditions. These grade of service variables may be probability of loss, dial tone delay, etc.* To this definition the recommendation furthermore supplies the following notes:

- The parameter values assigned for grade of service variables are called grade of service standards.
- The values of grade of service parameters achieved under actual conditions are called grade of service results.

The key point to solve in the determination of the GoS standards is to apportion individual values to each network element in such a way that the target end-to-end QoS is obtained.

2.5.1 Comparison of GoS and QoS

It is not an easy task to find the GoS standards needed to support a certain QoS. This is due to the fact that the GoS and QoS concepts have different viewpoints. While the QoS views the situation from the customer's point of view, the GoS takes the network point of view. We illustrate this by the following example:

Example 2.5.1:
Say we want to fix the end to end call blocking probability at 1 % in a telephone network. A

customer will interpret this quantity to mean that he will be able to reach his destinations in 99 out of 100 cases on the average. Fixing this design target, the operator apportioned a certain blocking probability to each of the network elements, which a reference call could meet. In order to make sure that the target is met, the network has to be monitored. But this monitoring normally takes place all over the network and it can only be ensured that the network on the average can meet the target values. If we consider a particular access line its GoS target may well be exceeded, but the average for all access lines does indeed meet the target. □

GoS pertains to parameters that can be verified through network performance (the ability of a network or network portion to provide the functions related to *communications between users*) and the parameters hold only on average for the network. Even if we restrain ourselves only to consider the part of the QoS that is *traffic* related, the example illustrates, that even if the GoS target is fulfilled this need not be the case for the QoS.

2.5.2 Special features of QoS

Due to the different views taken by GoS and QoS a solution to take care of the problem has been proposed. This solution is called a service level agreement (*SLA*). This is really a contract between a user and a network operator. In this contract it is defined what the parameters in question really mean. It is supposed to be done in such a way, that it will be understood in the same manner by the customer and the network operator. Furthermore the SLA defines, what is to happen in case the terms of the contract are violated. Some operators have chosen to issue an SLA for all customer relationships they have (at least in principle), while others only do it for big customers, who know what the terms in the SLA really mean.

2.5.3 Network performance

As mentioned above the network performance concerns the ability of a network or network portion to provide the functions related to communications between users. In order to establish how a certain network performs, it is necessary to perform measurements and the measurements have to cover all the aspects of the performance parameters (i.e. trafficability, dependability, transmission and charging).

Furthermore, the network performance aspects in the GoS concept pertains only to the factors related to trafficability performance in the QoS terminology. But in the QoS world *network performance* also includes the following concepts:

- dependability,
- transmission performance, and
- charging correctness.

It is not enough just to perform the measurements. It is also necessary to have an organisation that can do the proper surveillance and can take appropriate action when problems arise. As the network complexity keeps growing so does the number of parameters needed to consider. This means that automated tools will be required in order to make it easier to get an overview of the most important parameters to consider.

2.5.4 Reference configurations

In order to obtain an overview of the network under consideration, it is often useful to produce a so-called reference configuration. This consists of one or more simplified drawing(s) of the path a call (or connection) can take in the network including appropriate reference points, where the interfaces between entities are defined. In some cases the reference points define an interface between two operators, and it is therefore important to watch carefully what happens at this point. From a *GoS* perspective the importance of the reference configuration is the partitioning of the *GoS* as described below. Consider a telephone network with terminals, subscriber switches and transit switches. In the example we ignore the signalling network. Suppose the call can be routed in one of three ways:

1. terminal → subscriber switch → terminal

   This is drawn as a reference configuration shown in Fig. 2.12.

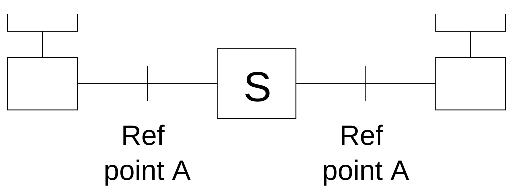


Figure 2.12: *Reference configuration for case 1.*

2. terminal → subscriber switch → transit switch → subscriber switch → terminal

   This is drawn as a reference configuration shown in Fig. 2.13.

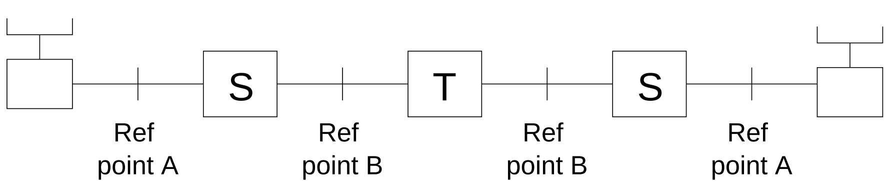


Figure 2.13: *Reference configuration for case 2.*

3. terminal → subscriber switch → transit switch → transit switch → subscriber switch → terminal
   This is drawn as a reference configuration shown in Fig. 2.14.

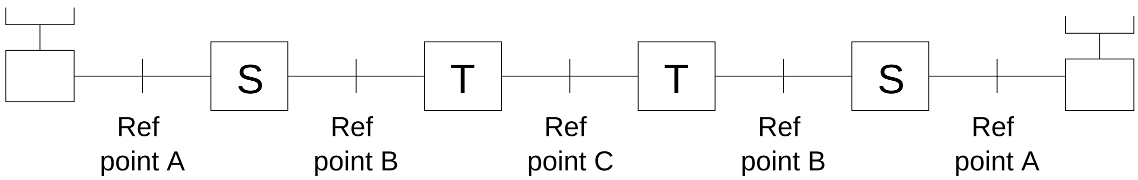


Figure 2.14: *Reference configuration for case 3.*

Based on a given set of QoS requirements, a set of *GoS* parameters are selected and defined on an end-to-end basis within the network boundary, for each major service category provided by a network. The selected *GoS* parameters are specified in such a way that the *GoS* can be derived at well-defined reference points, i.e. *traffic significant points*. This is to allow the partitioning of end-to-end *GoS* objectives to obtain the *GoS* objectives for each network stage or component, on the basis of some well-defined reference connections.

As defined in Recommendation E.600, for traffic engineering purposes, a connection is an association of resources providing means for communication between two or more devices in, or attached to, a telecommunication network. There can be different types of connections as the number and types of resources in a connection may vary. Therefore, the concept of a reference connection is used to identify representative cases of the different types of connections without involving the specifics of their actual realizations by different physical means.

Typically, different network segments are involved in the path of a connection. For example, a connection may be local, national, or international. The purposes of reference connections are for clarifying and specifying traffic performance issues at various interfaces between different network domains. Each domain may consist of one or more service provider networks. Recommendation I.380/Y.1540 defines performance parameters for *IP* packet transfer; its companion Draft Recommendation Y.1541 specifies the corresponding allocations and performance objectives. Recommendation E.651 specifies reference connections for *IP-access* networks. Other reference connections are to be specified.

From the QoS objectives, a set of end-to-end *GoS* parameters and their objectives for different reference connections are derived. For example, end-to-end connection blocking probability and end-to-end packet transfer delay may be relevant *GoS* parameters. The GoS objectives should be specified with reference to traffic load conditions, such as under normal and high load conditions. The end-to-end GoS objectives are then apportioned to individual resource components of the reference connections for dimensioning purposes. In an operational network, to ensure that the GoS objectives have been met, performance measurements and performance monitoring are required.

In IP-based networks, performance allocation is usually done on a *cloud*, i.e. the set of routers and links under a single (or collaborative) jurisdictional responsibility, such as an Internet Service Provider, *ISP*. A cloud is connected to another cloud by a link, i.e. a gateway router in one cloud is connected via a link to a gateway router in another cloud. End-to-end communication between hosts is conducted on a path consisting of a sequence of clouds and interconnecting links. Such a sequence is referred to as a hypothetical reference path for performance allocation purposes.

Chapter 3

Probability Theory and Statistics

All time intervals we consider are non-negative, and therefore they can be expressed by non-negative *random variables*. A random variable is also called a *variate*. Time intervals of interests are, for example, service times, duration of congestion (blocking periods, busy periods), waiting times, holding times, *CPU*-busy times, inter-arrival times, etc. We denote these time durations as *lifetimes* and their distribution functions as *time distributions*. In this chapter we review the basic theory of probability and statistics relevant to teletraffic theory.

3.1 Distribution functions

A time interval can be described by a random variable T that is characterised by a distribution function $F(t)$:

$$\begin{aligned} F(t) &= \int_{0-}^{t} \mathrm{d}F(u) \qquad \text{for} \quad 0 \leq t < \infty \,, \\ F(t) &= 0 \qquad\qquad\qquad \text{for} \quad t < 0 \,. \end{aligned} \tag{3.1}$$

In (3.1) we integrate from $0-$ to keep record of a possible discontinuity at $t = 0$. When we consider waiting time systems, there is often a positive probability to have waiting times equal to zero, i.e. $F(0) \neq 0$. On the other hand, when we look at the inter-arrival times, we usually assume $F(0) = 0$ (Sec. 5.2.3).

The probability that the duration of a time interval is less than or equal to t becomes:

$$p(T \leq t) = F(t) \,.$$

Sometimes it is easier to consider the *complementary distribution function*:

$$F^c(t) = 1 - F(t)\,.$$

This is also called the *survival distribution function*.

We often assume that $F(t)$ is differentiable and that the following density function $f(t)$ exists:

$$\mathrm{d}F(t) = f(t)\cdot \mathrm{d}t = p\{t < T \le t + \mathrm{d}t\}, \qquad t \ge 0\,. \tag{3.2}$$

Usually, we assume that the service time is independent of the arrival process and that a service time is independent of other service times. Analytically, many calculations can be carried out for any time distribution. In general, we always assume that the mean value exists.

3.1.1 Characterisation of distributions

Time distributions which only assume positive arguments possess some advantageous properties. For the i'th *non-central moment*, which we usually denote the i'th moment, it may be shown that *Palm's identity* is valid:

$$\begin{aligned} E\{T^i\} = m_i &= \int_0^\infty t^i \cdot f(t)\,\mathrm{d}t & (3.3)\\ &= \int_0^\infty i\,t^{i-1}\cdot\{1-F(t)\}\,\mathrm{d}t\,, \qquad i = 1, 2, \ldots\,. \end{aligned}$$

Palm's identity (3.3), which is valid for life-time distributions (only defined for non-negative arguments), was first proved in (Palm, 1943,[79]) as follows.

$$\begin{aligned} \int_{t=0}^\infty i\,t^{i-1}\,\{1-F(t)\}\;\mathrm{d}t &= \int_{t=0}^\infty i\,t^{i-1}\left\{\int_{x=t}^\infty f(x)\,\mathrm{d}x\right\}\;\mathrm{d}t\\ &= \int_{t=0}^\infty\int_{x=t}^\infty i\,t^{i-1} f(x)\,\mathrm{d}x\,\mathrm{d}t\\ &= \int_{t=0}^\infty\int_{x=t}^\infty \mathrm{d}t^i\, f(x)\,\mathrm{d}x\\ &= \int_{x=0}^\infty\int_{t=0}^x \mathrm{d}t^i\, f(x)\,\mathrm{d}x\\ &= \int_{x=0}^\infty x^i\, f(x)\,\mathrm{d}x\\ &= m_i\,. \end{aligned}$$

The order of integration can be inverted because the integrand is non-negative.
Thus we have proved (3.3):

$$m_i = \int_0^\infty t^i \cdot f(t)\,\mathrm{d}t = \int_0^\infty i\,t^{i-1} \cdot \{1 - F(t)\}\,\mathrm{d}t\,, \quad i = 1, 2, \ldots .$$

The following simplified proof is correct because we assume that the moments exist:

$$\begin{aligned} m_i &= \int_{t=0}^\infty t^i\, f(t)\,\mathrm{d}t \\ &= -\int_{t=0}^\infty t^i\,\mathrm{d}\,\{1 - F(t)\} \\ &= -t^i\,\{1 - F(t)\}\,|_0^\infty + \int_{t=0}^\infty \{1 - F(t)\}\,\mathrm{d}t^i \\ &= \int_{t=0}^\infty i\,t^{i-1}\,\{1 - F(t)\}\,\mathrm{d}t \qquad \textit{q.e.d.} \end{aligned}$$

Example 3.1.1: Exponential distribution
For the exponential distribution we get:

$$m_2 = \int_{t=0}^\infty t^2\,\lambda\,\mathrm{e}^{-\lambda t}\,\mathrm{d}t = \int_{t=0}^\infty 2\,t\,\mathrm{e}^{-\lambda t}\,\mathrm{d}t = \frac{2}{\lambda^2}\,.$$

It may be surprising that the two integrals are identical. The two integrands can, apart from a constant, be transformed to an Erlang–3, respectively an Erlang–2, density function (4.8), which has the total probability mass one:

$$m_2 = \frac{2}{\lambda^2}\int_{t=0}^\infty \frac{(\lambda t)^2}{2}\,\mathrm{e}^{-\lambda t}\,\lambda\,\mathrm{d}t = \frac{2}{\lambda^2}\int_{t=0}^\infty \lambda t\,\mathrm{e}^{-\lambda t}\,\lambda\,\mathrm{d}t \; = \frac{2}{\lambda^2}\,.$$

□

Example 3.1.2: Constant time interval
For a constant time interval of duration h we have:

$$m_i = h^i\,.$$

□

Especially, we have the first two moments under the assumption that they exist:

$$m_1 = \int_0^\infty t\,f(t)\,\mathrm{d}t = \int_0^\infty \{1 - F(t)\}\mathrm{d}t\,, \tag{3.4}$$

$$m_2 = \int_0^\infty t^2\,f(t)\,\mathrm{d}t = \int_0^\infty 2t \cdot \{1 - F(t)\}\,\mathrm{d}t\,. \tag{3.5}$$

The *mean value* (expectation) is the first moment and often we leave out the index:

$$m = m_1 = E\{T\}\,. \tag{3.6}$$

The *i'th central moment* is defined as:

$$E\{(T-m_1)^i\} = \int_0^\infty (t-m_1)^i\, f(t)\,\mathrm{d}t\,. \tag{3.7}$$

The *variance* is the *2nd central moment:*

$$\sigma^2 = E\{(T-m_1)^2\}\,.$$

It is easy to show that:

$$\begin{aligned} \sigma^2 &= m_2 - m_1^2 \qquad \text{or} \\ m_2 &= \sigma^2 + m_1^2\,. \end{aligned} \tag{3.8}$$

A distribution is normally uniquely defined by all its moments. A normalised measure for the irregularity (dispersion) of a distribution is the *coefficient of variation*. It is defined as the ratio between the standard deviation and the mean value:

$$CV = \text{Coefficient of Variation} = \frac{\sigma}{m_1}\,. \tag{3.9}$$

This quantity is dimensionless, and we shall later apply it to characterise discrete distributions (state probabilities). Another measure of irregularity is *Palm's form factor* ε, which is defined as follows:

$$\varepsilon = \frac{m_2}{m_1^2} = 1 + \left(\frac{\sigma}{m_1}\right)^2 \geq 1\,. \tag{3.10}$$

The form factor ε as well as (σ/m_1) are independent of the choice of time scale, and they will appear in many formulæ in the following.

The larger a form factor, the more irregular is the time distribution, and the larger will for example the mean waiting time in a waiting time system be. The form factor takes the minimum value equal to one for constant time intervals ($\sigma = 0$).

To estimate a distribution from observations, we are often satisfied by knowing the first two moments (m and σ or ε) as higher order moments require extremely many observations to obtain reliable estimates. Time distributions can also be characterised in other ways. We consider some important ones below.

3.1.2 Residual lifetime

We wish to find the distribution of the residual life time, given that a certain age $x \geq 0$ has already been obtained.

The conditional distribution $F(t+x|x)$ is defined as follows, assuming $p\{T>x\}>0$ and $t \geq 0$:

$$\begin{aligned} p\{T>t+x|T>x\} &= \frac{p\{(T>t+x)\wedge(T>x)\}}{p\{T>x\}} \\ &= \frac{p\{T>t+x\}}{p\{T>x\}} \\ &= \frac{1-F(t+x)}{1-F(x)}\,, \end{aligned}$$

and thus:

$$\begin{aligned} F(t+x|x) &= p\{T \leq t+x \,|\, T>x\} \\ &= \frac{F(t+x)-F(x)}{1-F(x)}\,, \end{aligned} \tag{3.11}$$

$$f(t+x|x) = \frac{f(t+x)}{1-F(x)}\,. \tag{3.12}$$

Fig. 3.1 illustrates these calculations graphically.

The mean value $m_{1,r}$ of the residual lifetime can be written as (3.4):

$$m_{1,r}(x) = \frac{1}{1-F(x)} \cdot \int_{t=0}^{\infty} \{1-F(t+x)\}\,\mathrm{d}t, \qquad x \geq 0\,. \tag{3.13}$$

The *Death rate at time* x, i.e. the probability, that the considered lifetime terminates within an interval $(x, x+\mathrm{d}x)$, under the condition that age x has been achieved, is obtained from (3.11) by letting $t = \mathrm{d}x$:

$$\begin{aligned} \mu(x) \cdot \mathrm{d}x &= \frac{F(x+\mathrm{d}x)-F(x)}{1-F(x)} \\ &= \frac{\mathrm{d}F(x)}{1-F(x)}\,. \end{aligned} \tag{3.14}$$

The conditional density function $\mu(x)$ is also called the *hazard function*. If this function is given, then $F(x)$ may be obtained as the solution to the following differential equation:

$$\frac{\mathrm{d}F(x)}{\mathrm{d}x} + \mu(x) \cdot F(x) = \mu(x)\,, \tag{3.15}$$

which has the following solution (assuming $F(0)=0$):

$$F(t) = 1 - \exp\left\{-\int_0^t \mu(u)\,\mathrm{d}u\right\}\,, \tag{3.16}$$

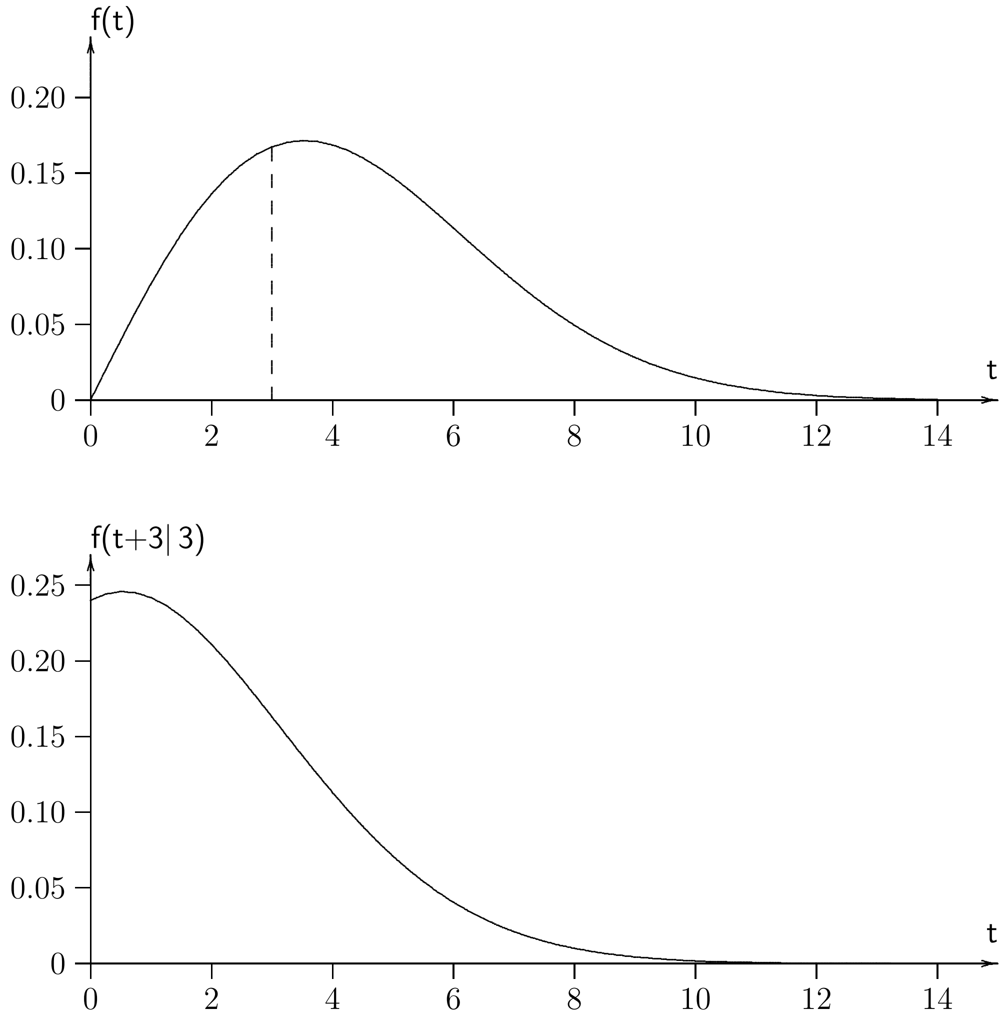


Figure 3.1: *The density function of the residual life time conditioned by a given age x (3.11). The example is based on a Weibull distribution We(2,5) where $x = 3$ and $F(3) = 0.3023$.*

$$f(t) = \mu(t) \cdot \exp\left\{ -\int_0^t \mu(u)\,\mathrm{d}u \right\} . \tag{3.17}$$

The death rate $\mu(t)$ is constant if and only if the lifetime is exponentially distributed (Chap. 4). This is a fundamental characteristic of the exponential distribution which is called the Markovian property (*lack of memory* (age)): The probability of terminating is independent of the actual age (history) (Sec. 4.1).

One would expect that the *mean residual lifetime* $m_{1,r}(x)$ decreases for increasing x, corresponding to that the expected *residual lifetime* decreases when the age x increases. This is not always the case. For an exponential distribution with form factor $\varepsilon = 2$ (Sec. 5.1), we have $m_{1,r} = m$. For steep distributions $(1 \leq \varepsilon \leq 2)$ we have $m_{1,r} \leq m$ (Sec. 4.2), whereas for flat distributions $(2 \leq \varepsilon < \infty)$, we have $m_{1,r} \geq m$ (Sec. 4.3).

Example 3.1.3: Waiting-time distribution
The waiting time distribution $W_s(t)$ for a random customer usually has a positive probability mass (atom) at $t = 0$, because some of the customers get service immediately without waiting. We thus have $W_s(0) > 0$. The waiting time distribution $W_+(t)$ for customers having positive waiting times then becomes (3.11):

$$W_+(t) = \frac{W_s(t) - W_s(0)}{1 - W_s(0)}\,,$$

or if we denote the probability of a positive waiting time $\{1 - W_s(0)\}$ by D (probability of delay):

$$D \cdot \{1 - W_+(t)\} = 1 - W_s(t)\,. \tag{3.18}$$

For the density function we have (3.11):

$$D \cdot w_+(t) = w_s(t)\,. \tag{3.19}$$

For mean values we get:

$$D \cdot w = W\,, \tag{3.20}$$

where the mean value for all customers is denoted by W, and the mean value for the delayed customers is denoted by w. These formulæ are valid for any queueing system. □

3.1.3 Load from holding times of duration less than x

So far we have attached the same importance to all lifetimes independently of their duration. The importance of a lifetime is often proportional to its duration, for example when we consider the load of queueing system, charging of *CPU*-times, telephone conversations etc.

If we allocate a weight factor to a life time proportional to its duration, then the average weight of all time intervals (of course) becomes equal to the mean value:

$$m = \int_0^\infty t\, f(t)\, \mathrm{d}t\,, \tag{3.21}$$

where $f(t)\, dt$ is the probability of an observation within the interval $(t, t + \mathrm{d}t)$, and t is the weight of this observation.

In a traffic process we are interested in calculating the proportion of the total traffic which is due to holding times of duration less than x:

$$\rho_x = \frac{\displaystyle\int_0^x t\, f(t)\, \mathrm{d}t}{m}\,. \tag{3.22}$$

(This is the same as the proportion of the mean value which is due to contributions from lifetimes less than x).

Often relatively few service times make up a relatively large proportion of the total load. From Fig. 3.2 we see that if the form factor ε is 5, then 75% of the service times only contribute with 30% of the total load (Vilfred Pareto's rule). This fact can be utilised to give priority to short tasks without delaying the longer tasks very much (Chap. 13).

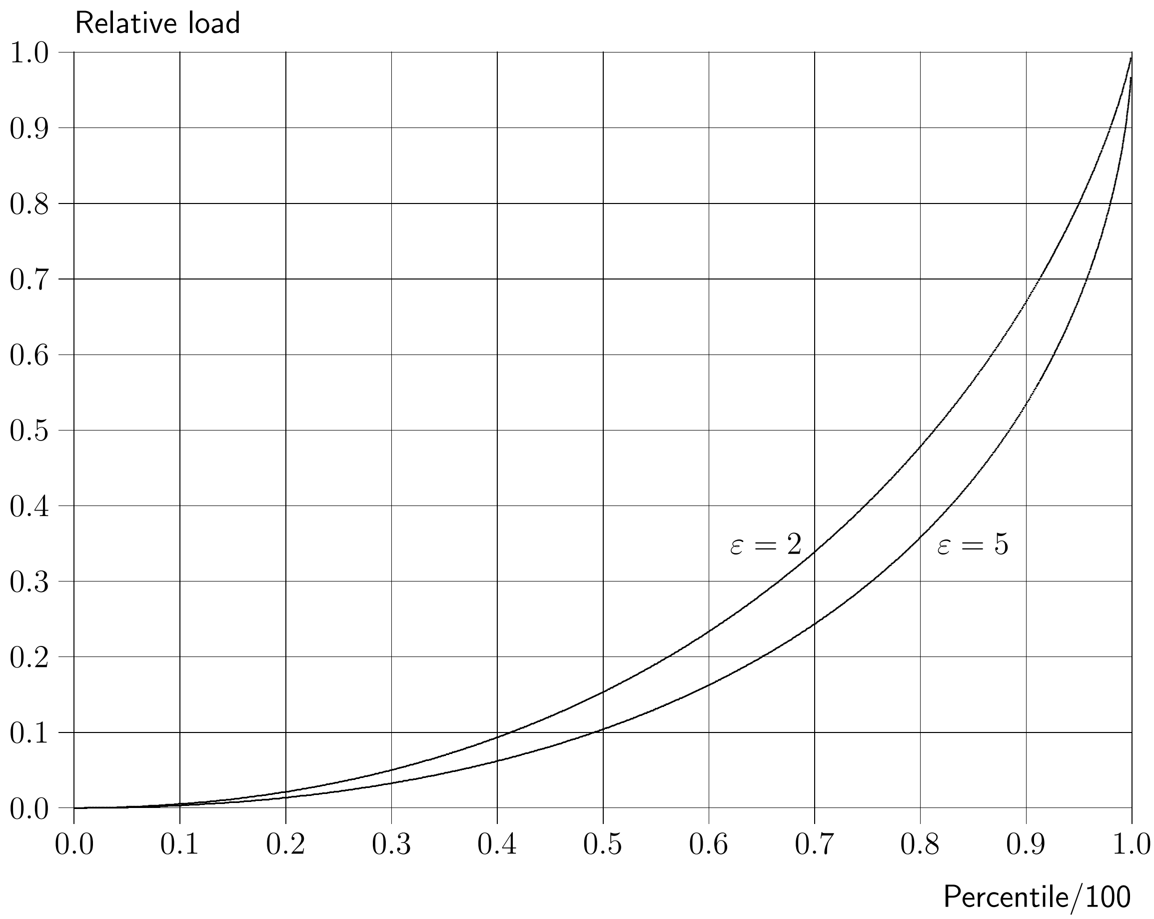


Figure 3.2: *Example of the relative traffic load from holding times shorter than a given value given by the percentile of the holding time distribution (3.22). Here $\varepsilon = 2$ corresponds to an exponential distribution and $\varepsilon = 5$ corresponds to a Pareto-distribution. We note that the* 10% *largest holding times contributes with* 33%*, respectively* 47%*, of the load (cf. customer averages and time averages in Chap. 5).*

3.1.4 Forward recurrence time

The residual lifetime from a random point of time is called the *forward recurrence time*. In this section we shall derive some important formulæ. To formulate the problem we consider an example. We wish to investigate the lifetime distribution of cars and ask car-owners chosen at random about the age of their car. As the point of time is chosen at random, then the probability of choosing a car is proportional to the total lifetime of the car. The distribution of the future residual lifetime will then be identical with the already achieved lifetime.

By choosing a sample in this way, the probability of choosing a car is proportional to the lifetime of the car, i.e. we will preferably choose cars with longer lifetimes (length-biased sampling). The probability of choosing a car having a total lifetime x is given by (cf. moment distribution in statistics) (cf. the derivation of (3.22)):

$$\frac{x\, f(x)\, \mathrm{d}x}{m}\,.$$

As we consider a random point of time, the distribution of the remaining lifetime will be uniformly distributed in $(0, x\,]$:

$$f(t|\, x) = \frac{1}{x}\,, \quad 0 < t \leq x\,.$$

Then the density function of the remaining lifetime at a random point of time is as follows:

$$\begin{aligned} v(t) &= \int_t^\infty \frac{1}{x} \cdot \frac{x}{m} \cdot f(x)\, \mathrm{d}x\,, \\ v(t) &= \frac{1 - F(t)}{m}\,. \end{aligned} \tag{3.23}$$

where $F(t)$ is the distribution function of the total lifetime and m is the mean value.

By applying the identity (3.3), we note that the i'th moment of $v(t)$ is given by the $(i+1)$'th moment of $f(t)$:

$$\begin{aligned} m_{i,v} &= \int_0^\infty t^i\, v(t)\, \mathrm{d}t \\ &= \int_0^\infty t^i\, \frac{1 - F(t)}{m}\, \mathrm{d}t \\ &= \frac{1}{i+1} \cdot \frac{1}{m} \cdot \int_0^\infty (i+1) \cdot t^i \cdot \{1 - F(t)\}\, \mathrm{d}t\,, \\ m_{i,v} &= \frac{1}{i+1} \cdot \frac{1}{m} \cdot m_{i+1,f}\,. \end{aligned} \tag{3.24}$$

We obtain the mean value:

$$m_{1,v} = \frac{m}{2} \cdot \varepsilon\,, \tag{3.25}$$

where ε is the form factor of the lifetime distribution. These formulæ are also valid for discrete time distributions.

3.1.5 Distribution of the j'th largest of k random variables

Let us assume that k random variables $\{T_1, T_2, \ldots, T_k\}$ are independent and identically distributed with distribution function $F(t)$. The distribution of the j'th largest variable will then be given by:

$$p\{j\text{'th largest} \leq t\} = \sum_{i=0}^{j-1} \binom{k}{i} \{1 - F(t)\}^i \, F(t)^{k-i} \,. \tag{3.26}$$

as at most $j-1$ variables may be larger than t. The smallest one (or k'th largest, $j=k$) has the distribution function:

$$F_{\min}(t) = 1 - \{1 - F(t)\}^k \,, \tag{3.27}$$

and the largest one ($j=1$) has the distribution function:

$$F_{\max}(t) = F(t)^k \,. \tag{3.28}$$

If the random variables has individual distribution functions $F_i(t)$, we get an expression more complex than (3.26). For the smallest and the largest we get:

$$F_{\min}(t) = 1 - \prod_{i=1}^{k} \{1 - F_i(t)\} \,, \tag{3.29}$$

$$F_{\max}(t) = \prod_{i=1}^{k} F_i(t) \,. \tag{3.30}$$

3.2 Combination of random variables

We can combine lifetimes by putting them in series or in parallel or by a combination of the two.

3.2.1 Random variables in series

A linking in series of k independent time intervals corresponds to addition of k independent random variables, i.e. convolution of the random variables.

If we denote the mean value and the variance of the i'th time interval by $m_{1,i}$, σ_i^2, respectively, then the sum of the random variables has the following mean value and variance:

$$m = m_1 = \sum_{i=1}^{k} m_{1,i} \,, \tag{3.31}$$

$$\sigma^2 = \sum_{i=1}^{k} \sigma_i^2 \,. \tag{3.32}$$

In general, we should add the so-called cumulants, and the first three cumulants are identical with the first three central moments.

The distribution function of the sum is obtained by the convolution:

$$F(t) = F_1(t) \otimes F_2(t) \otimes \cdots \otimes F_k(t)\,, \tag{3.33}$$

where $\otimes$ is the convolution operator (Sec. 6.2.2).

Example 3.2.1: Binomial distribution and Bernoulli trials
Let the probability of success in a trial (e.g. throwing a dice) be equal to p and the probability of failure thus equal to $1-p$. The number of successes in a single trial will then be given by the Bernoulli distribution:

$$p_1(i) = \begin{cases} 1-p\,, & i=0\,,\\ p\,, & i=1\,. \end{cases} \tag{3.34}$$

If we in total make S trials, then the distribution of number of successes is Binomial distributed:

$$p_S(i) = \binom{S}{i} p^i\,(1-p)^{S-i}\,, \tag{3.35}$$

which therefore is obtainable by convolving S Bernoulli distributions. If we make one additional trial, then the distribution of the total number of successes is obtained by convolution of the Binomial distribution (3.35) and the Bernoulli distribution (3.34):

$$\begin{aligned}
p_{S+1}(i) &= p_S(i)\cdot p_1(0) + p_S(i-1)\cdot p_1(1)\\
&= \binom{S}{i} p^i\,(1-p)^{S-i}\cdot(1-p) + \binom{S}{i-1} p^{i-1}\,(1-p)^{S-i+1}\cdot p\\
&= \left\{\binom{S}{i} + \binom{S}{i-1}\right\} p^i\,(1-p)^{S-i+1}\\
&= \binom{S+1}{i} p^i\,(1-p)^{S-i+1}\,, \qquad \textit{q.e.d.}
\end{aligned}$$

□

3.2.2 Random variables in parallel

By the weighting of ℓ independent random variables, where the i'th variable appears with weight factor p_i, where

$$\sum_{i=1}^{\ell} p_i = 1\,,$$

and has mean value $m_{1,i}$ and variance σ_i^2, the random variable of the sum has the mean value and variance as follows:

$$m = \sum_{i=1}^{\ell} p_i \cdot m_{1,i}\,, \tag{3.36}$$

$$\sigma^2 = \sum_{i=1}^{\ell} p_i \cdot (\sigma_i^2 + m_{1,i}^2) - m^2\,. \tag{3.37}$$

In this case we must weight the non-central moments. For the j'th moment we have

$$m_j = \sum_{i=1}^{\ell} p_i \cdot m_{j,i}\,, \tag{3.38}$$

where $m_{j,i}$ is the j'th non-central moment of the distribution of the i'th interval.

The distribution function (compound distribution) is as follows:

$$F(t) = \sum_{i=1}^{\ell} p_i \cdot F_i(t)\,. \tag{3.39}$$

A similar formula is valid for the density function.

3.3 Stochastic sum

By a stochastic sum we understand the sum of a stochastic number of random variables (Feller, 1950 [27]). Let us consider a trunk group without congestion, where the arrival process and the holding times are stochastically independent. If we consider a fixed time interval T, then the number of arrivals is a random variable N. In the following N is characterised by:

$$\begin{aligned} N: \quad & \text{density function } \; p(i)\,, \\ & \text{mean value } \; m_{1,n}\,, \\ & \text{variance } \; \sigma_n^2\,, \end{aligned} \tag{3.40}$$

Arriving call number i has the holding time T_i. All T_i have the same distribution, and each arrival (request) will contribute with a certain number of time units (the holding times) which is a random variable characterised by:

$$\begin{aligned} T: \quad & \text{density function } \; f(t)\,, \\ & \text{mean value } \; m_{1,t}\,, \\ & \text{variance } \; \sigma_t^2\,, \end{aligned} \tag{3.41}$$

The total traffic volume generated by all arrivals (requests) arriving within the considered time interval T is then a random variable itself:

$$S_T = T_1 + T_2 + \cdots + T_N \,. \tag{3.42}$$

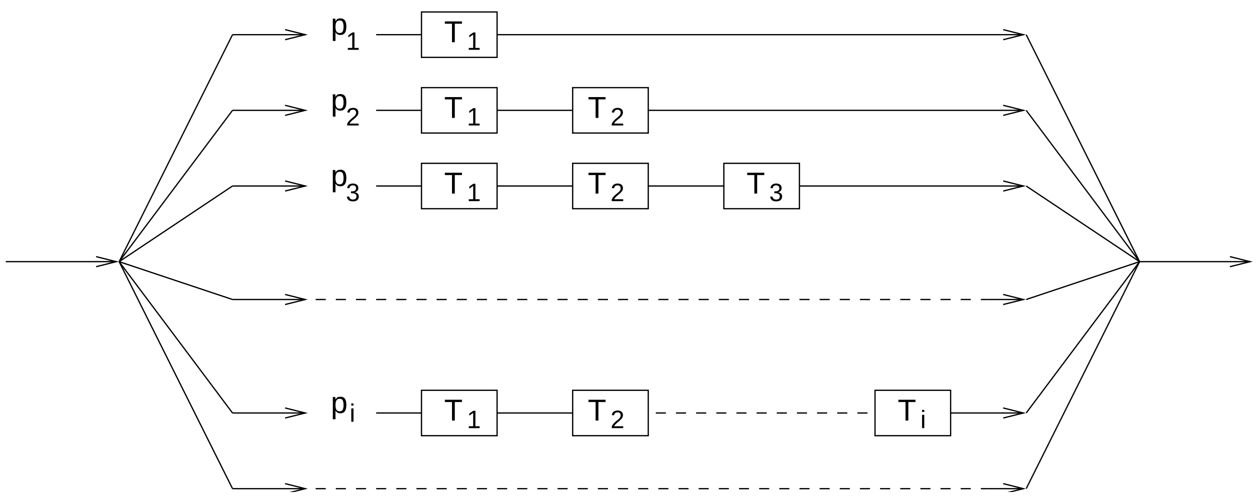


Figure 3.3: *A stochastic sum may be interpreted as a series/parallel combination of random variable.*

In the following we assume that T_i and N are stochastically independent. This will be fulfilled when the congestion is zero.

The following derivations are valid for both discrete and continuous random variables (summation is replaced by integration or vice versa). The stochastic sum becomes a combination of random variables in series and parallel as shown in Fig. 3.3 and dealt with in Sec. 3.2. For a given branch i we find (Fig. 3.3):

$$m_{1,i} = i \cdot m_{1,t} \,, \tag{3.43}$$

$$\sigma_i^2 = i \cdot \sigma_t^2 \,, \tag{3.44}$$

$$m_{2,i} = i \cdot \sigma_t^2 + (i \cdot m_{1,t})^2 \,. \tag{3.45}$$

By summation over all possible values (branches) i we get:

$$\begin{aligned} m_{1,s} &= \sum_{i=1}^{\infty} p(i) \cdot m_{1,i} \\ &= \sum_{i=1}^{\infty} p(i) \cdot i \cdot m_{1,t} \,, \end{aligned}$$

$$m_{1,s} = m_{1,t} \cdot m_{1,n} \,, \tag{3.46}$$

$$
\begin{aligned}
m_{2,s} &= \sum_{i=1}^{\infty} p(i) \cdot m_{2,i} \\
&= \sum_{i=1}^{\infty} p(i) \cdot \{ i \cdot \sigma_t^2 + (i \cdot m_{1,t})^2 \} \,, \\
m_{2,s} &= m_{1,n} \cdot \sigma_t^2 + m_{1,t}^2 \cdot m_{2,n} \,, \qquad (3.47) \\
\sigma_s^2 &= m_{1,n} \cdot \sigma_t^2 + m_{1,t}^2 \cdot (m_{2,n} - m_{1,n}^2) \,, \\
\sigma_s^2 &= m_{1,n} \cdot \sigma_t^2 + m_{1,t}^2 \cdot \sigma_n^2 \,. \qquad (3.48)
\end{aligned}
$$

We notice there are two contributions to the total variance: one term because the number of calls is a random variable (σ_n^2), and a term because the duration of the calls is a random variable (σ_t^2).

Example 3.3.1: Special case 1: $N = n =$ constant ($m_n = n$)

$$
\begin{aligned}
m_{1,s} &= n \cdot m_{1,t} \,, \\
\sigma_s^2 &= \sigma_t^2 \cdot n \,. \qquad (3.49)
\end{aligned}
$$

This corresponds to counting the number of calls at the same time as we measure the traffic volume so that we can estimate the mean holding time. □

Example 3.3.2: Special case 2: $T = t =$ constant ($m_t = t$)

$$
\begin{aligned}
m_{1,s} &= m_{1,n} \cdot t \,, \\
\sigma_s^2 &= t^2 \cdot \sigma_n^2 \,. \qquad (3.50)
\end{aligned}
$$

If we change the scale from 1 to $m_{1,t}$, then the mean value has to be multiplied by $m_{1,t}$ and the variance by $m_{1,t}^2$. The mean value $m_{1,t} = 1$ corresponds to counting the number of calls, i.e. a problem of counting. □

Example 3.3.3: Stochastic sum
As a non-teletraffic example N may denote the number of rain showers during one month and T_i may denote the precipitation due to the i'th shower. S_T is then a random variable describing the total precipitation during a month. N may also for a given time interval denote the number of accidents registered by an insurance company and T_i denotes the compensation for the i'th accident. S_T then is the total amount paid by the company for the considered period. □

Chapter 4

Time Interval Distributions

The exponential distribution is the most important time distribution within teletraffic theory. This time distribution is dealt with in Sec. 4.1.

Combining exponential distributed time intervals in series, we get a class of distributions called Erlang distributions (Sec. 4.2). Combining them in parallel, we obtain hyper–exponential distribution (Sec. 4.3). Combining exponential distributions both in series and in parallel, possibly with feedback, we obtain phase-type distributions, which is a class of general distributions. One important sub–class of phase-type distributions is Coxian-distributions (Sec. 4.4). We note that an arbitrary distribution can be expressed by a Cox–distribution which can be used in analytical models in a relatively simple way. Finally, we also deal with other time distributions which are employed in teletraffic theory (Sec. 4.5). Some examples of observations of life times are presented in Sec. 4.6.

4.1 Exponential distribution

In teletraffic theory this distribution is also called *the negative exponential distribution*. It has already been mentioned in Sec. 3.1.2 and it will appear again in Sec. 6.2.1.

In principle, we may use any distribution function with non–negative values to model a life–time. However, the exponential distribution has some unique characteristics which make this distribution qualified for both analytical and practical uses. The exponential distribution plays a key role among all life–time distributions.

This distribution is characterised by a single parameter, the *intensity* or *rate* λ:

$$F(t) = 1 - \mathrm{e}^{-\lambda t}, \qquad \lambda > 0, \quad t \geq 0, \tag{4.1}$$

$$f(t) = \lambda \mathrm{e}^{-\lambda t}, \qquad \lambda > 0, \quad t \geq 0. \tag{4.2}$$

The gamma function is defined by:

$$\Gamma(n+1) = \int_0^\infty t^n \, \mathrm{e}^{-t} \, \mathrm{d}t = n! \,. \tag{4.3}$$

We replace t by λt and get the ν'th moment:

$$m_\nu = \frac{\nu!}{\lambda^\nu}, \tag{4.4}$$

$$\begin{aligned}
\text{Mean value } m = \quad m_1 &= \frac{1}{\lambda}, \\
\text{Second moment:} \quad m_2 &= \frac{2}{\lambda^2}, \\
\text{Variance:} \quad \sigma^2 &= \frac{1}{\lambda^2}, \\
\text{Form factor:} \quad \varepsilon &= 2,
\end{aligned}$$

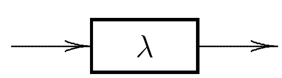


Figure 4.1: *In phase diagrams an exponentially distributed time interval is shown as a box with the intensity. The box thus means that a customer arriving to the box is delayed an exponentially distributed time interval before leaving the box.*

The exponential distribution is very suitable for describing physical time intervals (Fig. 6.2). The most fundamental characteristic of the exponential distribution is its *lack of memory*. The distribution of the residual time of a telephone conversation is independent of the actual duration of the conversation, and it is equal to the distribution of the total life-time (3.11):

$$\begin{aligned}
f(t+x|x) &= \frac{\lambda \mathrm{e}^{-(t+x)\lambda}}{\mathrm{e}^{-\lambda x}} \\
&= \lambda \mathrm{e}^{-\lambda t} \\
&= f(t).
\end{aligned}$$

If we remove the probability mass of the interval $(0, x)$ from the density function and normalise the residual mass in (x, ∞) to unity, then the new density function becomes congruent with the original density function. The only continuous distribution function having this property is the exponential distribution, whereas the geometric distribution is the only discrete distribution having this property. An example with the Weibull distribution where this property is not valid is shown in Fig. 3.1. For $k = 1$ the Weibull distribution becomes identical with the exponential distribution. Therefore, the mean value of the residual life-time is $m_{1,r} = m$, and the probability of observing a life–time in the interval $(t, t + \mathrm{d}t)$, given that it occurs after t, is given by

$$\begin{aligned} p\{t < X \le t + \mathrm{d}t | X > t\} &= \frac{f(t)\,\mathrm{d}t}{1 - F(t)} \\ &= \lambda\,\mathrm{d}t\,. \end{aligned} \tag{4.5}$$

Thus it depends only upon λ and $\mathrm{d}t$, but it is independent of the actual age t.

4.1.1 Minimum of k exponentially distributed random variables

We assume that two random variables X_1 and X_2 are mutually independent and exponentially distributed with intensities λ_1 and λ_2, respectively. A new random variable X is defined as:

$$X = \min\{X_1, X_2\}\ .$$

The distribution function of X is:

$$p\{X \le t\} = 1 - \mathrm{e}^{-(\lambda_1+\lambda_2)t}\,. \tag{4.6}$$

This distribution function itself is also an exponential distribution with intensity $(\lambda_1 + \lambda_2)$.

Under the assumption that the first (smallest) event happens within the time interval $t, t+\mathrm{d}t$, then the probability that the random variable X_1 is realized first (i.e. takes places in this interval and the other takes place later) is given by:

$$\begin{aligned} p\{X_1 < X_2 |\, t\} &= \frac{\lambda_1\,\mathrm{e}^{-\lambda_1 t}\,\mathrm{d}t \cdot \mathrm{e}^{-\lambda_2 t}}{(\lambda_1+\lambda_2)\,\mathrm{e}^{-(\lambda_1+\lambda_2)t}\,\mathrm{d}t} \\ &= \frac{\lambda_1}{\lambda_1+\lambda_2}\,, \end{aligned} \tag{4.7}$$

i.e. independent of t. Thus we do not need to integrate over all values of t.

These results can be generalised to k variables and make up the basic principle of the simulation technique called the *roulette method*, a Monte Carlo simulation methodology.

4.1.2 Combination of exponential distributions

If one exponential distribution (i.e. one parameter) cannot describe the time intervals in sufficient detail, then we may have to use a combination of two or more exponential distributions. Conny Palm introduced two classes of distributions: steep and flat.

A steep distribution corresponds to a set of stochastic independent exponential distributions in series (Fig. 4.2), and a flat distribution corresponds to exponential distributions in parallel (Fig. 4.4). This structure naturally corresponds to the shaping of traffic processes in telecommunication and data networks.

By the combination of steep and flat distribution, we may obtain an arbitrary good approximation for any distribution function (see Fig. 4.7 and Sec. 4.4). The diagrams in Figs. 4.2 & 4.4 are called phase-diagrams.

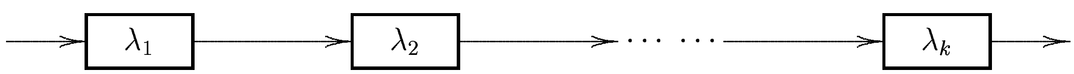


Figure 4.2: *By combining k exponential distributions in series we get a steep distribution ($\varepsilon \leq 2$). If all k distributions are identical ($\lambda_i = \lambda$), then we get an Erlang–k distribution.*

4.2 Steep distributions

Steep distributions are also called hypo–exponential distributions or generalised Erlang distributions with a form factor in the interval $1 < \varepsilon \leq 2$. This generalised distribution function is obtained by convolving k exponential distributions (Fig. 4.2). Here we only consider the case where all k exponential distributions are identical. Then we obtain the following density function which is called the *Erlang-k distribution:*

$$f(t) = \frac{(\lambda t)^{k-1}}{(k-1)!} \cdot \lambda \cdot \mathrm{e}^{-\lambda t}, \quad \lambda > 0, \quad t \geq 0, \quad k = 1, 2, \ldots\,. \tag{4.8}$$

$$F(t) = \sum_{j=k}^{\infty} \frac{(\lambda t)^j}{j!} \cdot \mathrm{e}^{-\lambda t} \tag{4.9}$$

$$= 1 - \sum_{j=0}^{k-1} \frac{(\lambda t)^j}{j!} \cdot \mathrm{e}^{-\lambda t} \qquad \text{(cf. Sec. 6.1)}\,. \tag{4.10}$$

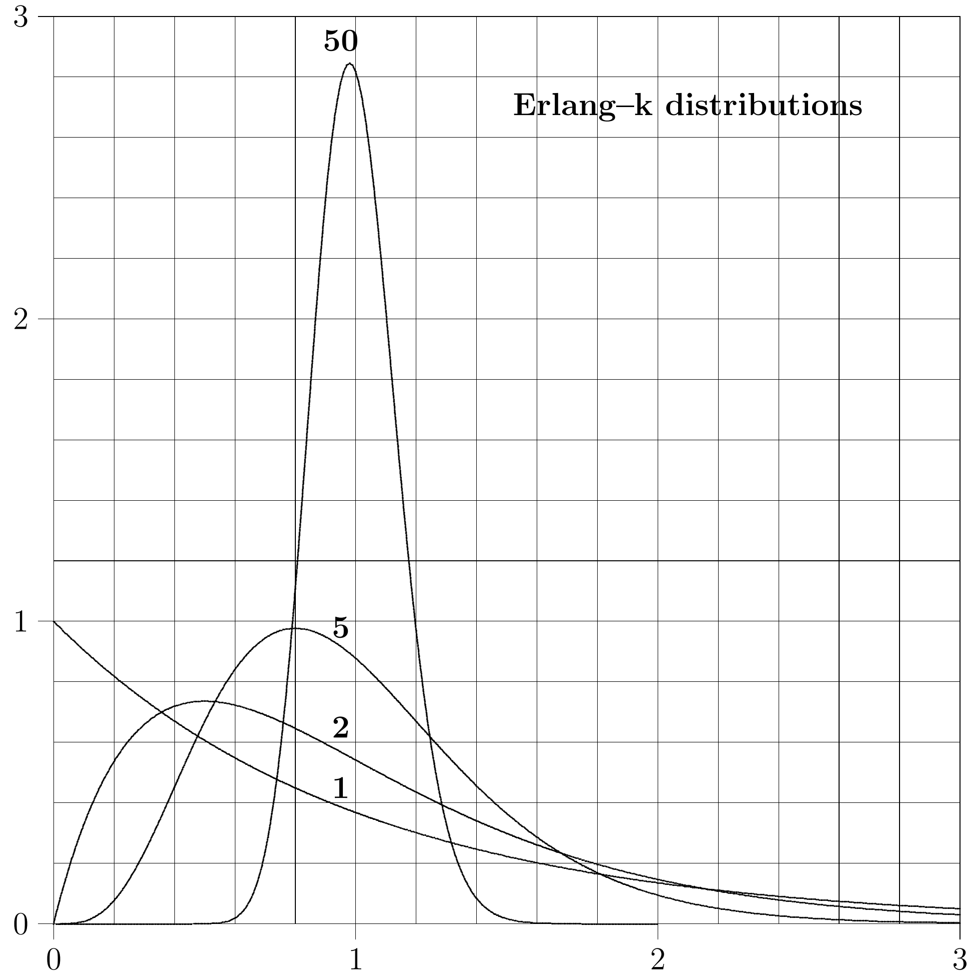


Figure 4.3: *Erlang–k distributions with mean value equal to one. The case $k = 1$ corresponds to an exponential distribution (density functions).*

The following moments can be found by using (3.31) and (3.32):

$$m = \frac{k}{\lambda}\,, \tag{4.11}$$

$$\sigma^2 = \frac{k}{\lambda^2}\,, \tag{4.12}$$

$$\varepsilon = 1 + \frac{\sigma^2}{m^2} = 1 + \frac{1}{k}\,, \tag{4.13}$$

The i'th non-central moment is:

$$m_i = \frac{(i+k-1)!}{(k-1)!} \cdot \left(\frac{1}{\lambda}\right)^i . \tag{4.14}$$

The density function is derived in Sec. 6.2.2. The mean residual life–time $m_{1,r}(x)$ for $x \geq 0$

will be less than the mean value:

$$m_{1,r}(x) \leq m, \quad x \geq 0\,.$$

With this distribution we have two parameters (λ, k) available to be estimated from observations. The mean value is often kept fixed. To study the influence of the parameter k in the distribution function, we normalise all Erlang–k distributions to the same mean value as the Erlang–1 distribution, i.e. the exponential distribution with mean $1/\lambda$, by replacing t by kt or λ by $k\lambda$:

$$f(t)\,\mathrm{d}t = \frac{(\lambda k t)^{k-1}}{(k-1)!}\,\mathrm{e}^{-\lambda k t}\,k\lambda\,\mathrm{d}t\,, \tag{4.15}$$

$$m = \frac{1}{\lambda}\,, \tag{4.16}$$

$$\sigma^2 = \frac{1}{k\lambda^2}\,, \tag{4.17}$$

$$\varepsilon = 1 + \frac{1}{k}\,. \tag{4.18}$$

Notice that the form factor is independent of time scale. The density function (4.15) is illustrated in Fig. 4.3 for different values of k with $\lambda = 1$. The case $k = 1$ corresponds to the exponential distribution. When $k \rightarrow \infty$ we get a constant time interval ($\varepsilon = 1$). By solving $f'(t) = 0$ we find the maximum value at:

$$\lambda t = \frac{k-1}{k}\,. \tag{4.19}$$

The so-called steep distributions are named so because the distribution functions increase quicker from 0 to 1 than the exponential distribution do. In teletraffic theory we sometimes use the name Erlang–distribution for the truncated Poisson distribution (Sec. 7.3).

4.3 Flat distributions

The general distribution function is in this case a weighted sum of exponential distributions (compound distribution) with a form factor $\varepsilon \geq 2$:

$$F(t) = \int_0^\infty \left(1 - \mathrm{e}^{-\lambda t}\right) \mathrm{d}W(\lambda)\,, \qquad \lambda > 0\,, \quad t \geq 0\,, \tag{4.20}$$

where the weight function may be discrete or continuous (Stieltjes integral). This distribution class corresponds to a parallel combination of the exponential distributions (Fig. 4.4). The density function is called complete monotone due to the alternating signs (Palm, 1957 [82]):

$$(-1)^\nu \cdot f^{(\nu)}(t) \geq 0\,. \tag{4.21}$$

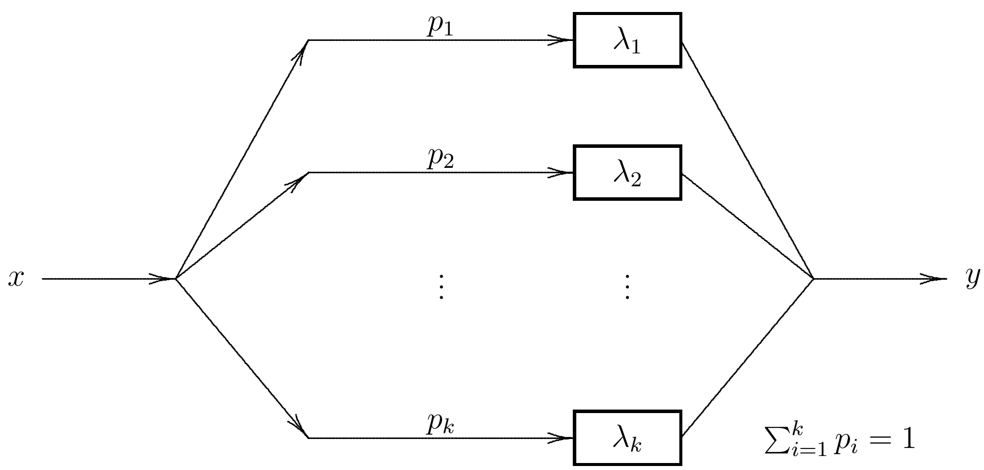


Figure 4.4: *By combining k exponential distributions in parallel and choosing branch number i with the probability p_i, we get a hyper–exponential distribution, which is a flat distribution ($\varepsilon \geq 2$).*

The mean residual life-time $m_{1,r}(x)$ for all $x \geq 0$ is larger than the mean value:

$$m_{1,r}(x) \geq m, \qquad x \geq 0\,. \tag{4.22}$$

4.3.1 Hyper-exponential distribution

In this case, $W(\lambda)$ is discrete. Suppose we have the following given values:

$$\lambda_1,\ \lambda_2,\ \ldots\ ,\ \lambda_k\,,$$

and that $W(\lambda)$ has the positive increases:

$$p_1,\ p_2,\ \ldots\ ,\ p_k\,,$$

where

$$\sum_{i=1}^{k} p_i = 1\,. \tag{4.23}$$

For all other values $W(\lambda)$ is constant. In this case (4.20) becomes:

$$F(t) = 1 - \sum_{i=1}^{k} p_i \cdot \mathrm{e}^{-\lambda_i t}\,, \quad t \geq 0\,. \tag{4.24}$$

The mean values and form factor may be found from (3.36) and (3.37) ($\sigma_i = m_{1,i} = 1/\lambda_i$):

$$m_1 = \sum_{i=1}^{k} \frac{p_i}{\lambda_i}, \tag{4.25}$$

$$\varepsilon = 2\left\{\sum_{i=1}^{k} \frac{p_i}{\lambda_i^2}\right\} \Bigg/ \left\{\sum_{i=1}^{k} \frac{p_i}{\lambda_i}\right\}^2 \geq 2\,. \tag{4.26}$$

If $k = 1$ or all λ_i are equal, we get the exponential distribution.

This class of distributions is called hyper–exponential distributions and can be obtained by combining k exponential distributions in parallel, where the probability of choosing the i'th distribution is given by p_i. The distribution is called flat because its distribution function increases more slowly from 0 to 1 than the exponential distribution does.

In practice, it is difficult to estimate more than one or two parameters. The most important case is for $n = 2$ ($p_1 = p, p_2 = 1 - p$):

$$F(t) = 1 - p \cdot \mathrm{e}^{-\lambda_1 t} - (1-p) \cdot \mathrm{e}^{-\lambda_2 t}\,. \tag{4.27}$$

Statistical problems arise even when we deal with three parameters. So for practical applications we usually choose $\lambda_i = 2\lambda p_i$ and thus reduce the number of parameters to only two:

$$F(t) = 1 - p\mathrm{e}^{-2\lambda p t} - (1-p)\mathrm{e}^{-2\lambda(1-p)t}\,. \tag{4.28}$$

The mean value and form factor becomes:

$$m = \frac{1}{\lambda},$$

$$\varepsilon = \frac{1}{2p(1-p)}\,. \tag{4.29}$$

For this choice of parameters the two branches have the same contribution to the mean value. Fig. 4.5 illustrates an example.

4.4 Cox distributions

By combining the steep and flat distributions we obtain a general class of distributions (phase–type distributions) which can be described with exponential phase in both series and parallel (e.g. a $k \times \ell$ matrix). To analyse a model with this kind of distributions, we can apply the

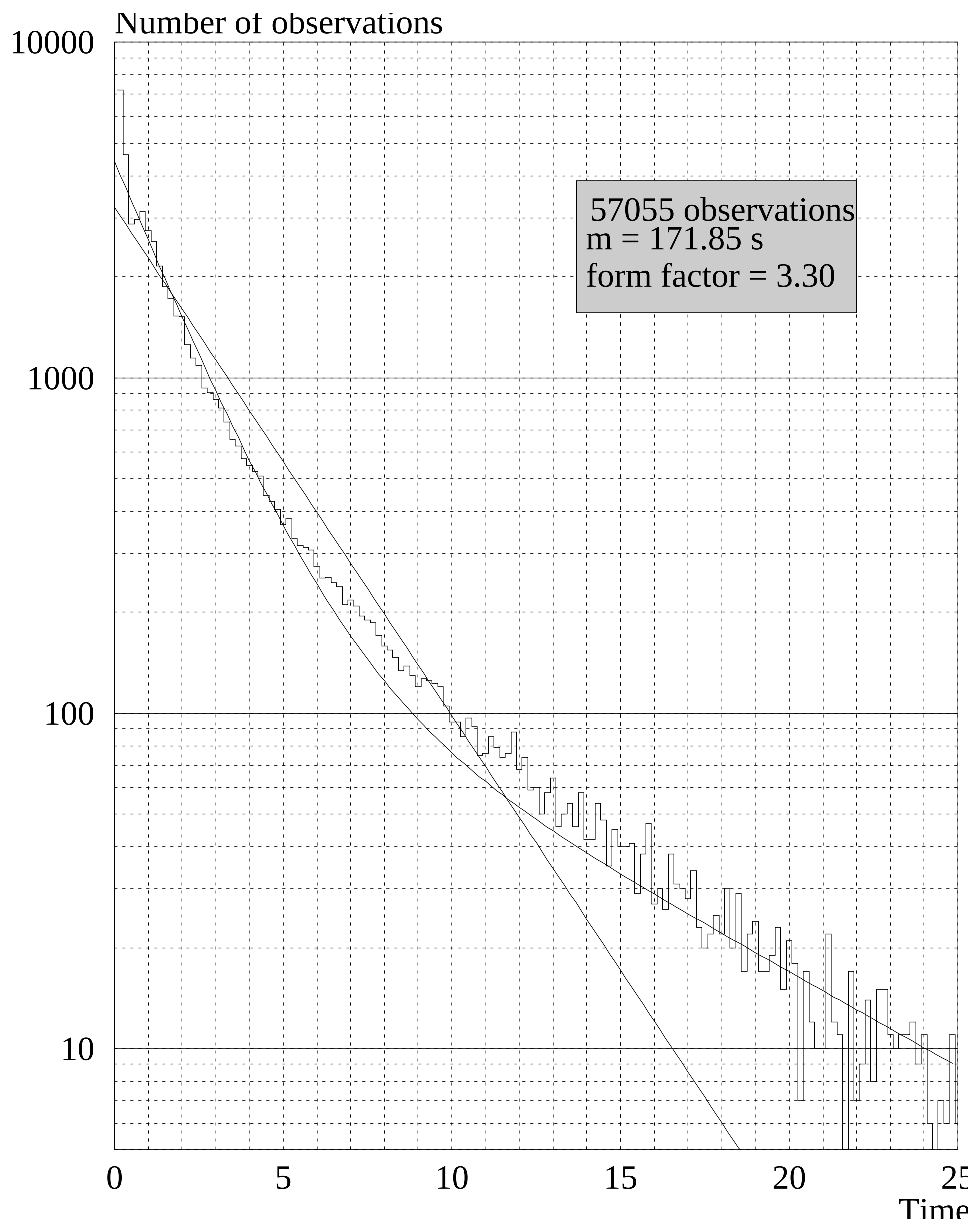


Figure 4.5: *Density (frequency) function for holding times observed on lines in a local exchange during busy hours.*

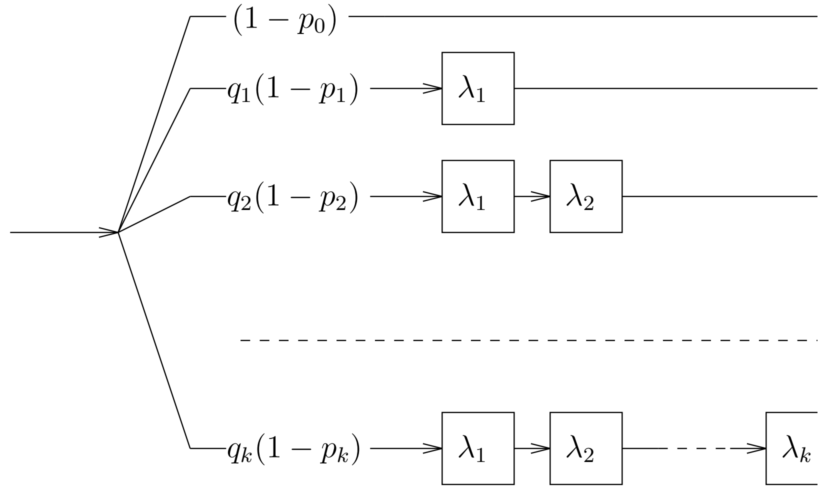


Figure 4.6: *A Cox–distribution is a generalised Erlang–distribution having exponential distributions in both parallel and series. The phase-diagram is equivalent to Fig. 4.7.*

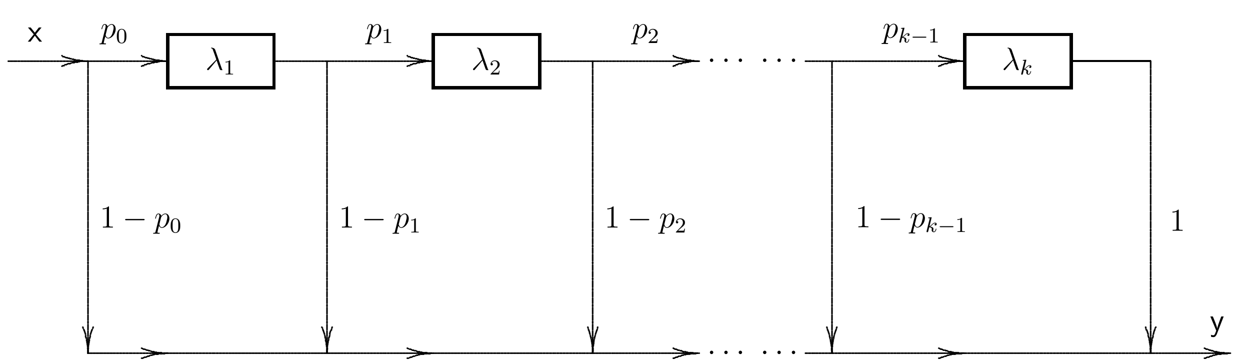


Figure 4.7: *The phase diagram of a Cox distribution, cf. Fig. 4.6.*

theory of Markov processes, for which we have powerful tools as the phase-method. In the more general case we can allow for loop back between the phases.

We shall only consider *Cox-distributions* as shown in Fig. 4.6 (Cox, 1955 [17]). These also appear under the name of "Branching Erlang" distribution (Erlang distribution with branches).

The mean value and variance of this Cox distribution (Fig. 4.7) are found from the formulae in Sec. 3.2 for random variables in series and parallel as shown in fig. 4.6:

$$m_1 = \sum_{i=1}^{k} q_i \, (1 - p_i) \left\{ \sum_{j=1}^{i} \frac{1}{\lambda_j} \right\} , \tag{4.30}$$

where

$$q_i = p_0 \cdot p_1 \cdot p_2 \cdot \cdots \cdot p_{i-1} \, . \tag{4.31}$$

The term $q_i(1 - p_i)$ is the probability of jumping out after being in i'th phase. It can be shown thet the mean value can be expressed by the simple form:

$$m_1 = \sum_{i=1}^{k} \frac{q_i}{\lambda_i} = \sum_{i=1}^{k} m_{1,i} \, , \tag{4.32}$$

where $m_{1,i} = q_i/\lambda_i$ is the i'th phase related mean value. The second moment becomes:

$$\begin{aligned} m_2 &= \sum_{i=1}^{k} \left\{ q_i \, (1 - p_i) \cdot m_{2,i} \right\} , \\ &= \sum_{i=1}^{k} \left\{ q_i \; (1 - p_i) \cdot \left\{ \sum_{j=1}^{i} \frac{1}{\lambda_j^2} + \left(\sum_{j=1}^{i} \frac{1}{\lambda_j} \right)^2 \right\} \right\} , \end{aligned} \tag{4.33}$$

where $m_{2,i}$ is obtained from (3.8): $m_{2,i} = \sigma_{2,i}^2 + m_{1,i}^2$. It can be shown that this can be written as:

$$m_2 = 2 \cdot \sum_{i=1}^{k} \left\{ \left(\sum_{j=1}^{i} \frac{1}{\lambda_j} \right) \cdot \frac{q_i}{\lambda_i} \right\} . \tag{4.34}$$

From this we get the variance (3.8):

$$\sigma^2 = m_2 - m_1^2 \, .$$

The addition of two Cox–distributed random variables yields another Cox-distributed variable, i.e. this class is closed under the operation of addition.

The distribution function of a Cox distribution can be written as a sum of exponential functions:

$$1 - F(t) = \sum_{i=1}^{k} c_i \cdot \mathrm{e}^{-\lambda_i t} \, , \tag{4.35}$$

where

$$0 \leq \sum_{i=1}^{k} c_i \leq 1\,,$$

and

$$-\infty < c_i < +\infty\,.$$

4.4.1 Polynomial trial

The following properties are of importance for later applications. If we consider a point of time chosen at random within a Cox–distributed time interval, then the probability that this point is within phase i is given by:

$$\frac{m_i}{m}\,, \qquad i = 1, 2, \ldots, k\,. \tag{4.36}$$

If we repeat this experiment y (independently) times, then the probability that phase i is observed y_i times is given by *multinomial distribution* (= polynomial distribution):

$$p\{y \,|\, y_1, y_2, \ldots, y_k\} = \binom{y}{y_1\, y_2\, \ldots\, y_k} \cdot \left(\frac{m_{1,1}}{m}\right)^{y_1} \cdot \left(\frac{m_{1,2}}{m}\right)^{y_2} \cdot \cdots \cdot \left(\frac{m_{1,k}}{m}\right)^{y_k}\,, \tag{4.37}$$

where

$$y = \sum_{i=1}^{k} y_i\,,$$

and

$$\binom{y}{y_1\, y_2\, \ldots\, y_k} = \frac{y!}{y_1! \cdot y_2! \cdot \cdots \cdot y_k!}\,. \tag{4.38}$$

These (4.38) are called the multinomial coefficients. By the property of "lack of memory" of the exponential distributions (phases) we have full information about the residual life-time, when we know the number of the actual phase.

4.4.2 Decomposition principles

Phase–diagrams are a useful tool for analysing Cox distributions. The following is a fundamental characteristic of the exponential distribution (Iversen & Nielsen, 1985 [41]):

Theorem 4.1 *An exponential distribution with intensity λ can be decomposed into a two-phase Cox distribution, where the first phase has an intensity $\mu > \lambda$ and the second phase intensity λ (cf. Fig. 4.8).*

According to Theorem 4.1 a hyper–exponential distribution with ℓ phases is equivalent to a Cox distribution with the same number of phases. The case $\ell = 2$ is shown in Fig. 4.10.

We have another property of Cox distributions (Iversen & Nielsen, 1985 [41]):

Theorem 4.2 *The phases in any Cox distribution can be ordered such as $\lambda_i \geq \lambda_{i+1}$.*

Theorem 4.1 shows that an exponential distribution is equivalent to a homogeneous Cox distribution (homogeneous: same intensities in all phases) with intensity m and an infinite number of phases (Fig. 4.8). We notice that the branching probabilities are constant. Fig. 4.9 corresponds to a weighted sum of Erlang–k distributions where the weighting factors are geometrically distributed.

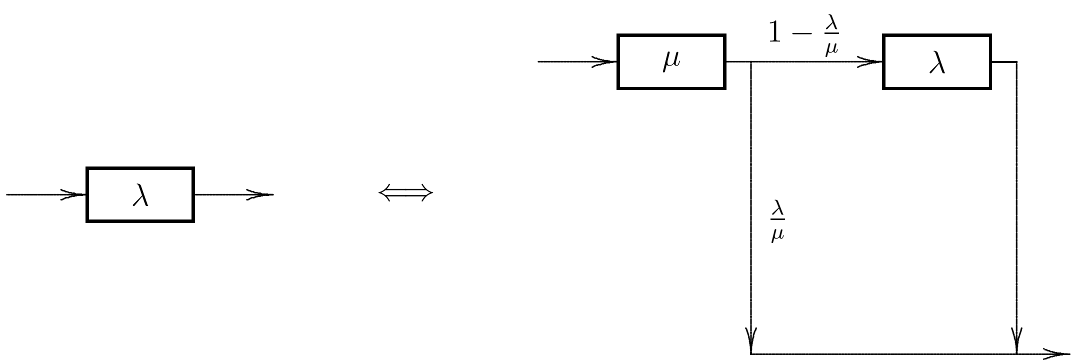


Figure 4.8: *An exponential distribution with rate λ is equivalent to the shown Cox distribution (Theorem 4.1).*

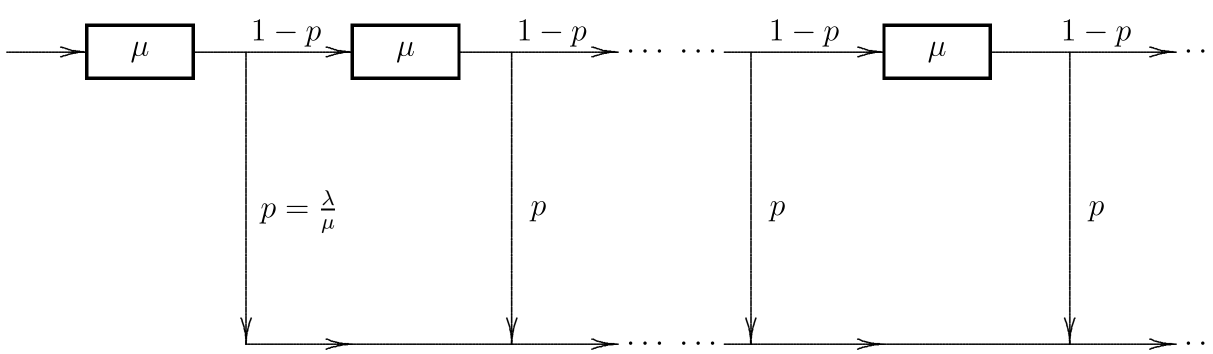


Figure 4.9: *An exponential distribution with rate λ is by successive decomposition transformed into a compound distribution of homogeneous Erlang–k distributions with rates $\mu > \lambda$, where the weighting factors follows a geometric distribution (quotient $p = \lambda/\mu$).*

By using phase diagrams it is easy to see that any exponential time interval (λ) can be decomposed into phase-type distributions (λ_i), where $\lambda_i \geq \lambda$. Referring to Fig. 4.11 we

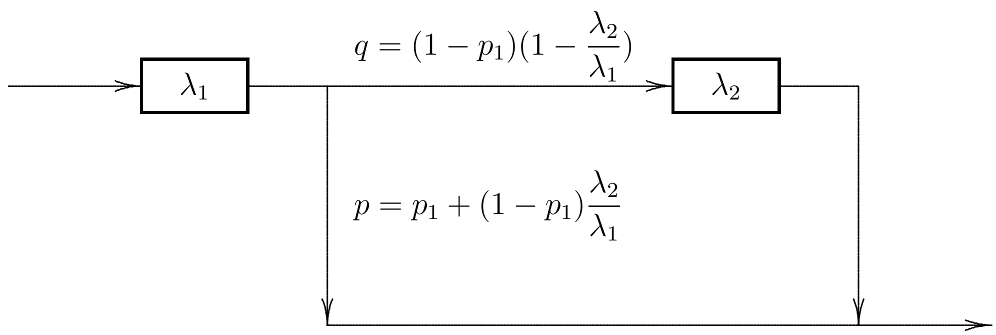


Figure 4.10: *A hyper–exponential distribution with two phases ($\lambda_1 > \lambda_2$, $p_2 = 1 - p_1$) can be transformed into a Cox–2 distribution (cf. Fig. 4.4).*

notice that the rate out of the macro-state (dashed box) is λ independent of the micro state. When the number of phases k is finite and there is no feedback the final phase must have rate λ.

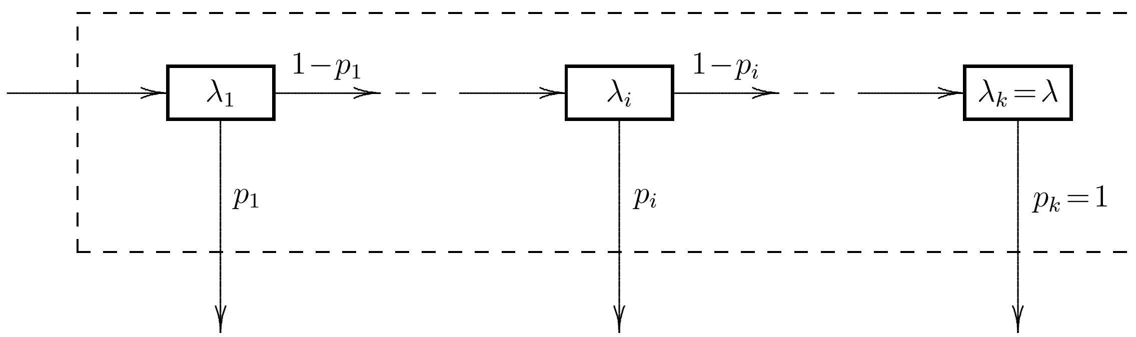


Figure 4.11: *This phase-type distribution is equivalent to a single exponential when $p_i \cdot \lambda_i = \lambda$. Thus $\lambda_i \geq \lambda$ as $0 < p_i \leq 1$.*

4.4.3 Importance of Cox distribution

Cox distributions have attracted a lot of attention during recent years. They are of great importance due to the following properties:

a. Cox distribution can be analysed using the method of phases.

b. One can approximate an arbitrary distribution arbitrarily well with a Cox distribution. If a property is valid for a Cox distribution, then it is valid for any distribution of practical interest.

By using Cox distributions we can with elementary methods obtain results which previously required very advanced mathematics.

In the connection with practical applications of the theory, we have used the methods to estimate the parameters of Cox distribution. In general there are $2\,k$ parameters in an unsolved statistical problem. Normally, we may choose a special Cox distribution (e.g. Erlang–k or hyper–exponential distribution) and approximate the first moment.

By numerical simulation on computers using the *Roulette method*, we automatically obtain the observations of the time intervals as Cox distribution with the same intensities in all phases.

4.5 Other time distributions

In principle, every distribution which has non–negative values, may be used as a time distribution to describe the time intervals. But in practice, one may work primarily with the above mentioned distributions.

We suppose the parameter k in Erlang-k distribution (4.8) takes non-negative real values and obtain the *gamma distribution*:

$$f(t) = \frac{1}{\Gamma(k)}(\lambda t)^{k-1} \cdot \mathrm{e}^{-\lambda t} \cdot \lambda\,, \qquad \lambda > 0\,, \quad t \geq 0\,. \tag{4.39}$$

The mean value and variance are given in (4.11) and (4.12).

A distribution also known in teletraffic theory is the *Weibull distribution*:

$$F(t) = 1 - \mathrm{e}^{-(\lambda t)^k}\,, \qquad t \geq 0\,, \quad k > 0\,, \quad \lambda > 0\,. \tag{4.40}$$

This distribution has a time-dependent death intensity (3.14):

$$\begin{aligned} \frac{\mathrm{d}F(t)}{1-F(t)} = \mu(t) \quad &= \quad \frac{\lambda \mathrm{e}^{-(\lambda t)^k} \cdot k\,(\lambda t)^{k-1}\,\mathrm{d}t}{\mathrm{e}^{-(\lambda t)^k}} \\ &= \quad \lambda k\,(\lambda t)^{k-1}\,. \end{aligned} \tag{4.41}$$

The distribution has its origin in the reliability theory. For $k = 1$ we get the exponential distribution.

The Pareto distribution is given by:

$$F(t) = 1 - (1+\eta_0 t)^{-\left(1+\frac{\lambda}{\eta_0}\right)}\,. \tag{4.42}$$

The mean value and form factor are as follows:

$$\begin{aligned} m_1 &= \frac{1}{\lambda}, \\ \varepsilon &= \frac{2\lambda}{\lambda - \eta_0}, \qquad 0 < \eta_0 < \lambda . \end{aligned} \tag{4.43}$$

Note that the variance does not exist for $\lambda \leq \eta_0$. Letting $\eta_0 \rightarrow 0$ (4.42) becomes an exponential distribution. If the intensity of a Poisson process is gamma distributed, then the inter-arrival times are Pareto-distributed.

Later, we will deal with a set of discrete distributions, which also describes the life–time, such as geometrical distribution, Pascal distribution, Binomial distribution, Westerberg distribution, etc. In practice, the parameters of distributions are not always stationary.

The service (holding) times can be physically correlated with the state of the system. In man–machine systems the service time changes because of busyness (decreases) or tiredness (increases). In the same way, electro–mechanical systems work more slowly during periods of high load because the voltage decreases.

For some distributions which are widely applied in the queueing theory, we have the following abbreviated notations (cf. Sec. 13.1):

M	$\sim$	Exponential distribution (Markov),
E_k	$\sim$	Erlang-k distribution,
H_n	$\sim$	Hyper-exponential distribution of order n,
D	$\sim$	Constant (Deterministic),
Cox	$\sim$	Cox distribution,
G	$\sim$	General = arbitrary distribution.

4.6 Observations of life-time distribution

Fig. 4.5 shows an example of observed holding times from a local telephone exchange. The holding time consists of both signalling time and, if the call is answered, conversation time. Fig. 6.2 shows observation and inter–arrival times of incoming calls to a transit telephone exchange during one hour.

From its very beginning, the teletraffic theory has been characterised by a strong interaction between theory and practice, and there has been excellent possibilities to carry out measurements.

Erlang (1920, [11]) reports a measurement where 2461 conversation times were recorded in a telephone exchange in Copenhagen in 1916. Palm (1943 [79]) analysed the field of traffic measurements, both theoretically and practically, and implemented extensive measurements in Sweden.

By the use of computer technology a large amount of data can be collected. The first stored program controlled by a mini-computer measurement is described in (Iversen, 1973 [35]). The importance of using discrete values of time when observing values is dealt with in Chapter 15. Bolotin (1994, [7]) has measured and modelled telecommunication holding times.

Numerous measurements on computer systems have been carried out. Where in telephone systems we seldom have a form factor greater than 6, we observe form factors greater than 100 in data traffic. This is the case for example for data transmission, where we send either a few characters or a large quantity of data. To describe these data we use *heavy-tailed distributions*. A distribution is heavy-tailed in strict sense if the tail of the distribution function behaves as a power law, i.e. as

$$1 - F(t) \approx t^{-\alpha}\,,\ 0 < \alpha \leq 2\,.$$

The Pareto distribution (4.42) is heavy-tailed in strict sense. Sometimes distributions with a tail heavier than the exponential distribution are classified as heavy-tailed. Examples are hyper-exponential, Weibull, and log-normal distributions. More recent extensive measurements have been performed and modelled using self-similar traffic models (Jerkins & al., 1999 [50]). These subjects are dealt with in more advanced chapters.

Chapter 5

Arrival Processes

Arrival processes, such as telephone calls arriving to an exchange are described mathematically as *stochastic point processes*. For a point process, we have to be able to distinguish two arrivals from each other. Informations concerning the single arrival (e.g. service time, number of customers) are ignored. Such information can only be used to determine whether an arrival belongs to the process or not.

The mathematical theory for point process was founded and developed by the Swede *Conny Palm* during the 1940'es. This theory has been widely applied in many subjects. It was mathematically refined by Khintchine ([62], 1968), and has been made widely applicable in many textbooks.

5.1 Description of point processes

In the following we only consider *simple* point processes, i.e. we exclude *multiple arrivals* as for example twin arrivals. For telephone calls this may be realized by a choosing sufficient detailed time scale.

Consider call arrival times where the i'th call arrives at time T_i:

$$0 = T_0 < T_1 < T_2 < \ldots < T_i < T_{i+1} < \ldots \,. \tag{5.1}$$

The first observation takes place at time $T_0 = 0$.

The number of calls in the half open interval $[0, t[$ is denoted as N_t. Here N_t is a random variable with continuous time parameters and discrete space. When t increases, N_t never decreases.

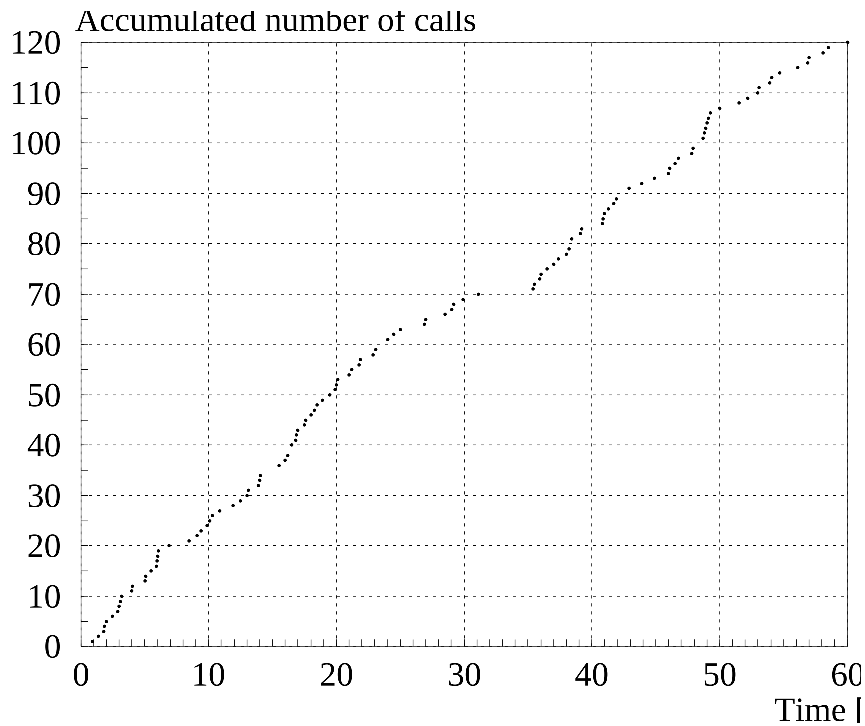


Figure 5.1: *The call arrival process at the incoming lines of a transit exchange.*

The time distance between two successive arrivals is:

$$X_i = T_i - T_{i-1}, \qquad i = 1, 2, \ldots \, . \tag{5.2}$$

This is called the *inter-arrival time*, and the distribution of this process is called the *inter-arrival time distribution*.

Corresponding to the two random variables N_t and X_i, the two processes can be characterised in two ways:

1. *Number representation* N_t: time interval t is kept constant, and we observe the random variable N_t for the number of calls in t.

2. *Interval representation* T_i: number of arriving calls is kept constant, and we observe the random variable T_i for the time interval until there has been n arrivals (especially $T_1 = X_1$).

The fundamental relationship between the two representations is given by the following simple relation:

$$\left.\begin{array}{rcll} N_t & < & n, & \text{if and only if} \\ T_n = \sum_{i=1}^{n} X_i & \geq & t, & n = 1, 2, \ldots \end{array}\right\} \tag{5.3}$$

This is expressed by *Feller-Jensen's identity:*

$$p\left\{N_t < n\right\} = p\left\{T_n \geq t\right\}, \qquad n = 1, 2, \ldots \tag{5.4}$$

Analysis of point process can be based on both of these representations. In principle they are equivalent. Interval representation corresponds to the usual time series analysis. If we for example let $i = 1$, we obtain *call averages*, i.e. statistics based on call arrivals.

Number representation has no parallel in time series analysis. The statistics we obtain are calculated per time unit and we get *time averages* (cf. the difference between call congestion and time congestion).

The statistics of interests when studying point processes can be classified according to the two representations.

5.1.1 Basic properties of number representation

There are two properties which are of theoretical interest:

1. *The total number of arrivals* in interval $[t_1, t_2[$ is equal to $N_{t_2} - N_{t_1}$.
   The average number of calls in the same interval is called the *renewal function* H:
$$H(t_1, t_2) = E\{N_{t_2} - N_{t_1}\}\ . \tag{5.5}$$

2. *The density* of arriving calls at time t (time average) is:
$$\lambda_t = \lim_{\Delta t \to 0} \frac{N_{t+\Delta t} - N_t}{\Delta t} = N'_t\,. \tag{5.6}$$
   We assume that λ_t exists and is finite. We may interpret λ_t as the *intensity* by which arrivals occur at time t (cf. Sec. 3.1.2).
   For *simple point processes*, we have:
$$\begin{aligned} p\{N_{t+\Delta t} - N_t \geq 2\} &= o(\Delta t)\,, && (5.7)\\ p\{N_{t+\Delta t} - N_t = 1\} &= \lambda_t \Delta t + o(\Delta t)\,, && (5.8)\\ p\{N_{t+\Delta t} - N_t = 0\} &= 1 - \lambda_t \Delta t + o(\Delta t)\,, && (5.9)\end{aligned}$$
   where by definition:
$$\lim_{\Delta t \to 0} \frac{o(\Delta t)}{\Delta t} = 0\,. \tag{5.10}$$

3. *Index of Dispersion for Counts, IDC.*
   To describe second order properties of the number representation we use the *index of dispersion for counts*, IDC. This describes the variations of the arrival process during a time interval t and is defined as:
$$IDC = \frac{Var\{N_t\}}{E\{N_t\}}\,. \tag{5.11}$$

By dividing the time interval t into x intervals of duration t/x and observing the number of events during these intervals we obtain an estimate of *IDC(t)*. For the Poisson process *IDC* becomes equal to one. *IDC* is equal to "peakedness", which we later introduce to characterise the number of busy channels in a traffic process (7.7).

5.1.2 Basic properties of interval representation

4. The distribution $f(t)$ of time intervals X_i (5.2) (and by convolving the distribution by itself $i-1$ times the distribution of the time until the i'th arrival).

$$F_i(t) = p\{X_i \leq t\} , \tag{5.12}$$

$$E\{X_i\} = m_{1,i} . \tag{5.13}$$

   The mean value is a call average. A *renewal process* is a point process, where sequential inter-arrival times are stochastic independent to each other and have the same distribution (except for X_1), i.e. $m_{1,i} = m_i$. (*IID* = *I*dentically and *I*ndependently *D*istributed).

5. The distribution $V(t)$ of the time interval from a random epoch until the first arrival occurs. The mean value of $V(t)$ is a time average, which is calculated per time unit.

6. *Index of Dispersion for Intervals, IDI.*
   To describe second order properties for the interval representation we use the Index of Dispersion for Intervals, *IDI*. This is defined as:

$$IDI = \frac{Var\{X_i\}}{E\{X_i\}^2} , \tag{5.14}$$

   where X_i is the inter-arrival time. For the Poisson process, which has exponentially distributed service times, *IDI* becomes equal to one. *IDI* is equal to Palm's form factor minus one (3.10). In general, *IDI* is more difficult to obtain from observations than *IDC*, and more sensitive to the accuracy of measurements and smoothing of the traffic process. The digital technology is more suitable for observation of *IDC*, whereas it complicates the observation of *IDI* (Chap. 15).

Which of the two representations one should use in practice really depends on the actual case. This can be illustrated by the following examples.

Example 5.1.1: Measuring principles
Measures of teletraffic performance are carried out by one of the two basic principles as follows:

1. *Passive measures*. Measuring equipment records at regular time intervals the number of arrivals since the last recording. This corresponds to the *scanning method*, which is suitable for computers. This corresponds to the number representation where the time interval is fixed.

2. *Active measures*. Measuring equipment records an event at the instant it takes place. We keep the number of events fixed and observe the measuring interval. Examples are recording instruments. This corresponds to the interval representation, where we obtain statistics for each single call.

□

Example 5.1.2: Test calls
Investigation of the *traffic* quality. In practice this is done in two ways:

1. The traffic quality is estimated by collecting statistics of the outcome of test calls made to specific (dummy-) subscribers. The calls are generated during busy hour independently of the actual traffic. The test equipment records the number of blocked calls etc. The obtained statistics corresponds to *time averages* of the performance measure. Unfortunately, this method increases the offered load on the system. Theoretically, the obtained performance measures will differ from the correct values.

2. The test equipments collect data from call number $N, 2N, 3N, \ldots$, where for example $N = 1000$. The traffic process is unchanged, and the performance statistics is a *call average*.

□

Example 5.1.3: Call statistics
A *subscriber* evaluates the quality by the fraction of calls which are blocked, i.e. call average.
The *operator* evaluates the quality by the proportion of time when all trunks are busy, i.e. time average. The two types of average values (time/call) are often mixed up, resulting in apparently conflicting statement. □

Example 5.1.4: Called party busy (B-Busy)
At a telephone exchange 10% of the subscribers are busy, but 20% of the call attempts are blocked due to B-busy (called party busy). This phenomenon can be explained by the fact that half of the subscribers are passive (i.e. make no call attempts and receive no calls), whereas 20% of the remaining subscribers are busy. G. Lind (1976 [73]) analysed the problem under the assumption that each subscriber on the average has the same number of incoming and outgoing calls. If mean value and form factor of the distribution of traffic per subscriber is b and ε, respectively, then the probability that a call attempts get B-busy is $b \cdot \varepsilon$. □

5.2 Characteristics of point process

Above we have discussed a very general structure for point processes. For specific applications we have to introduce further properties. Below we only consider *number representation*, but we could do the same based on the interval representation.

5.2.1 Stationarity (Time homogeneity)

This property can be described as, regardless of the position on the time axis, then the probability distributions describing the point process are independent of the instant of time. The following definition is useful in practice:

Definition: For an arbitrary $t_2 > 0$ and every $k \geq 0$, the probability that there are k arrivals in $[t_1, t_1 + t_2[$ is independent of t_1, i.e. for all t, k we have:

$$p\left\{N_{t_1+t_2} - N_{t_1} = k\right\} = p\left\{N_{t_1+t_2+t} - N_{t_1+t} = k\right\} . \tag{5.15}$$

There are many other definitions of stationarity, some stronger, some weaker.

Stationarity can also be defined by interval representation by requiring all X_i to be independent and identically distributed (*IID*). A weaker definition is that all first and second order moments (e.g. the mean value and variance) of a point process must be invariant with respect to time shifts. *Erlang* introduced the concept of *statistical equilibrium*, which requires that the derivatives of the process with respect to time are zero.

5.2.2 Independence

This property can be expressed as the requirement that the future evolution of the process only depends upon the present state.

Definition: The probability that k events (k is integer and ≥ 0) take place in $[t_1, t_1 + t_2[$ is independent of events before time t_1

$$p\left\{N_{t_2} - N_{t_1} = k | N_{t_1} - N_{t_0} = n\right\} = p\left\{N_{t_2} - N_{t_1} = k\right\} \tag{5.16}$$

If this holds for all t, then the process is a *Markov process:* the future evolution only depends on the present state, but is independent of how this has been obtained. This is the *lack of memory* property. If this property only holds for certain time points (e.g. arrival times), these points are called *equilibrium points* or *regeneration points*. The process then has a limited memory, and we only need to keep record of the past back the the latest regeneration point.

Example 5.2.1: Equilibrium points = regeneration points
Examples of point process with equilibrium points.

a) *Poisson process* is (as we will see in next chapter) memoryless, and all points of the time axes are equilibrium points.

b) *A scanning process*, where scannings occur at a regular cycle, has limited memory. The latest scanning instant has full information about the scanning process, and therefore all scanning points are equilibrium points.

c) If we superpose the above-mentioned Poisson process and scanning process (for instance by investigating the arrival processes in a computer system), the only equilibrium points in the compound process are the scanning instants.

d) Consider a queueing system with Poisson arrival process, constant service time and single server. The number of queueing positions can be finite or infinite. Let a point process be defined by the time instants when service starts. All time intervals when the system is idle, will be equilibrium points. During periods, where the system is busy, the time points for accept of new calls for service depends on the instant when the first call of the busy period started service.

□

5.2.3 Simple point process

We have already mentioned (5.7) that we exclude processes with multiple arrivals.

Definition: A point process is called simple, if the probability that there are more than one event at a given point is zero:

$$p\{N_{t+\Delta t} - N_t \geq 2\} = o(\Delta t)\,. \tag{5.17}$$

With interval representation, the inter-arrival time distribution must not have a probability mass (atom) at zero, i.e. the distribution is continuous at zero (3.1):

$$F(0+) = 0 \tag{5.18}$$

Example 5.2.2: Multiple events
Time points of traffic accidents will form a simple process. Number of damaged cars or dead people will be a non-simple point process with multiple events. □

5.3 Little's theorem

This is the only general result that is valid for all queueing systems. It was first published by Little (1961 [75]). The proof below was shown by applying the theory of stochastic process in (Eilon, 1969 [24]).

We consider a queueing system, where customers arrive according to a stochastic process. Customers enter the system at a random time and wait to get service, after being served they leave the system. In Fig. 5.2, both arrival and departure processes are considered as stochastic processes with cumulated number of customers as ordinate.

We consider a time space T and assume that the system is in *statistic equilibrium* at initial time $t = 0$. We use the following notation (Fig. 5.2):

$$
\begin{aligned}
N(T) &= \text{number of arrivals in period } T. \\
A(T) &= \text{the total service times of all customers in the period } T \\
&\quad = \text{the shadowed area between curves} \\
&\quad = \text{the carried traffic volume.} \\
\lambda(T) &= \tfrac{N(T)}{T} = \text{the average call intensity in the period } T. \\
W(T) &= \tfrac{A(T)}{N(T)} = \text{mean holding time in system per call in the period } T. \\
L(T) &= \tfrac{A(T)}{T} = \text{the average number of calls in the system in the period } T.
\end{aligned}
$$

We have the important relation among these variables:

$$L(T) = \frac{A(T)}{T} = \frac{W(T) \cdot N(T)}{T} = \lambda(T) \cdot W(T) \tag{5.19}$$

If the limits of $\lambda = \lim_{T\to\infty} \lambda(T)$ and $W = \lim_{T\to\infty} W(T)$ exist, then the limiting value of $L(T)$ also exists and it becomes:

$$L = \lambda \cdot W \qquad \text{(Little's theorem).} \tag{5.20}$$

This simple formula is valid for all general queueing system. The proof had been refined during the years. We shall use this formula in Chaps. 12–14.

Example 5.3.1: Little's formula
If we only consider the waiting positions, the formula shows:

The mean queue length is equal to call intensity multiplied by the mean waiting time.

If we only consider the servers, the formula shows:

The carried traffic is equal to arrival intensity multiplied by mean service time $(A = y \cdot s = \lambda/\mu)$.

This corresponds to the definition of offered traffic in Sec. 2.1. □

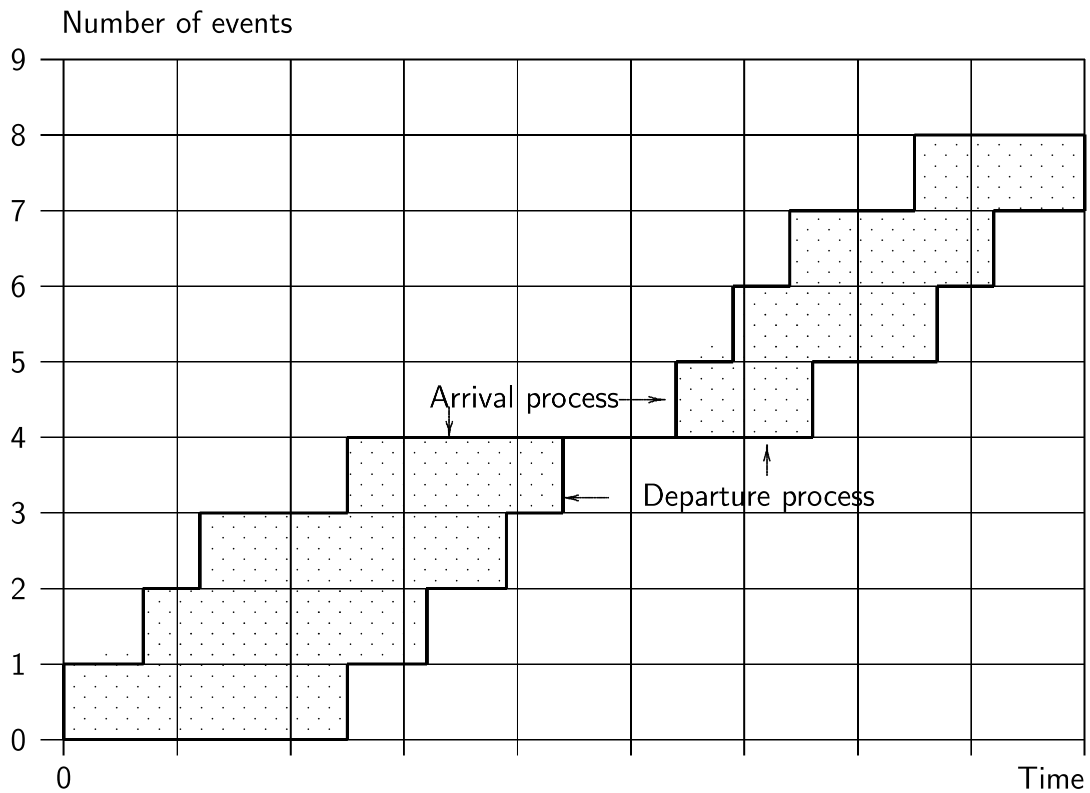


Figure 5.2: *A queueing system with arrival and departure of customers. The vertical distance between the two curves is equal to the actual number of customers being served. The customers in general don't depart in the the same order as they arrive, so the horizontal distance between the curves don't describe the actual time in the system of a customer.*

Chapter 6

The Poisson process

The Poisson process is the most important point process. Later we will realize that its role among point processes is as fundamental as the role of the Normal distribution among statistical distributions. By the central limit theorem we obtain the Normal distribution when adding random variables. In a similar way we obtain the exponential distribution when superposing stochastic point processes.

Most other applied point processes are generalisations or modifications of the Poisson process. This process gives a surprisingly good description of many real–life processes. This is because it is the most random process. The more complex a process is, the better it will in general be modelled by a Poisson process.

Due to its great importance in practice, we shall study the Poisson process in detail in this chapter. First (Sec. 6.2) we base our study on a physical model with main emphasis upon the distributions associated to the process, and then we shall consider some important properties of the Poisson process (Sec. 6.3). Finally, in Sec. 6.4 we consider the interrupted Poisson process as an example of generalisation.

6.1 Characteristics of the Poisson process

The fundamental properties of the Poisson process are defined in Sec. 5.2:

a. *Stationary*,

b. *Independent* at all time instants (epochs), and

c. *Simple*.

(b) and (c) are fundamental properties, whereas (a) is unnecessary. Thus we may allow a Poisson process to have a time–dependent intensity. From the above properties we may derive other properties that are sufficient for defining the Poisson process. The two most important ones are:

- *Number representation*: The number of events within a time interval of fixed length is *Poisson distributed*. Therefore, the process is named *the Poisson process*.
- *Interval representation*: The time distance X_i (5.2) between consecutive events is *exponentially distributed*.

In this case using (4.8) and (4.10) *Feller–Jensen's identity* (5.4) shows the fundamental relationship between the cumulated Poisson distribution and the Erlang distribution (Sec. 6.2.2):

$$\sum_{j=0}^{n-1} \frac{(\lambda t)^j}{j!} \cdot \mathrm{e}^{-\lambda t} = \int_{x=t}^{\infty} \frac{(\lambda x)^{n-1}}{(n-1)!}\, \lambda \cdot \mathrm{e}^{-\lambda x}\, \mathrm{d}x = 1 - F(t)\,. \tag{6.1}$$

This formula can also be obtained by repeated partial integration.

6.2 Distributions of the Poisson process

In this section we consider the Poisson process in a dynamical and physical way (Fry, 1928 [30]) & (Jensen, 1954 [11]). The derivations are based on a simple physical model and concentrate on the probability distributions associated with the Poisson process.

The physical model is as follows: Events (arrivals) are placed at random on the real axis in such a way that every event is placed *independently* of all other events. So we put the events uniformly and independently on the real axes.

The average density is chosen as λ events (arrivals) per time unit. If we consider the axis as a time axis, then on the average we shall have λ arrivals per time unit. The probability that a given arrival pattern occurs within a time interval is independent of the location of the interval on the time axis.

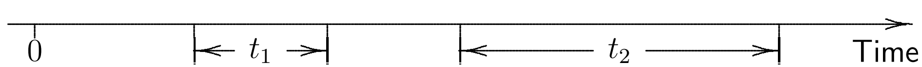


Figure 6.1: *When deriving the Poisson process, we consider arrivals within two non–overlapping time intervals of duration t_1 and t_2, respectively.*

Let $p(\nu, t)$ denote the probability that ν events occur within a time interval of duration t. The mathematical formulation of the above model is as follows:

1. *Independence:* If t_1 and t_2 are two non–overlapping intervals (Fig. 6.1), we have because of the independence assumption:

$$p\,(0, t_1) \cdot p\,(0, t_2) = p\,(0, t_1 + t_2)\ . \tag{6.2}$$

2. The mean value of the time interval between two successive arrivals is $1/\lambda$ (3.4):

$$\int_0^\infty p(0,t)\,\mathrm{d}t = \frac{1}{\lambda}, \qquad 0 < \frac{1}{\lambda} < \infty\,. \tag{6.3}$$

   Here $p(0,t)$ is the probability that there are no arrivals within the time interval $(0,t)$, which is identical to the probability that the time until the first event is larger than t (the complementary distribution). The mean value (6.3) is obtained directly from (3.4). Formula (6.3) can also be interpreted as the area under the curve $p(0,t)$, which is a never–increasing function decreasing from 1 to 0.

3. We notice that (6.2) implies that the event *"no arrivals within the interval of length 0"* is sure to take place:

$$p(0,0) = 1\,. \tag{6.4}$$

4. We also notice that (6.3) implies that the probability of *"no arrivals within a time interval of length* ∞*"* is zero and never takes place:

$$p(0,\infty) = 0\,. \tag{6.5}$$

6.2.1 Exponential distribution

The fundamental step in the following derivation of the Poisson distribution is to derive $p(0,t)$ which is the probability of no arrivals within a time interval of length t, i.e. the probability that the first arrival appears later than t. We will show that $\{1 - p(0,t)\}$ is an exponential distribution (cf. Sec. 4.1).

From (6.2) we have:

$$\ln p\left(0,t_1\right) + \ln p\left(0,t_2\right) = \ln p\left(0,t_1+t_2\right)\,. \tag{6.6}$$

Letting $\ln p(0,t) = f(t)$, (6.6) can be written as:

$$f\left(t_1\right) + f\left(t_2\right) = f\left(t_1+t_2\right)\,. \tag{6.7}$$

By differentiation with respect to e.g. t_2 we have:

$$f'(t_2) = f'_{t_2}\left(t_1+t_2\right)\,.$$

From this we notice that $f'(t)$ must be a constant and therefore:

$$f(t) = a + b\,t\,. \tag{6.8}$$

By inserting (6.8) into (6.7), we obtain $a = 0$. Therefore $p(0,t)$ has the form:

$$p(0,t) = \mathrm{e}^{bt}\,.$$

From (6.3) we obtain b:

$$\frac{1}{\lambda} = \int_0^\infty p(0,t)\,\mathrm{d}t = \int_0^\infty \mathrm{e}^{bt}\,\mathrm{d}t = -\frac{1}{b}\,,$$

or:

$$b = -\lambda\,.$$

Thus on the basis of item (1) and (2) above we have shown that:

$$p(0,t) = \mathrm{e}^{-\lambda t}\,. \tag{6.9}$$

If we consider $p(0,t)$ as the probability that the next event arrives later than t, then the time until next arrival is exponentially distributed (Sec. 4.1):

$$1 - p(0,t) = F(t) \;=\; 1 - \mathrm{e}^{-\lambda t}, \qquad \lambda > 0\,, \quad t \geq 0\,, \tag{6.10}$$

$$F'(t) = \; f(t) \;=\; \lambda \cdot \mathrm{e}^{-\lambda t}\,, \qquad \lambda > 0\,, \quad t \geq 0\,. \tag{6.11}$$

We have the following mean value and variance (4.4):

$$\begin{aligned} m_1 &= \frac{1}{\lambda}\,, \\ \sigma^2 &= \frac{1}{\lambda^2}\,. \end{aligned} \tag{6.12}$$

The probability that the next arrival appears within the interval $(t, t+\mathrm{d}t)$ may be written as:

$$\begin{aligned} f(t)\,\mathrm{d}t &= \lambda \mathrm{e}^{-\lambda t}\,\mathrm{d}t \\ &= p(0,t)\,\lambda\,\mathrm{d}t\,, \end{aligned} \tag{6.13}$$

i.e. the probability that an arrival appears within the interval $(t, t+\mathrm{d}t)$ is equal to $\lambda\,\mathrm{d}t$, *independent of* t and *proportional to dt* (3.17).

Because λ is independent of the actual *age* t, the exponential distribution has no memory (cf. Secs. 4.1 & 3.1.2). The process has no age.

The parameter λ is called the intensity or rate of both the exponential distribution and of the related Poisson process and it corresponds to the intensity in (5.6). The exponential distribution is in general a very good model of call inter-arrival times when the traffic is generated by human beings (Fig. 6.2).

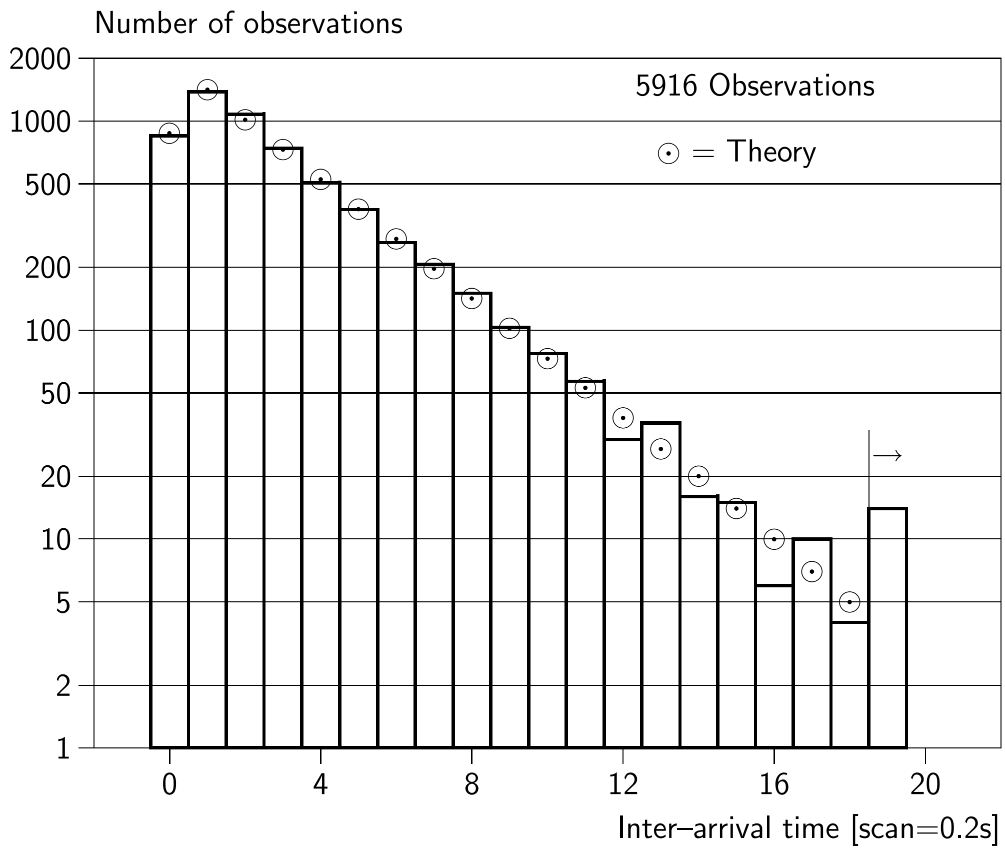


Figure 6.2: *Inter–arrival time distribution of calls at a transit exchange. The theoretical values are based on the assumption of exponentially distributed inter–arrival times. Due to the measuring principle (scanning method) the continuous exponential distribution is transformed into a discrete Westerberg distribution (15.14) (χ^2-test = 18.86 with 19 degrees of freedom, percentile = 53).*

6.2.2 Erlang–k distribution

From the above we notice that the time until exactly k arrivals have appeared is a sum of k *IID* (independently and identically distributed) exponentially distributed random variables.

The distribution of this sum is an *Erlang–k distribution* (Sec. 4.2) and the density is given by (4.8):

$$g_k(t)\,\mathrm{d}t = \lambda\,\frac{(\lambda t)^{k-1}}{(k-1)!}\,\mathrm{e}^{-\lambda t}\,\mathrm{d}t\,, \qquad \lambda > 0\,, \quad t \geq 0\,, \quad k = 1, 2, \ldots\,. \tag{6.14}$$

For $k = 1$ we of course get the exponential distribution. The distribution $g_{k+1}(t)$, $k > 0$, is obtained by convolving $g_k(t)$ and $g_1(t)$. If we assume that the expression (6.14) is valid for

$g_k(t)$, then we have by *convolution*:

$$\begin{aligned} g_{k+1}(t) &= \int_0^t g_k(t-x)\, g_1(x)\,\mathrm{d}x \\ &= \int_0^t \lambda\, \frac{\{\lambda(t-x)\}^{k-1}}{(k-1)!}\, \mathrm{e}^{-\lambda(t-x)}\, \lambda\, \mathrm{e}^{-\lambda x}\, \mathrm{d}t \\ &= \frac{\lambda^{k+1}}{(k-1)!}\, \mathrm{e}^{-\lambda t} \int_0^t (t-x)^{k-1}\, \mathrm{d}x \\ &= \lambda \cdot \frac{(\lambda t)^k}{k!} \cdot \mathrm{e}^{-\lambda t}\,. \end{aligned}$$

As the expression is valid for $k = 1$, we have by induction shown that it is valid for any k. The Erlang-k distribution is, from a statistical point of view, a special *gamma-distribution*.

The mean value and the variance are obtained from (6.12):

$$\begin{aligned} m_1 &= \frac{k}{\lambda}\,, \\ \sigma^2 &= \frac{k}{\lambda^2}\,, \\ \varepsilon &= 1 + \frac{1}{k}\,. \end{aligned} \tag{6.15}$$

Example 6.2.1: Call statistics from an SPC-system (cf. Example 5.1.2)
Let calls arrive to a stored program–controlled telephone exchange (*SPC–system*) according to a Poisson process. The exchange automatically collects full information about every 1000'th call. The inter-arrival times between two registrations will then be *Erlang–1000* distributed and have the form factor $\varepsilon = 1.001$, i.e. the registrations will take place very regularly. □

6.2.3 Poisson distribution

We shall now show that the number of arrivals in an interval of fixed length t is Poisson distributed with mean value λt. When we know the above-mentioned exponential distribution and the Erlang distribution, the derivation of the Poisson distribution is only a matter of applying simple combinatorics. The proof can be carried through by induction.

We want to derive $p(i, t)$ = probability of i arrivals within a time interval t. Let us assume that:

$$p(i-1, t) = \frac{(\lambda t)^{i-1}}{(i-1)!} \cdot \mathrm{e}^{-\lambda t}\,, \qquad \lambda > 0\,, \quad i = 1, 2, \ldots$$

This is correct for $i = 0$ (6.9). The interval $(0, t)$ is divided into three non–overlapping intervals $(0, t_1), (t_1, t_1 + \mathrm{d}t_1)$ and $(t_1 + \mathrm{d}t_1, t)$. From the earlier independence assumption we know that events within an interval are independent of events in the other intervals, because the intervals are non–overlapping. By choosing t_1 so that the last arrival within $(0, t)$ appears in $(t_1, t_1 + \mathrm{d}t_1)$, then the probability $p(i, t)$ is obtained by the integrating over all possible values of t_1 as a product of the following three probabilities:

a) The probability that $(i - 1)$ arrivals occur within the time interval $(0, t_1)$:

$$p\,(i-1, t_1) = \frac{(\lambda t_1)^{i-1}}{(i-1)!} \cdot \mathrm{e}^{-\lambda t_1}, \qquad 0 \le t_1 \le t\,.$$

b) The probability that there is just one arrival within the time interval from t_1 to $t_1 + \mathrm{d}t_1$:

$$\lambda\, \mathrm{d}t_1\,.$$

c) The probability that no arrivals occur from $t_1 + \mathrm{d}t_1$ to t:

$$\mathrm{e}^{-\lambda(t-t_1)}\,.$$

The product of the first two probabilities is the probability that the i'th arrival appears in $(t_1, t_1 + \mathrm{d}t_1)$, i.e. the *Erlang distribution* from the previous section.

By integration we have:

$$\begin{aligned}
p(i,t) &= \int_0^t \frac{(\lambda t_1)^{i-1}}{(i-1)!}\, \mathrm{e}^{-\lambda t_1}\, \lambda\, \mathrm{d}t_1\, \mathrm{e}^{-\lambda(t-t_1)} \\
&= \frac{\lambda^i}{(i-1)!}\, \mathrm{e}^{-\lambda t} \int_0^t t_1^{i-1}\, \mathrm{d}t_1\,, \\
p(i,t) &= \frac{(\lambda t)^i}{i!} \cdot \mathrm{e}^{-\lambda t}\,. \qquad (6.16)
\end{aligned}$$

This is the Poisson distribution which we thus have obtained from (6.9) by induction. The mean value and variance are:

$$m_1 = \lambda \cdot t\,, \qquad (6.17)$$

$$\sigma^2 = \lambda \cdot t\,. \qquad (6.18)$$

The Poisson distribution is in general a very good model for the number of calls in a telecommunication system (Fig. 6.3) or jobs in a computer system.

Example 6.2.2: Slotted Aloha Satellite System
Let us consider a digital satellite communication system with constant packet length h. The satellite

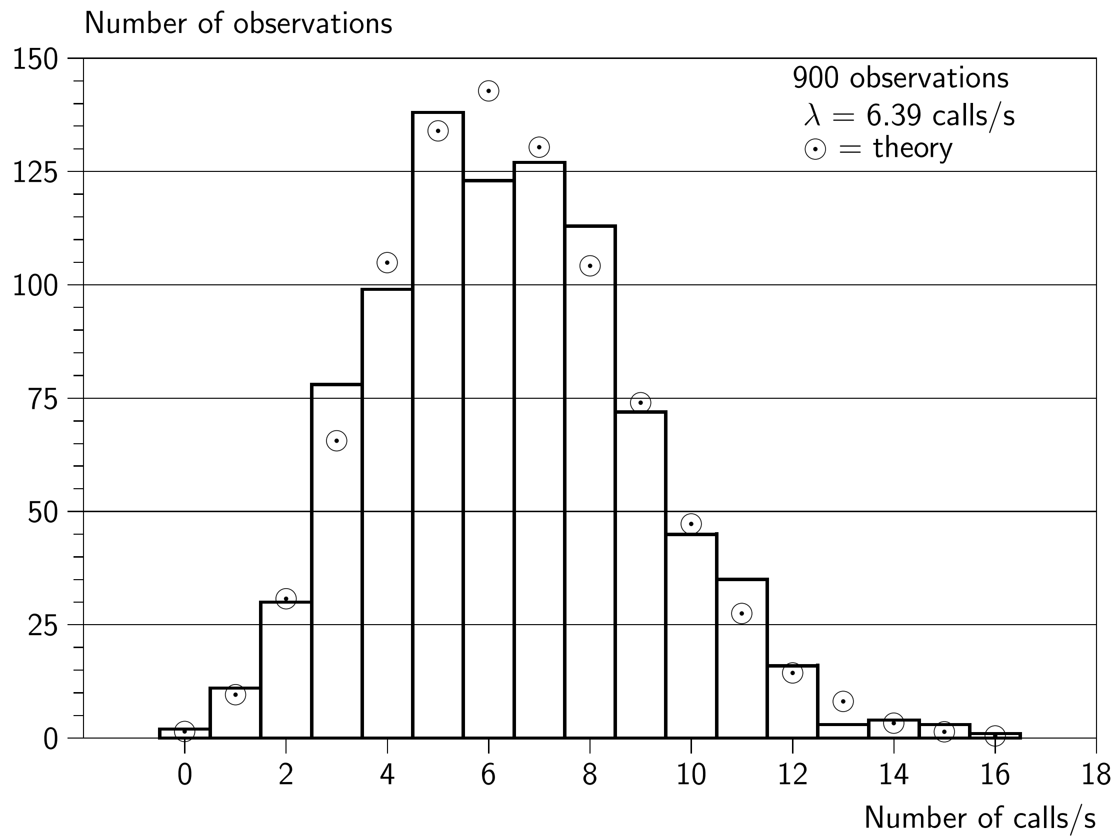


Figure 6.3: *Number of Internet dial-up calls per second. The theoretical values are based on the assumption of a Poisson distribution. A statistical test accepts the hypothesis of a Poisson distribution.*

is in a geostationary position about 36.000 km above equator, so the round trip delay is about 280 ms. The time axes is divided into slots of fixed duration corresponding to the packet length h. The individual terminal (earth station) transmits packets so that they are synchronised with the time slots. All packets generated during a time slot are transmitted in the next time-slot. The transmission of a packet is only correct if it is the only packet being transmitted in a time slot. If more packets are transmitted simultaneously, we have a collision and all packets are lost and must be retransmitted. All earth stations receive all packets and can thus decide whether a packet is transmitted correctly. Due to the time delay, the earth stations transmit packets independently. If the total arrival process is a Poisson process (rate λ), then we get a Poisson distributed number of packets in each time slot.

$$p(i) = \frac{(\lambda h)^i}{i!} \cdot \mathrm{e}^{-\lambda h}. \tag{6.19}$$

The probability of a correct transmission is:

$$p(1) = \lambda h \cdot \mathrm{e}^{-\lambda h}. \tag{6.20}$$

This corresponds to the proportion of the time axes which is utilised effectively. This function, which is shown in Fig. 6.4, has an optimum for $\lambda h = 1$, as the derivative with respect to λh is zero for this value:

$$p'_{\lambda h}(1) = \mathrm{e}^{-\lambda h} \cdot (1 - \lambda h)\,, \tag{6.21}$$

$$\mathrm{Max}\{p(1)\} = e^{-1} = 0.3679\,. \tag{6.22}$$

We thus have a maximum utilisation of the channel equal to 0.3679, when on the average we transmit one packet per time slot. A similar result holds when there is a limited number of terminals and the number of packets per time slot is Binomially distributed. □

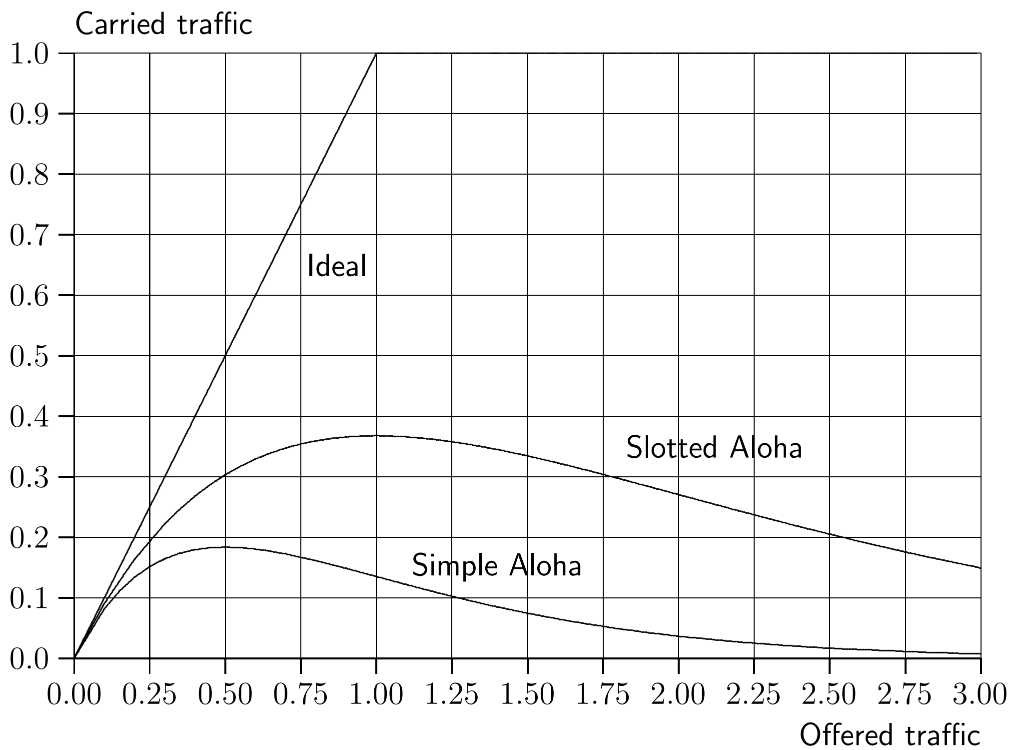


Figure 6.4: *The carried traffic in a slotted Aloha system has a maximum (example 6.2.2). The Simple Aloha protocol is dealt with in example 7.2.1.*

6.2.4 Static derivation of the distributions of the Poisson process

As it is known from statistics, these distributions can also be derived from the *Binomial process* by letting the number of trials n (e.g. throws of a die) increase to infinity and at the same time letting the probability of success in a single trial p converge to zero in such a way that the average number of successes $n \cdot p$ is constant.

This approach is static and does not stress the fundamental properties of the *Poisson process* which has a dynamic independent existence. But it shows the relationship between the two processes as illustrated in Table 6.1.

The exponential distribution is the *only continuous* distribution with lack of memory, and the geometrical distribution is the *only discrete* distribution with lack of memory. For example, the next outcome of a throw of a die is independent of the previous outcome. The distributions of the two processes are shown in Table 6.1.

6.3 Properties of the Poisson process

In this section we shall show some fundamental properties of the Poisson process. From the physical model in Sec. 6.2 we have seen that the Poisson process is the most random point process that may be found (*maximum disorder process*). It yields a good description of physical processes when many different factors are behind the total process.

6.3.1 Palm's theorem (Superposition theorem)

The fundamental properties of the Poisson process among all other point processes were first discussed by the Swede Conny Palm. He showed that the exponential distribution plays the same role for stochastic point processes (e.g. inter–arrival time distributions), where point processes are superposed, as the Normal distribution does when stochastic variables are added up (the central limit theorem).

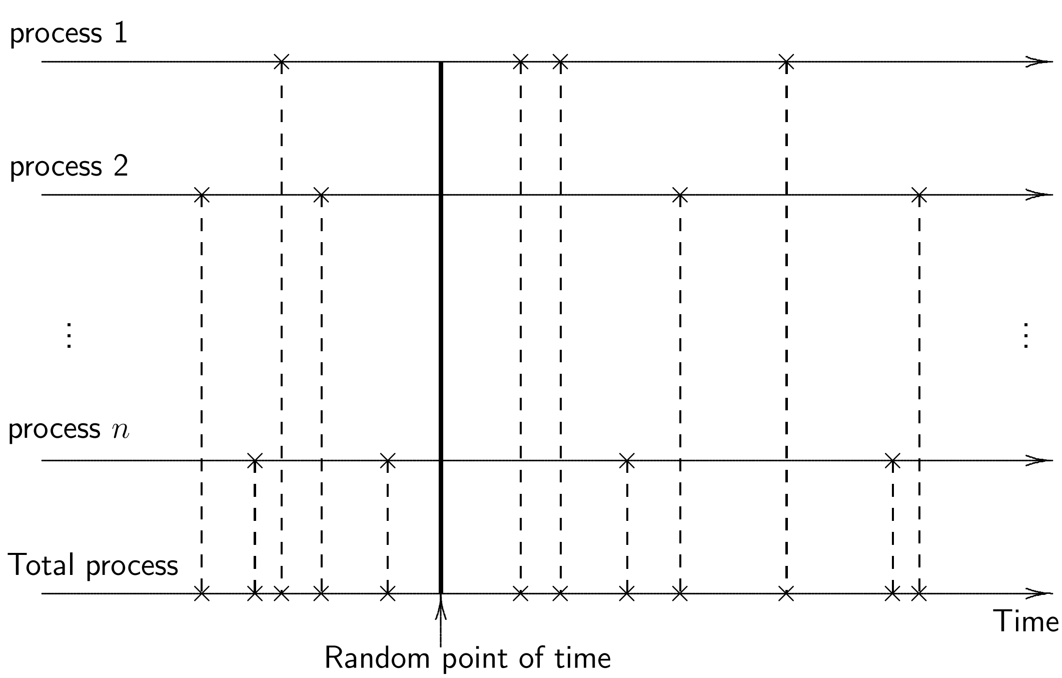


Figure 6.5: *By superposition of n point processes we obtain under certain assumptions a process which locally is a Poisson process.*

BINOMIAL PROCESS Discrete time Probability of success: $p\,, \quad 0 < p < 1$	POISSON PROCESS Continuous time Intensity of succes: $\lambda\,, \quad \lambda > 0$
Number of attempts since previous success or since a random attempt to get a success	Interval between two successes or from a random point until next success
GEOMETRIC DISTRIBUTION $p(n) = p \cdot (1-p)^{n-1}\,, \quad n = 1, 2, \ldots$ $m_1 = \dfrac{1}{p}\,, \qquad \sigma^2 = \dfrac{1-p}{p^2}$	EXPONENTIAL DISTRIBUTION $f(t) = \lambda \cdot \mathrm{e}^{-\lambda t}\,, \quad t \geq 0$ $m_1 = \dfrac{1}{\lambda}\,, \qquad \sigma^2 = \dfrac{1}{\lambda^2}$
Number of attempts to get k successes	Time interval until k'th success
PASCAL = NEGATIVE BINOMIAL DISTR. $p(n\|k) = \dbinom{n-1}{k-1} p^k\,(1-p)^{n-k}\,, \; n \geq k$ $m_1 = \dfrac{k}{p}\,, \qquad \sigma^2 = \dfrac{k(1-p)}{p^2}$	ERLANG–K DISTRIBUTION $f(t\|k) = \dfrac{(\lambda t)^{k-1}}{(k-1)!} \cdot \lambda \cdot \mathrm{e}^{-\lambda t}\,, \qquad t \geq 0$ $m_1 = \dfrac{k}{\lambda}\,, \qquad \sigma^2 = \dfrac{k}{\lambda^2}$
Number of successes in n attempts	Number of successes in a time interval t
BINOMIAL DISTRIBUTION $p(x\|\,n) = \dbinom{n}{x} p^x\,(1-p)^{n-x}\,, \quad x = 0, 1, \ldots$ $m_1 = p\,n\,, \qquad \sigma^2 = p\,n \cdot (1-p)$	POISSON DISTRIBUTION $f(x\|\,t) = \dfrac{(\lambda t)^x}{x!} \cdot \mathrm{e}^{-\lambda t}\,, \quad t \geq 0$ $m_1 = \lambda\,t\,, \qquad \sigma^2 = \lambda\,t$

Table 6.1: *Correspondence between the distributions of the Binomial process and the Poisson process. A success corresponds to an event or an arrival in a point process. Mean value =* m_1, *variance =* σ^2. *For the geometric distribution we may start with a zero class. The mean value is then reduced by one whereas the variance is the same.*

Theorem 6.1 *Palm's theorem: by superposition of many independent point processes the resulting total process will locally be a Poisson process.*

The term"*locally*" means that we consider time intervals which are so short that each process contributes at most with one event during this interval. This is a natural requirement since no process may dominate the total process (similar conditions are assumed for the central limit theorem). The theorem is valid only for simple point processes. If we consider a random point of time in a certain process, then the time until the next arrival is given by (3.23).

We superpose n processes into one total process. By appropriate choice of the time unit the mean distance between arrivals in the total process is kept constant, independent of n. The time from a random point of time to the next event in the total process is then given by (3.23):

$$p\{T \le t\} = 1 - \prod_{i=1}^{n} \left\{ 1 - V_i\left(\frac{t}{n}\right) \right\} . \tag{6.23}$$

If all sub-processes are identical, we get:

$$p\{T \le t\} = 1 - \left\{ 1 - V\left(\frac{t}{n}\right) \right\}^n . \tag{6.24}$$

From (3.23) and (5.18) we find (letting $\mu = 1$):

$$\lim_{\Delta t \to 0} v(\Delta t) = 1 ,$$

and thus:

$$V(\Delta t) = \int_0^{\Delta t} 1 \, \mathrm{d}t = \Delta t . \tag{6.25}$$

Therefore, we get from (6.24) by letting the number of sub-processes increase to infinity:

$$\begin{aligned} p\{T \le t\} &= \lim_{n\to\infty} \left\{ 1 - \left(1 - \frac{t}{n} \right)^n \right\} \\ &= 1 - \mathrm{e}^{-t} . \end{aligned} \tag{6.26}$$

which is the exponential distribution. We have thus shown that by superposition of identical processes we locally get a Poisson process. In a similar way we may superpose non-identical processes and obtain a Poisson process locally.

Example 6.3.1: Life-time of a route in an ad-hoc network
A route in a network consists of a number of links connecting the end-points of the route (Chap. 11). In an ad-hoc network links exist for a limited time period. The life-time of a route is therefore the time until the first link is disconnected. From Palm's theorem we see that the life-time of the route tends to be exponentially distributed. □

6.3.2 Raikov's theorem (Decomposition theorem)

A similar theorem, *the decomposition theorem*, is valid when we split a point process into sub-processes, when this is done in a random way. If there are n times fewer events in a sub-process, then it is natural to reduce the time axes with a factor n.

Theorem 6.2 *Raikov's theorem: by a random decomposition of a point process into sub-processes, the individual sub-process converges to a Poisson process, when the probability that an event belongs to the sub-process tends to zero.*

In addition to superposition and decomposition (merge and split, or join and fork), we can make another operation on a point process, namely *translation* (displacement) of the individual events. When this translation for every event is a random variable, independent of all other events, an arbitrary point process will converge to a Poisson process.

As concerns point processes occurring in real–life, we may, according to the above, expect that they are Poisson processes when a sufficiently large number of independent conditions for having an event are fulfilled. This is why the Poisson process is a good description of for instance the arrival processes to a local exchange from all local subscribers.

As an example of limitations in Palm's theorem (Theorem 6.1) it can be shown that the superposition of two independent processes yields an exact Poisson process only if both sub–processes are Poisson processes.

6.3.3 Uniform distribution – a conditional property

In Sec. 6.2 we have seen that *a uniform distribution in a very large interval* corresponds to a Poisson process. The inverse property is also valid:

Theorem 6.3 *If for a Poisson process we have n arrivals within an interval of duration t, then these arrivals are uniformly distributed within this interval.*

The length of this interval can itself be a random variable if it is independent of the Poisson process. This is for example the case in traffic measurements with variable measuring intervals (Chap. 15). This can be shown both from the Poisson distribution (number representation) and from the exponential distribution (interval presentation).

6.4 Generalisation of the stationary Poisson process

The Poisson process has been generalised in many ways. In this section we only consider the interrupted Poisson process, but further generalisations are *MMPP* (Markov Modulated

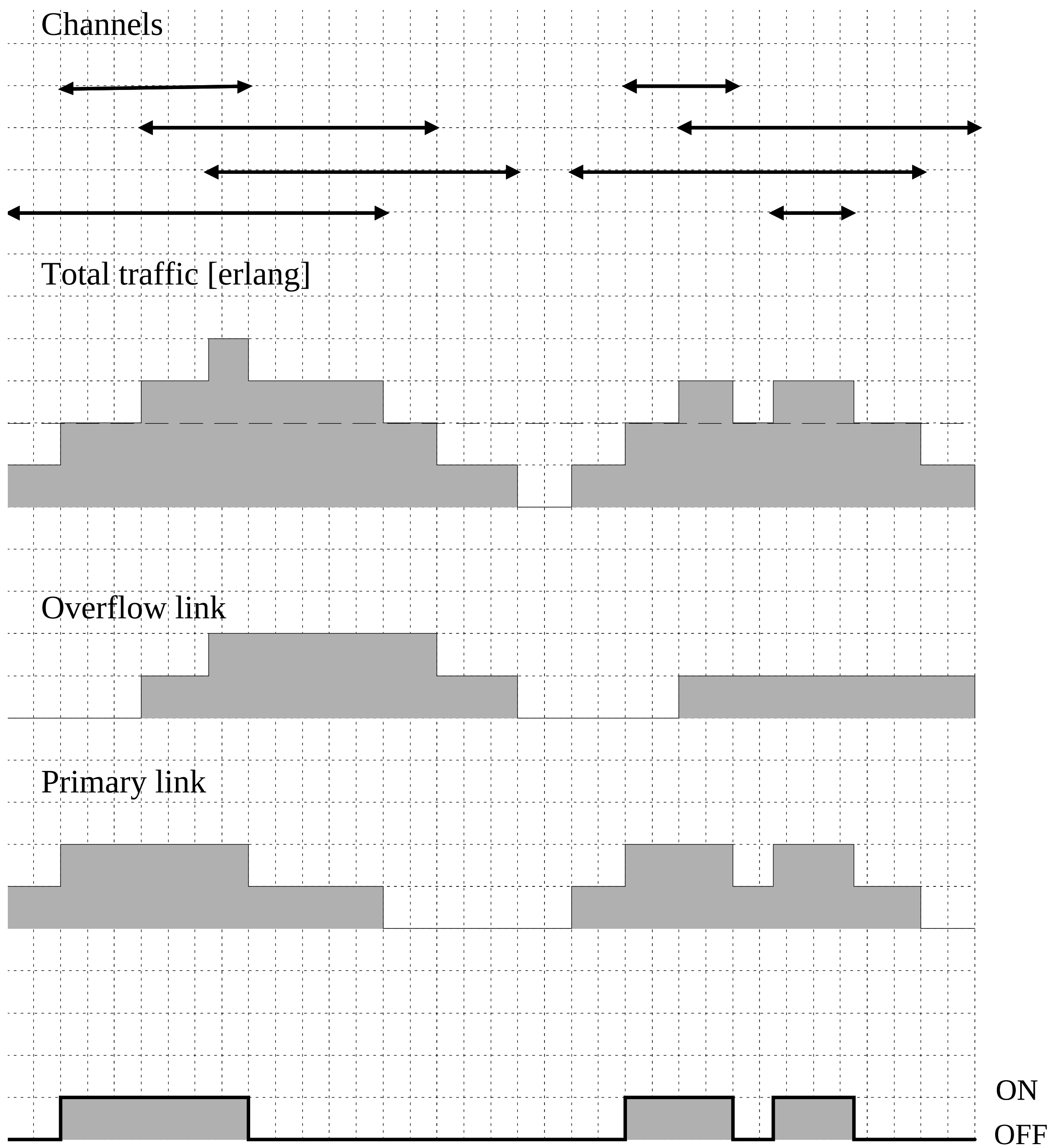


Figure 6.6: *Overflow system with Poisson arrival process (intensity λ). Normally, calls arrive to the primary group. During periods when all n trunks in the primary group are busy, all calls are offered to the overflow group.*

Poisson Processes) and *MAP* (Markov Arrival Processes).

6.4.1 Interrupted Poisson process (IPP)

Due to its lack of memory the Poisson process is very easy to apply. In some cases, however, the Poisson process is not sufficient to describe a real arrival process as it has only one parameter. Kuczura (1973 [70]) proposed a generalisation which has been widely used.

The idea of generalisation comes from the overflow problem (Fig. 6.6 & Sec. 9.2). Customers arriving at the system will first try to be served by a primary system with limited capacity (n servers). If the primary system is busy, then the arriving customers will be served by the overflow system. Arriving customers are routed to the overflow system only when the

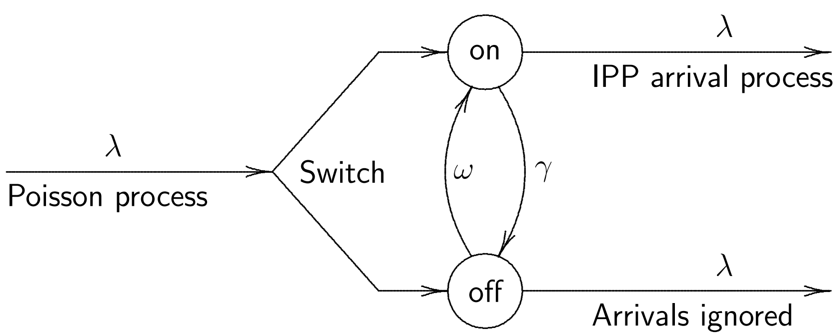


Figure 6.7: *Illustration of the interrupted Poisson process (IPP) (cf. Fig. (6.6)). The position of the switch is controlled by a two-state Markov process.*

primary system is busy. During the busy periods customers arrive at the overflow system according to the Poisson process with intensity λ. During the non-busy periods no calls arrive to the overflow system, i.e. the arrival intensity is zero. Thus we can consider the arrival process to the overflow system as a Poisson process which is either *On* or *Off* (Fig. 6.7). As a

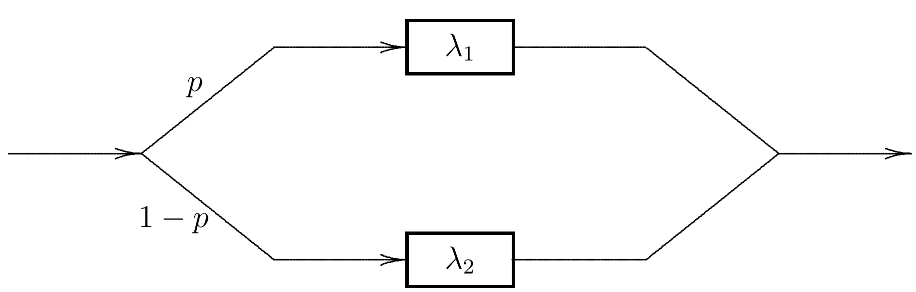


Figure 6.8: *The interrupted Poisson process is equivalent to a hyper–exponential arrival process (6.27).*

simplified model to describe these *On* (*Off*) intervals, Kuczura used exponentially distributed time intervals with intensity γ (ω). He showed that this corresponds to hyper-exponentially distributed inter–arrival times to the overflow link, which are illustrated by a phase–diagram in Fig 6.8. It can be shown that the parameters are related as follows:

$$\begin{aligned}
\lambda &= p\,\lambda_1 + (1-p)\lambda_2\,, \\
\lambda \cdot \omega &= \lambda_1 \cdot \lambda_2\,, \qquad (6.27)\\
\lambda + \gamma + \omega &= \lambda_1 + \lambda_2\,.
\end{aligned}$$

Because a hyper–exponential distribution with two phases can be transformed into a *Cox–2* distribution (Sec. 4.4.2), the *IPP* arrival process is a *Cox-2* arrival processes as shown in Fig. 4.10. We have three parameters available, whereas the Poisson process has only one parameter. This makes it more flexible for modelling empirical data.

Chapter 7

Erlang's loss system and B–formula

In this and the following chapters we consider the classical teletraffic theory developed by Erlang, Engset and Fry & Molina. It has successfully been applied for more than 80 years. In this chapter we only consider the fundamental Erlang-B formula. In Sec. 7.1 we put forward the assumptions for the model. Sec. 7.2 deals with the case with infinite capacity, which results in a Poisson distributed number of busy channels. In Sec. 7.3 we consider a limited number of channels and obtain the truncated Poisson distribution and Erlang's B-formula. In Sec. 7.4 we describe a standard procedure (cook book) for dealing with state transition diagrams. This is the key to classical teletraffic theory. We also derive an accurate recursive formula for numerical evaluation of Erlang's B-formula in Sec. 7.5. Finally, in Sec. 7.6 we study the basic principles of dimensioning, where we balance the Grade–of–Service (GoS) and the costs of the system.

7.1 Introduction

Erlang's B-formula is based on the following model, described by the three elements *structure, strategy, and traffic:*

a. *Structure:* We consider a system of n identical channels (servers, trunks, slots) working in parallel. This is called a *homogeneous group*.

b. *Strategy:* A call arriving at the system is accepted for service if any channel is idle. Every call needs one and only one channel. We say the group has *full accessibility*. Often the term *full availability* is used, but this terminology will only be used in connection with reliability aspect. If all channels are busy a call attempt is lost, and it disappears without any after-effect (the rejected call attempt may be accepted by an alternative route). This strategy is the most important one and has been applied with success for many years. It is called *Erlang's loss model* or the *Lost Calls Cleared* = *LCC*–model.

c. *Traffic:* In the following we assume that the service times are exponentially distributed with intensity μ (corresponding to a mean value $1/\mu$), and that the arrival process is a Poisson process with rate λ. This type of traffic is called *Pure Chance Traffic type One, PCT-I*. The traffic process then becomes a *pure birth and death process*, a simple Markov process which is easy to deal with mathematically.

   Definition of offered traffic: We define the offered traffic as the traffic carried when the number of channels (the capacity) is infinite (2.2). In Erlang's loss model with Poisson arrival process this definition of offered traffic is equivalent to the average number of call attempts per mean holding time:

$$A = \lambda \cdot \frac{1}{\mu} = \frac{\lambda}{\mu} \,. \tag{7.1}$$

We consider two cases:

1. $n = \infty$: Poisson distribution (Sec. 7.2),

2. $n < \infty$: Truncated Poisson distribution (Sec. 7.3).

We shall later see that the model is insensitive to the holding time distribution, i.e. only the mean holding time is of importance for the state probabilities. The type of distribution has no importance for the state probabilities.

Performance–measures: The most important grade-of-service measures for loss systems are time congestion E, call congestion B, and traffic (load) congestion C. They are all equal for Erlang's loss model because of the Poisson arrival process (*PASTA*–property, Sec. 6.3).

7.2 Poisson distribution

We assume the arrival process is a Poisson process and that the holding times are exponentially distributed, i.e. we consider *PCT-I* traffic. The number of channels is assumed to be infinite, so we never observe congestion (blocking).

7.2.1 State transition diagram

We define the state of the system, $[\,i\,]$, as the number of busy channels i $(i = 0, 1, 2, \ldots)$. In Fig. 7.1 all states of the system are shown as circles, and the rates by which the traffic process changes from one state to another state are shown upon the arcs of arrows between the states. As the process is simple, we only have transitions to neighbouring states. If we assume the system is in *statistical equilibrium*, then the system will be in state $[\,i\,]$ the proportion $p(i)$ of time, where $p(i)$ is the probability of observing the system in state $[\,i\,]$ at a random point of time, i.e. a time average. When the process is in state $[\,i\,]$ it will jump to state $[\,i\!+\!1]$ λ

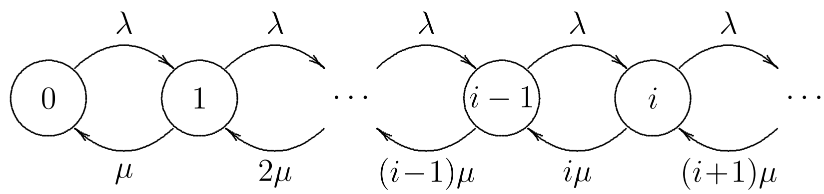


Figure 7.1: *The Poisson distribution. State transition diagram for a system with infinitely many channels, Poisson arrival process (λ), and exponentially distributed holding times (μ).*

times per time unit and to state $[\,i-1]$ $i\,\mu$ times per time unit. Of course, the process will leave state $[\,i\,]$ at the moment there is a state transition.

The equations describing the states of the system under the assumption of statistical equilibrium can be set up in two ways, which both are based on the principle of global balance:

a. *Node equations*
In statistical equilibrium the number of transitions per time unit into state $[\,i\,]$ equals the number of transitions out of state $[\,i\,]$. The equilibrium state probability $p(i)$ denotes the proportion of time (total time per time unit) the process is in state $[\,i\,]$. The average number of jumps from state $[\,0\,]$ to state $[\,1\,]$ is $\lambda\, p(0)$ per time unit, and the average number of jumps from state $[\,1\,]$ to state $[\,0\,]$ is $\mu\, p(1)$ per time unit. For state $[\,i\,]$ we get the following equilibrium or *balance* equation:

$$\lambda \cdot p(0) = \mu \cdot p(1)\,, \qquad i = 0\,, \tag{7.2}$$

$$\lambda \cdot p(i-1) + (i+1)\,\mu \cdot p(i+1) = (\lambda + i\,\mu) \cdot p(i)\,, \quad i > 0\,. \tag{7.3}$$

The node equations are always applicable, also for state transition diagrams in several dimensions, which we consider later.

b. *Cut equations*
In many cases we may exploit a simple structure of the state transition diagram. If we put a fictitious cut for example between the states $[\,i-1\,]$ and $[\,i\,]$ (corresponding to a global cut around the states $[\,0\,], [\,1\,], \ldots [\,i-1]\,$), then in statistical equilibrium the traffic process changes from state $[\,i-1\,]$ to $[\,i\,]$ the same number of times as it changes from state $[\,i\,]$ to $[\,i-1\,]$. In statistical equilibrium we thus have per time unit:

$$\lambda \cdot p(i-1) = i\,\mu \cdot p(i)\,, \qquad i = 1, 2, \ldots\,. \tag{7.4}$$

Cut equations are primarily used for one-dimensional state transition diagrams, whereas node equations are applicable to any diagram.

As the system always will be in some state, we have the normalisation restriction:

$$\sum_{i=0}^{\infty} p(i) = 1\,, \qquad p(i) \geq 0\,. \tag{7.5}$$

We notice that node equations (7.3) involve three state probabilities, whereas the cut equations (7.4) only involve two. Therefore, it is easier to solve the cut equations. Loss system will always be able to enter statistical equilibrium if the arrival process is independent of the state of the system. We shall not consider the mathematical conditions for statistical equilibrium in this chapter.

7.2.2 Derivation of state probabilities

For one-dimensional state transition diagrams the application of cut equations is the most appropriate approach. From Fig. 7.1 we get the following balance equations:

$$
\begin{aligned}
\lambda \cdot p(0) &= \mu \cdot p(1)\,, \\
\lambda \cdot p(1) &= 2\,\mu \cdot p(2)\,, \\
\dots \quad & \quad \dots \\
\lambda \cdot p(i{-}2) &= (i-1)\,\mu \cdot p(i{-}1)\,, \\
\lambda \cdot p(i{-}1) &= i\,\mu \cdot p(i)\,, \\
\lambda \cdot p(i) &= (i+1)\,\mu \cdot p(i{+}1)\,, \\
\dots \quad & \quad \dots\,.
\end{aligned}
$$

Expressing all state probabilities by $p(0)$ yields, as we introduce the offered traffic $A = \lambda/\mu$:

$$
\begin{array}{lclcr}
p(0) & & & = & p(0)\,, \\
p(1) & & & = & A \cdot p(0)\,, \\
p(2) & = & \dfrac{A}{2} \cdot p(1) & = & \dfrac{A^2}{2} \cdot p(0)\,, \\
\dots & & \dots & & \dots \\
p(i{-}1) & = & \dfrac{A}{i{-}1} \cdot p(i{-}2) & = & \dfrac{A^{i-1}}{(i-1)!} \cdot p(0)\,, \\
p(i) & = & \dfrac{A}{i} \cdot p(i{-}1) & = & \dfrac{A^i}{i\,!} \cdot p(0)\,, \\
p(i{+}1) & = & \dfrac{A}{i{+}1} \cdot p(i) & = & \dfrac{A^{i+1}}{(i+1)!} \cdot p(0)\,, \\
\dots & & \dots & & \dots
\end{array}
$$

The normalisation constraint (7.5) implies:

$$
\begin{aligned}
1 &= \sum_{j=0}^{\infty} p(j) \\
&= p(0) \cdot \left\{ 1 + A + \frac{A^2}{2\,!} + \cdots + \frac{A^i}{i\,!} + \ldots \right\} \\
&= p(0) \cdot \mathrm{e}^{A} \,, \\
p(0) &= \mathrm{e}^{-A} \,,
\end{aligned}
$$

and thus the Poisson distribution:

$$
p(i) = \frac{A^i}{i\,!} \cdot \mathrm{e}^{-A} \,, \quad i = 0, 1, 2, \ldots \,. \tag{7.6}
$$

The number of busy channels at a random point of time is therefore Poisson distributed with both mean value (6.17) and variance (6.18) equal to A. We have earlier shown that the number of calls in a fixed time interval also is Poisson distributed (6.16). Thus the Poisson distribution is valid both in time and in space. We would, of course, obtain the same solution by using node equations.

7.2.3 Traffic characteristics of the Poisson distribution

From a dimensioning point of view, the system with unlimited capacity is not very interesting. We summarise the important traffic characteristics of the loss system:

$$
\begin{aligned}
&\text{Time congestion:} & E &= 0 \,, \\
&\text{Call congestion:} & B &= 0 \,, \\
&\text{Carried traffic:} & Y &= \sum_{i=1}^{\infty} i \cdot p(i) = A \,, \\
&\text{Lost traffic:} & A_\ell &= A - Y = 0 \,, \\
&\text{Traffic congestion:} & \mathrm{C} &= 0 \,.
\end{aligned}
$$

Carried traffic by the i'th trunk assuming sequential hunting is given later in (7.14).
Peakedness Z is defined as the ratio between variance and mean value of the distribution of state probabilities (cf. *IDC*, Index of Dispersion of Counts). For the Poisson distribution we find (6.17) & (6.18):

$$
Z = \frac{\sigma^2}{m_1} = 1 \,. \tag{7.7}
$$

The peakedness has dimension [*number of channels*] and is different from the coefficient of variation which has no dimension (3.9).

Duration of state $[\,i\,]$*:*

In state $[\,i\,]$ the process has the total intensity $(\lambda + i\,\mu)$ away from the state. Therefore, the time until the first transition (state transition to either $i+1$ or $i-1$) is exponentially distributed (Sec. 4.1.1):

$$f_i(t) = (\lambda + i\,\mu)\mathrm{e}^{-(\lambda + i\,\mu)\,t}\,, \quad t \geq 0\,.$$

Example 7.2.1: Simple Aloha protocol
In example 6.2.2 we considered the slotted Aloha protocol, where the time axes was divided into time slots. We now consider the same protocol in continuous time. We assume that packets arrive according to a Poisson process and that they are of constant length h. The system corresponds to the Poisson distribution which also is valid for constant holding times (Sec. 7.2). The state probabilities are given by the Poisson distribution (7.6), where $A = \lambda\,h$. A packet is only transmitted correctly if (a) the system is in state $[\,0\,]$ at the arrival time and (b) no other packets arrive during the service time h. We find:

$$p_{correct} = p(0) \cdot \mathrm{e}^{-\lambda\,h} = \mathrm{e}^{-2\,A}\,.$$

The traffic transmitted correctly thus becomes:

$$A_{correct} = A \cdot p_{correct} = A \cdot \mathrm{e}^{-2\,A}\,.$$

This is the proportion of the time axis which is utilised efficiently. It has an optimum for $\lambda h = A = 1/2$, where the derivative with respect to A equals zero:

$$\frac{\partial A_{correct}}{\partial A} = \mathrm{e}^{-2\,A} \cdot (1 - 2\,A)\,,$$

$$\max\{A_{correct}\} = \frac{1}{2e} = 0.1839\,. \tag{7.8}$$

We thus obtain a maximum utilisation equal to 0.1839 when we offer 0.5 erlang. This is half the value we obtained for a slotted system by synchronising the satellite transmitters. The models are compared in Fig. 6.4. □

7.3 Truncated Poisson distribution

We still assume *Pure Chance Traffic Type I (PCT-I)* as in Sec. 7.2. The number of channels is now limited so that n is finite. The number of states becomes $n+1$, and the state transition diagram is shown in Fig. 7.2.

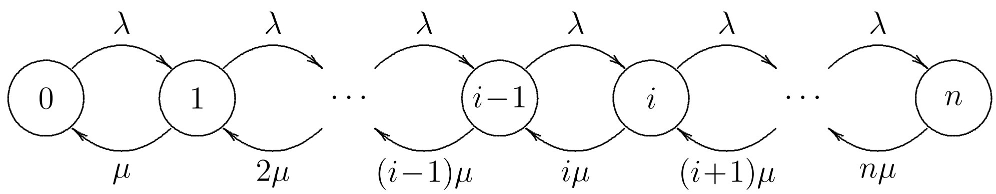


Figure 7.2: *The truncated Poisson distribution. State transition diagram for a system with a limited number of channels (n), Poisson arrival process (λ), and exponential service times (μ).*

7.3.1 State probabilities

We get similar cut equations as for the Poisson case, but the number of states is limited and the normalisation condition (7.5) now becomes:

$$p(0) = \left\{ \sum_{j=0}^{n} \frac{A^j}{j!} \right\}^{-1} .$$

We get the so-called *truncated Poisson distribution* (Erlang's first formula):

$$p(i) = \frac{\dfrac{A^i}{i!}}{\displaystyle\sum_{j=0}^{n} \frac{A^j}{j!}} , \qquad 0 \le i \le n . \tag{7.9}$$

The name *truncated* means cut off and is due to the fact that the solution may be interpreted as a conditional Poisson distribution $p(i \,|\, i \le n)$. This is easily seen by multiplying both numerator and denominator by e^{-A}.

7.3.2 Traffic characteristics of Erlang's B-formula

Knowing the state probabilities we are able to find performance measures defined by state probabilities.

Time congestion:

The probability that all n channels are busy at a random point of time is equal to the proportion of time all channels are busy (time average). This is obtained from (7.9) for $i = n$:

$$E_n(A) = p(n) = \frac{\dfrac{A^n}{n!}}{1 + A + \dfrac{A^2}{2!} + \cdots + \dfrac{A^n}{n!}} . \tag{7.10}$$

This is *Erlang's famous B-formula* (1917, [11]). It is denoted by $E_n(A) = E_{1,n}(A)$, where index "one" refers to the alternative name *Erlang's first formula*.

Call congestion:

The probability that a random call will be lost is equal to the proportion of call attempts blocked. If we consider one time unit, we find $B = B_n(A)$:

$$B = \frac{\lambda \cdot p(n)}{\sum_{\nu=0}^{n} \lambda \cdot p(\nu)} = p(n) = E_n(A)\,. \tag{7.11}$$

Carried traffic:

If we use the cut equation for the cut between states $[\,i-1\,]$ and $[\,i\,]$ we get:

$$\begin{aligned} Y &= \sum_{i=1}^{n} i \cdot p(i) = \sum_{i=1}^{n} \frac{\lambda}{\mu} \cdot p(i-1) = A \cdot \{1 - p(n)\}\,, \\ Y &= A \cdot \{1 - E_n(A)\}\,, \end{aligned} \tag{7.12}$$

where A is the offered traffic. The carried traffic will be less than both A and n.

Lost traffic:

$$A_\ell = A - Y = A \cdot E_n(A)\,.$$

Traffic congestion:

$$C = \frac{A - Y}{A} = E_n(A)\,.$$

We thus have $E = B = C$ because the call intensity is independent of the state. This is the *PASTA*–property which is valid for all systems with Poisson arrival processes. In all other cases at least two of the three measures of congestion are different. Erlang's B-formula is shown graphically in Fig. 7.3 for some selected values of the parameters.

Traffic carried by the i'th trunk (the utilisation a_i):

1. *Random hunting:* In this case all channels carry the same traffic on the average. The total carried traffic is independent of the hunting strategy and we find the utilisation:

$$a_i = a = \frac{Y}{n} = \frac{A\,\{1 - E_n(A)\}}{n}\,. \tag{7.13}$$

   This function is shown in Fig. 7.4, and we observe that for a given congestion E we obtain the highest utilisation for large channel groups (*economy of scale*).

2. *Ordered hunting = sequential hunting:* The traffic carried by channel i is the difference between the traffic lost from $i-1$ channels and the traffic lost from i channels:

$$a_i = A \cdot \{E_{i-1}(A) - E_i(A)\}\,. \tag{7.14}$$

It should be noticed that the traffic carried by channel i is independent of the total number of channels. Thus channels after channel i have no influence upon the traffic carried by channel i. There is no feed-back.

Improvement function:

This denotes the increase in carried traffic when the number of channels is increased by one from n to $n+1$:

$$\begin{aligned} F_n(A) &= Y_{n+1} - Y_n\,, \\ &= A\{1-E_{n+1}\} - A\{1-E_n\}\,, \end{aligned} \tag{7.15}$$

$$\begin{aligned} F_n(A) &= A\,\{E_n(A) - E_{n+1}(A)\} \\ &= a_{n+1}\,. \end{aligned} \tag{7.16}$$

We have:

$$0 \le F_n(A) \le 1\,.$$

The boundary values can be shown to be:

$$\begin{aligned} F_0(A) &= \frac{A}{1+A}\,, \qquad F_\infty(A) = 0\,, \\ F_n(0) &= 0\,, \qquad F_n(\infty) = 1\,. \end{aligned}$$

The improvement function $F_n(A)$ is tabulated in *Moe's Principle* (Arne Jensen, 1950 [49]) and shown in Fig. 7.5. In Sec. 7.6.2 we consider the application of this principle for optimal economic dimensioning.

Peakedness:

This is defined as the ratio between the variance and the mean value of the distribution of the number of busy channels, cf. *IDC* (5.11). For the truncated Poisson distribution it can be shown by using (7.14) that:

$$Z = \frac{\sigma^2}{m} = 1 - A\,\{E_{n-1}(A) - E_n(A)\} = 1 - a_n\,, \tag{7.17}$$

The dimension is [*number of channels*]. In a group with ordered hunting we may thus estimate the peakedness from the traffic carried by the last channel.

Duration of state $[\,i\,]$*:*

The total intensity for leaving state $[\,i\,]$ is constant and equal to $(\lambda + i\,\mu)$, and therefore the duration of the time in state $[\,i\,]$ (sojourn time) is exponentially distributed with density function:

$$\begin{aligned} f_i(t) &= (\lambda + i\,\mu)\cdot \mathrm{e}^{-(\lambda + i\,\mu)\,t}\,, \quad 0 \le i < n\,, \\ f_n(t) &= (n\,\mu)\cdot \mathrm{e}^{-(n\,\mu)\,t}\,, \qquad i = n\,. \end{aligned} \tag{7.18}$$

7.3.3 Generalisations of Erlang's B-formula

The literature on the *B–formula* is very extensive. Here we only mention a couple of important properties.

Insensitivity:
It can be shown that Erlang's B-formula, which above is derived under the assumption of exponentially distributed holding times, is valid for arbitrary holding time distributions. The state probabilities for both the Poisson distribution (7.6) and the truncated Poisson distribution (7.9) only depend on the holding time distribution through the mean value which is included in the offered traffic A. It can be shown that all classical loss systems with full accessibility are insensitive to the holding time distribution.

The fundamental assumption for the validity of Erlang's B-formula is thus a Poisson arrival process. According to *Palm's theorem* this is fulfilled when the traffic is originated by many independent sources. This is fulfilled in ordinary telephone systems under normal traffic conditions. The formula is thus very robust. Both the arrival process and the service time process are described by a single parameter A. This explains the wide application of the B-formula both in the past and today.

Continuous number of channels:
Erlang's B–formula can mathematically be generalised to non-integral number of channels (including a negative number of channels). This is useful when we for instance want to find the number of channels n for a given offered traffic A and blocking probability E. In Chap. 9 we will also use this for dealing with overflow traffic.

7.4 Standard procedures for state transition diagrams

The most important tool in teletraffic theory is formulation and solution of models by means of state transition diagrams. From the previous sections we identify the following standard procedure for dealing with state transition diagrams. It consists of a number of steps and is formulated in general terms. The procedure is also applicable for multi-dimensional state transition diagrams, which we consider later. We always go through the following steps:

a. Construction of the state transition diagram.

   - Define the states of the system in an unique way,
   - Draw the states as circles,
   - Consider the states one at a time and draw all possible arrows for transitions away from the state due to
     - the arrival process (new arrival or phase shift in the arrival process),

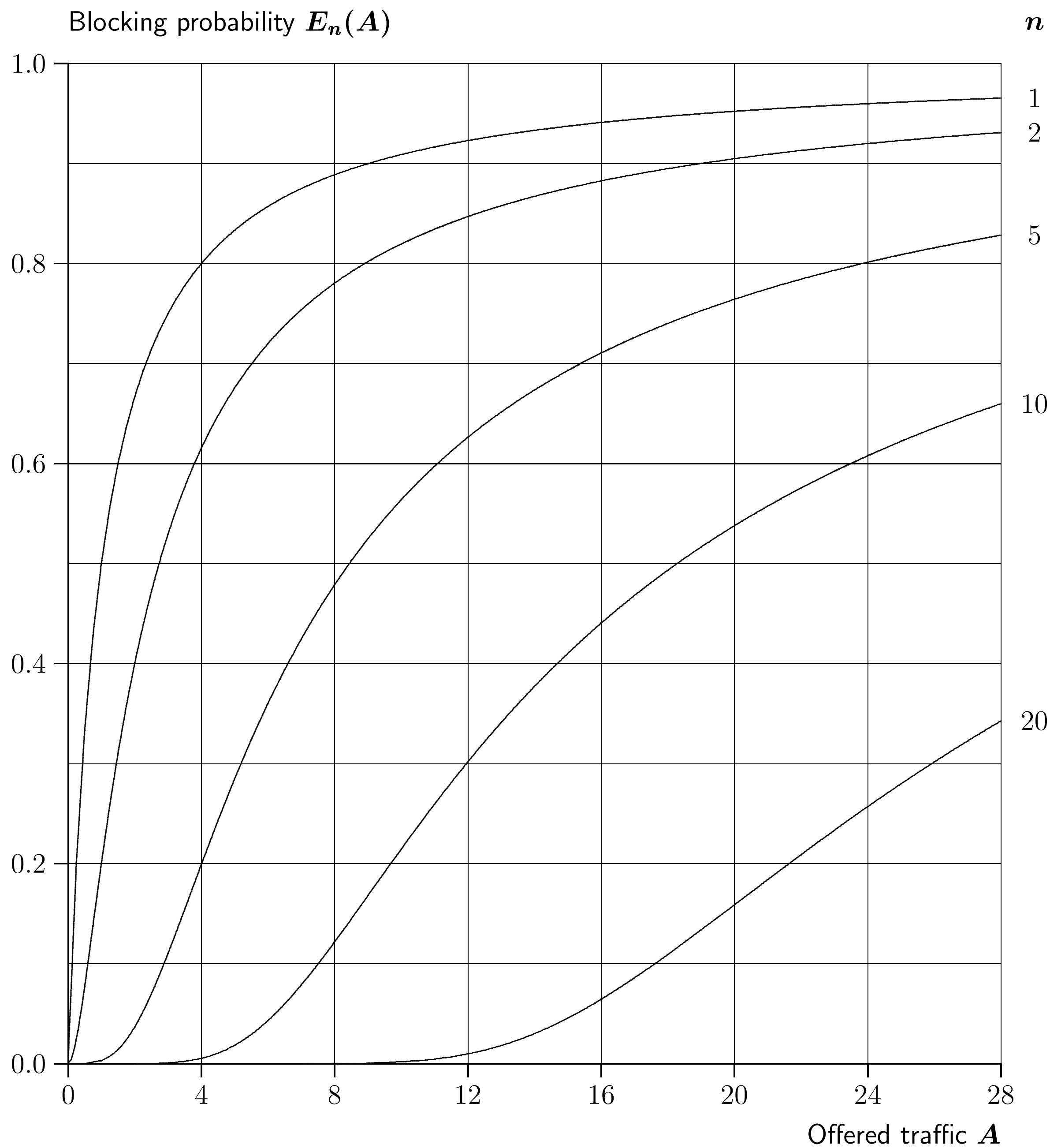


Figure 7.3: *Blocking probability $E_n(A)$ as a function of the offered traffic A for various values of the number of channels n (7.9).*

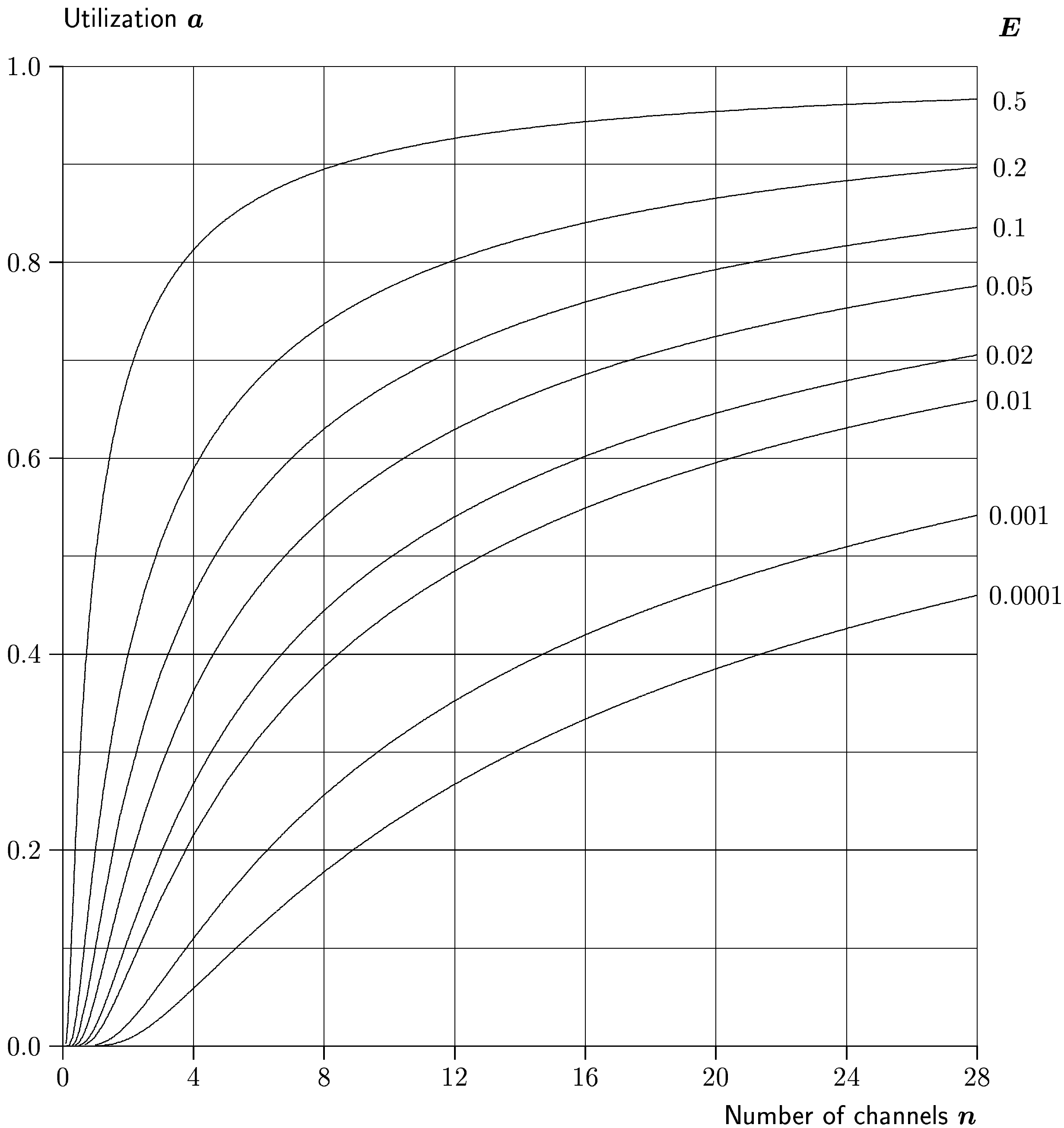


Figure 7.4: *The average utilisation per channel a (7.13) as a function of the number of channels n for given values of the congestion E.*

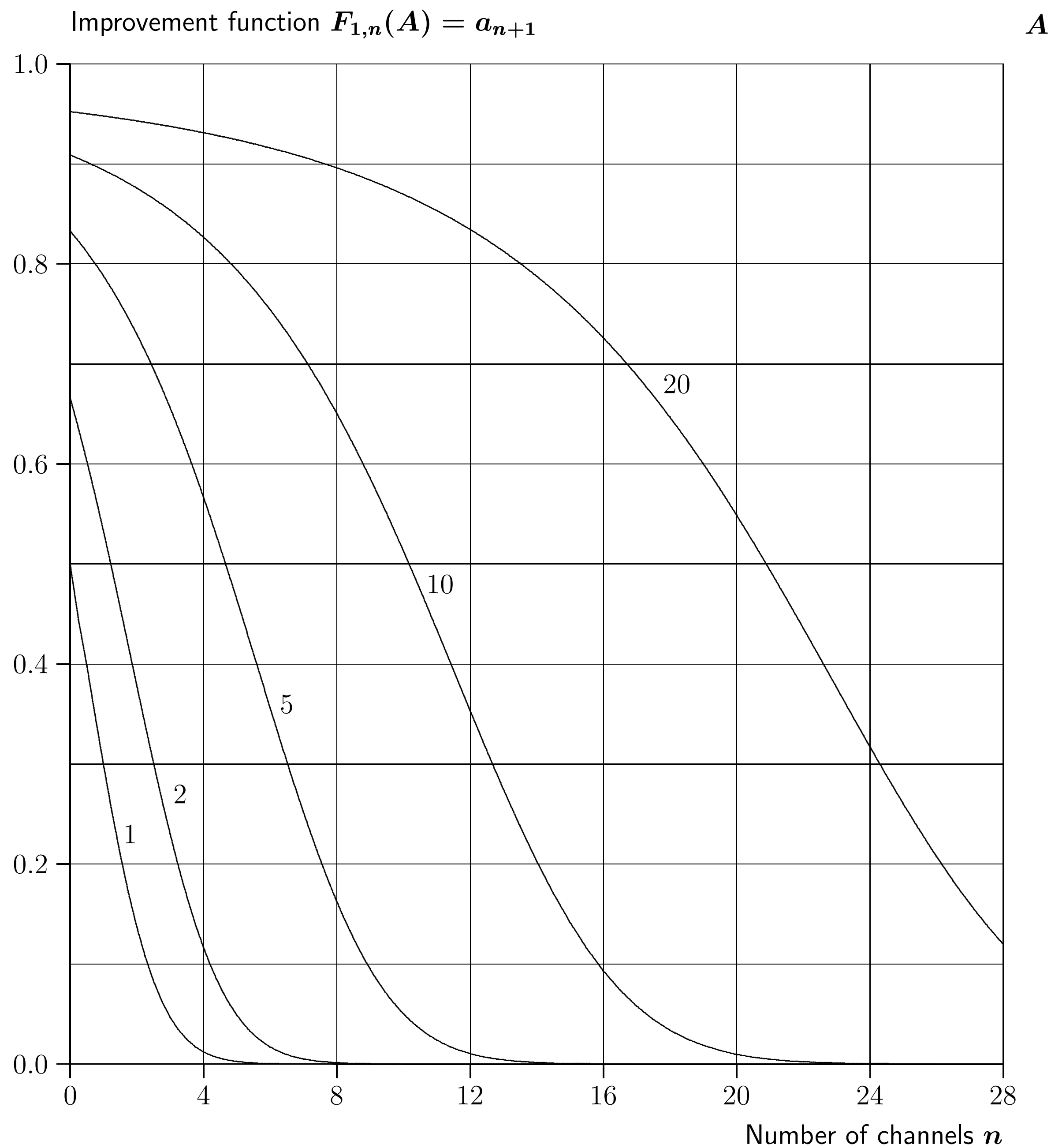


Figure 7.5: *Improvement function $F_n(A)$ (7.16) of Erlang's B–formula. By sequential hunting $F_n(A)$ equals the traffic a_{n+1} carried on channel number $(n+1)$.*

    * the departure (service) process (the service time terminates or shifts phase).

  In this way we obtain the complete state transition diagram.

b. Set up the equations describing the system.

  – If the conditions for statistical equilibrium are fulfilled, the steady state equations can be obtained from:
    * node equations,
    * cut equations.

c. Solve the balance equations assuming statistical equilibrium.

  – Express all state probabilities by for example the probability of state $[\,0\,]$, $p(0)$.
  – Find $p(0)$ by normalisation.

d. Calculate the performance measures expressed by the state probabilities.

In practise, we let the non-normalised value of the state probability $q(0)$ equal to one, and then calculate the relative values $q(i), (i = 1, 2, \ldots)$. By normalising we then find:

$$p(i) = \frac{q(i)}{Q_n}, \quad i = 0, 1, \ldots, n\,, \tag{7.19}$$

where

$$Q_n = \sum_{\nu=0}^{n} q(\nu)\,. \tag{7.20}$$

The time congestion becomes:

$$p(n) = \frac{q(n)}{Q_n} = 1 - \frac{Q_{n-1}}{Q_n}\,. \tag{7.21}$$

7.4.1 Recursion formula

If $q(i)$ becomes very large (e.g. 10^{10}), then we may multiply all $q(i)$ by the same constant (e.g. 10^{-10}) as we know that all probabilities are within the interval $[0, 1]$. In this way we avoid numerical problems. If $q(i)$ becomes very small, then we may truncate the state space as the density function of $p(i)$ often will be bell-shaped (unimodal) and therefore has a maximum. In many cases we are theoretically able to control the error introduced by truncating the state space (Stepanov, 1989 [95]).

We may of course also normalise after every step which implies more calculations. If we only are interested in the absolute value of $p(n)$, i.e. in obtaining the time congestion $E = p(n)$, we can do it in a simpler way. Let us assume we have the following recursion formula (based on cut equations) for the non-normalised state probabilities:

$$q(x) = \frac{\lambda_{x-1}}{x\,\mu} \cdot q(x-1)\,, \quad x = 1, 2, \ldots\ , \tag{7.22}$$

and that we want to find the normalised state probabilities, given x channels: $p_x(i)\,, i = 0, 1, \cdots x$. The index x indicates that it is the state probabilities for a system with a total of x channels.

We assume we already have obtained the normalised state probabilities for $x-1$ channels:

$$\{p_{x-1}(0),\ p_{x-1}(1),\ p_{x-1}(2),\ \ldots\ ,\ p_{x-1}(x-1)\}\,.$$

The relative values of state probabilities do not change when we add one channel more, so we let:

$$q_x(i) = p_{x-1}(i)\,, \quad i = 0, 1, 2, \ldots, x-1,$$

and calculate:

$$q_x(x) = \frac{\lambda_{x-1}}{x\,\mu} \cdot p_{x-1}(x-1) = \frac{\lambda_{x-1}}{x\,\mu} \cdot q_x(x-1)\,.$$

The new normalisation constant becomes:

$$\begin{aligned} Q_x &= \sum_{i=0}^{x} q_x(i) \\ &= 1 + q_x(x)\,, \end{aligned}$$

because we in the previous step normalised the sum of terms ranging from 0 to $x-1$ so they add to one. We thus get:

$$p_x(x) = \frac{q_x(x)}{Q_x} \tag{7.23}$$

$$= \frac{q_x(x)}{1 + q_x(x)}\,. \tag{7.24}$$

The initial value for the recursion is $Q_0 = p_0(0) = q_0(0) = 1$. Inserting (7.22) and using the notation $E_x = p_x(x)$ (time congestion) we get:

$$E_x = \frac{\lambda_{x-1} \cdot E_{x-1}}{x\,\mu + \lambda_{x-1} \cdot E_{x-1}}\,, \qquad E_0 = 1\,. \tag{7.25}$$

Introducing the inverse time congestion probability $I_x = E_x^{-1}$, we get:

$$I_x = 1 + \frac{x\,\mu}{\lambda_{x-1}} \cdot I_{x-1}\,, \qquad I_0 = 1\,. \tag{7.26}$$

This is a general recursion formula for calculating time congestion for all systems with state dependent arrival rates λ_i and homogeneous servers.

Example 7.4.1: Calculating probabilities of the Poisson distribution
If we want to calculate the Poisson distribution (7.6) for very large mean values $m_1 = A = \lambda/\mu$, then it is advantageously to let $q(m) = 1$, where m is equal to the integral part of $(m_1 + 1)$. The relative values of $q(i)$ for both decreasing values $i = m-1, m-2, \ldots, 0$ and for increasing values $i = m+1, m+2, \ldots$ will then be decreasing, and we may stop the calculations when for example $q(i) < 10^{-20}$ and finally normalise $q(i)$. In practice there will be no problems by normalising the probabilities. □

7.5 Evaluation of Erlang's B-formula

For numerical calculations the formula (7.10) is not very appropriate, since both $n!$ and A^n increase quickly so that overflow in the computer will occur. If we apply (7.25), then we get the recursion formula:

$$E_x(A) = \frac{A \cdot E_{x-1}(A)}{x + A \cdot E_{x-1}(A)}\,, \qquad E_0(A) = 1\,. \tag{7.27}$$

From a numerical point of view, the linear form (7.26) is the most stable:

$$I_x(A) = 1 + \frac{x}{A} \cdot I_{x-1}(A)\,, \qquad I_0(A) = 1\,, \tag{7.28}$$

where $I_n(A) = 1/E_n(A)$. This recursion formula is exact, and even for large values of (n, A) there are no round off errors. It is the basic formula for numerous tables of the Erlang B-formula, i.a. the classical table (Palm, 1947 [81]). For very large values of n there are more efficient algorithms. Notice that a recursive formula, which is accurate for increasing index, usually is inaccurate for decreasing index, and vice versa.

Example 7.5.1: Erlang's loss system
We consider an Erlang-B loss system with $n = 6$ channels, arrival rate $\lambda = 2$ calls per time unit, and departure rate $\mu = 1$ departure per time unit, so that the offered traffic is $A = 2$ erlang. If we denote the non-normalised relative state probabilities by $q(i)$, we get by setting up the state transition diagram the values shown in the following table:

i	$\lambda(i)$	$\mu(i)$	$q(i)$	$p(i)$	$i \cdot p(i)$	$\lambda(i) \cdot p(i)$
0	2	0	1.0000	0.1360	0.0000	0.2719
1	2	1	2.0000	0.2719	0.2719	0.5438
2	2	2	2.0000	0.2719	0.5438	0.5438
3	2	3	1.3333	0.1813	0.5438	0.3625
4	2	4	0.6667	0.0906	0.3625	0.1813
5	2	5	0.2667	0.0363	0.1813	0.0725
6	2	6	0.0889	0.0121	0.0725	0.0242
Total			7.3556	1.0000	1.9758	2.0000

We obtain the following blocking probabilities:

$$\begin{aligned}
&\textit{Time congestion:} \qquad & E_6(2) &= p(6) = 0.0121\,. \\
&\textit{Traffic congestion:} \qquad & C_6(2) &= \frac{A-Y}{A} = \frac{2-1.9758}{2} = 0.0121\,. \\
&\textit{Call congestion:} \qquad & B_6(2) &= \{\lambda(6)\cdot p(6)\} \Bigg/ \left\{\sum_{i=0}^{6} \lambda(i)\cdot p(i)\right\} = \frac{0.0242}{2.0000} = 0.0121\;.
\end{aligned}$$

We notice that $E = B = C$ due to the *PASTA*–property.

By applying the recursion formula (7.27) we of course obtain the same results:

$$\begin{aligned}
E_0(2) &= 1\,, \\
E_1(2) &= \frac{2\cdot 1}{1+2\cdot 1} = \frac{2}{3}\,, \\
E_2(2) &= \frac{2\cdot\frac{2}{3}}{2+2\cdot\frac{2}{3}} = \frac{2}{5}\,, \\
E_3(2) &= \frac{2\cdot\frac{2}{5}}{3+2\cdot\frac{2}{5}} = \frac{4}{19}\,, \\
E_4(2) &= \frac{2\cdot\frac{4}{19}}{4+2\cdot\frac{4}{19}} = \frac{2}{21}\,, \\
E_5(2) &= \frac{2\cdot\frac{2}{21}}{5+2\cdot\frac{2}{21}} = \frac{4}{109}\,, \\
E_6(2) &= \frac{2\cdot\frac{4}{109}}{6+2\cdot\frac{4}{109}} = \frac{4}{331} = 0.0121\,.
\end{aligned}$$

□

Example 7.5.2: Calculation of $E_x(A)$ for large x
By recursive application of (7.28) we find:

$$I_x(A) = 1 + \frac{x}{A} + \frac{x\,(x-1)}{A^2} + \cdots + \frac{x!}{A^x}\,,$$

which of course is the inverse blocking probability of the B-formula. For large values of x and A this formula can be applied for fast calculation of the B-formula, because we can truncate the sum when the terms become very small. □

7.6 Principles of dimensioning

When dimensioning service systems we have to balance grade-of-service requirements against economic restrictions. In this chapter we shall see how this can be done on a rational basis. In telecommunication systems there are several measures to characterise the service provided. The most extensive measure is *Quality-of-Service (QoS)*, comprising all aspects of a connection as voice quality, delay, loss, reliability etc. We consider a subset of these, *Grade-of-Service (GoS)* or network performance, which only includes aspects related to the capacity of the network.

By the publication of Erlang's formulæ there was already before 1920 a functional relationship between number of channels, offered traffic, and grade-of-service (blocking probability) and thus a measure for the quality of the traffic. At the same time there were direct connections between all exchanges in the Copenhagen area which resulted in many small trunk groups. If Erlang's B-formula were applied with a fixed blocking probability for dimensioning these groups, then the utilisation would become poor.

Kai Moe (1893-1949), who was chief engineer in the Copenhagen Telephone Company, made some quantitative economic evaluations and published several papers, where he introduced marginal considerations, as they are known today in mathematical economics. Similar considerations were done by P.A. Samuelson in his famous book, first published in 1947. On the basis of Moe's works the fundamental principles are formulated for telecommunication systems in *Moe's Principle* (Jensen, 1950 [49]).

7.6.1 Dimensioning with fixed blocking probability

For proper operation, a loss system should be dimensioned for a low blocking probability. In practice the number of channels n should be chosen so that $E_{1,n}(A)$ is about 1% to avoid overload due to many non-completed and repeated call attempts which both load the system and are a nuisance to subscribers.

Tab. 7.1 shows the offered traffic for a fixed blocking probability $E = 1\%$ for some values of n. The table also gives the average utilisation of channels, which is highest for large groups. If we increase the offered traffic by 20 % to $A_1 = 1.2 \cdot A$, we notice that the blocking probability increases for all n, but most for large values of n.

From Tab. 7.1 two features are observed:

a. The utilisation a per channel is, for a given blocking probability, highest in large groups (Fig. 7.4). A single channel can at a blocking probability $E = 1$ % on the average only be used 36 seconds per hour!

n	1	2	5	10	20	50	100
$A\ (E = 1\%)$	0.010	0.153	1.361	4.461	12.031	37.901	84.064
a	0.010	0.076	0.269	0.442	0.596	0.750	0.832
$F_{1,n}(A)$	0.000	0.001	0.011	0.027	0.052	0.099	0.147
$A_1 = 1.2 \cdot A$	0.012	0.183	1.633	5.353	14.437	45.482	100.877
E [%]	1.198	1.396	1.903	2.575	3.640	5.848	8.077
a	0.012	0.090	0.320	0.522	0.696	0.856	0.927
$F_{1,n}(A_1)$	0.000	0.002	0.023	0.072	0.173	0.405	0.617

Table 7.1: *Upper part: For a fixed value of the blocking probability $E = 1\%$ n trunks can be offered the traffic A. The average utilisation of the trunks is a, and the improvement function is $F_{1,n}(A)$ (7.16). Lower part: The values of E, a and $F_{1,n}(A)$ are obtained for an overload of 20%.*

b. Large channel groups are more sensitive to a given percentage overload than small channel groups. This is explained by the low utilisation of small groups, which therefore have a higher spare capacity (elasticity).

Thus two conflicting factors are of importance when dimensioning a channel group: we may choose among a high sensitivity to overload or a low utilisation of the channels.

7.6.2 Improvement principle (Moe's principle)

As mentioned in Sec. 7.6.1 a fixed blocking probability results in a low utilisation (bad economy) of small channel groups. If we replace the requirement of a fixed blocking probability with an economic requirement, then the improvement function $F_{1,n}(A)$ (7.16) should take a fixed value so that the extension of a group with one additional channel increases the carried traffic by the same amount for all groups.

In Tab. 7.2 we show the congestion for some values of n and an improvement value $F = 0.05$. We notice from the table that the utilisation of small groups becomes better corresponding to a high increase of the blocking probability. On the other hand the congestion in large groups decreases to a smaller value. See also Fig. 7.7. If therefore we have a telephone system with trunk group size and traffic values as given in the table, then we cannot increase the carried traffic by rearranging the channels among the groups.

This service criteria will therefore in comparison with Sec. 7.6.1 allocate more channels to large groups and fewer channels to small groups, which is the trend we were looking for.

n	1	2	5	10	20	50	100
$A\ (F_B = 0.05)$	0.271	0.607	2.009	4.991	11.98	35.80	78.73
a	0.213	0.272	0.387	0.490	0.593	0.713	0.785
$E_{1,n}(A)$ [%]	21.29	10.28	3.72	1.82	0.97	0.47	0.29
$A_1 = 1.2 \cdot A$	0.325	0.728	2.411	5.989	14.38	42.96	94.476
$E\ \{\%\}$	24.51	13.30	6.32	4.28	3.55	3.73	4.62
a	0.245	0.316	0.452	0.573	0.693	0.827	0.901
$F_{1,n}(A_1)$	0.067	0.074	0.093	0.120	0.169	0.294	0.452

Table 7.2: *For a fixed value of the improvement function we have calculated the same values as in table 7.1.*

The improvement function is equal to the difference quotient of the carried traffic with respect to number of channels n. When dimensioning according to the improvement principle we thus choose an operating point on the curve of the carried traffic as a function of the number of channels where the slope is the same for all groups ($\Delta A/\Delta n =$ constant). A marginal increase of the number of channels increases the carried traffic with the same amount for all groups.

It is easy to set up a simple economical model for determination of $F_{1,n}(A)$. Let us consider a certain time interval (e.g. a time unit). Denote the income per carried erlang per time unit by g. The cost of a cable with n channels is assumed to be a linear function:

$$c_n = c_0 + c \cdot n\,. \tag{7.29}$$

The total costs for a given number of channels is then (a) cost of cable and (b) cost due to lost traffic (missing income):

$$C_n = g \cdot A\, E_{1,n}(A) + c_0 + c \cdot n\,, \tag{7.30}$$

Here A is the offered traffic, i.e. the potential traffic demand on the group considered. The costs due to lost traffic will decrease with increasing n, whereas the expenses due to cable increase with n. The total costs may have a minimum for a certain value of n. In practice n is an integer, and we look for a value of n, for which we have (cf. Fig. 7.6):

$$C_{n-1} > C_n \quad \text{and} \quad C_n \leq C_{n+1}\,.$$

As $E_{1,n}(A) = E_n(A)$ we get:

$$A\,\{E_{n-1}(A) - E_n(A)\} > \frac{c}{g} \geq A\,\{E_n(A) - E_{n+1}(A)\}\ , \tag{7.31}$$

or:

$$F_{1,n-1}(A) > F_B \geq F_{1,n}(A)\,, \tag{7.32}$$

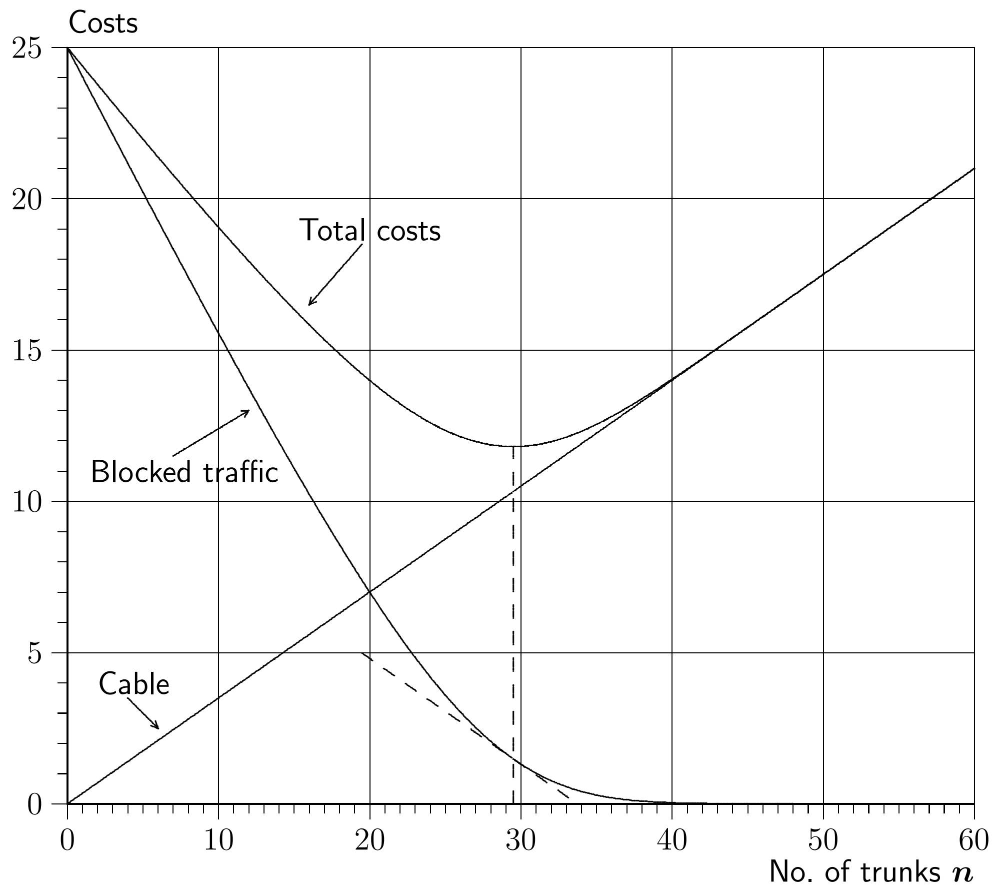


Figure 7.6: *The total costs are composed of costs for cable and lost income due to blocked traffic (7.30). Minimum of the total costs are obtained when (7.31) is fulfilled, i.e. when the two cost functions have the same slope with opposite signs (difference quotient). ($F_B = 0.35$, $A = 25$ erlang). Minimum is obtained for $n = 30$ trunks.*

where:

$$F_B = \frac{c}{g} = \frac{\textit{cost per extra channel}}{\textit{income per extra channel}} \;. \tag{7.33}$$

F_B is called *the improvement value*. We notice that c_0 does not appear in the condition for minimum. It determines whether it is profitable to carry traffic at all. We must require that for some positive value of n we have:

$$g \cdot A\,\{1 - E_n(A)\} > c_0 + c \cdot n\,. \tag{7.34}$$

Fig. 7.7 shows blocking probabilities for some values of F_B. We notice that the economic demand for profit results in a certain improvement value. In practice we choose F_B partly independent of the cost function.

In Denmark the following values have been used:

$$
\begin{aligned}
F_B &= 0.35 \quad \text{for primary trunk groups.} \\
F_B &= 0.20 \quad \text{for service protecting primary groups.} \\
F_B &= 0.05 \quad \text{for groups with no alternative route.}
\end{aligned}
\tag{7.35}
$$

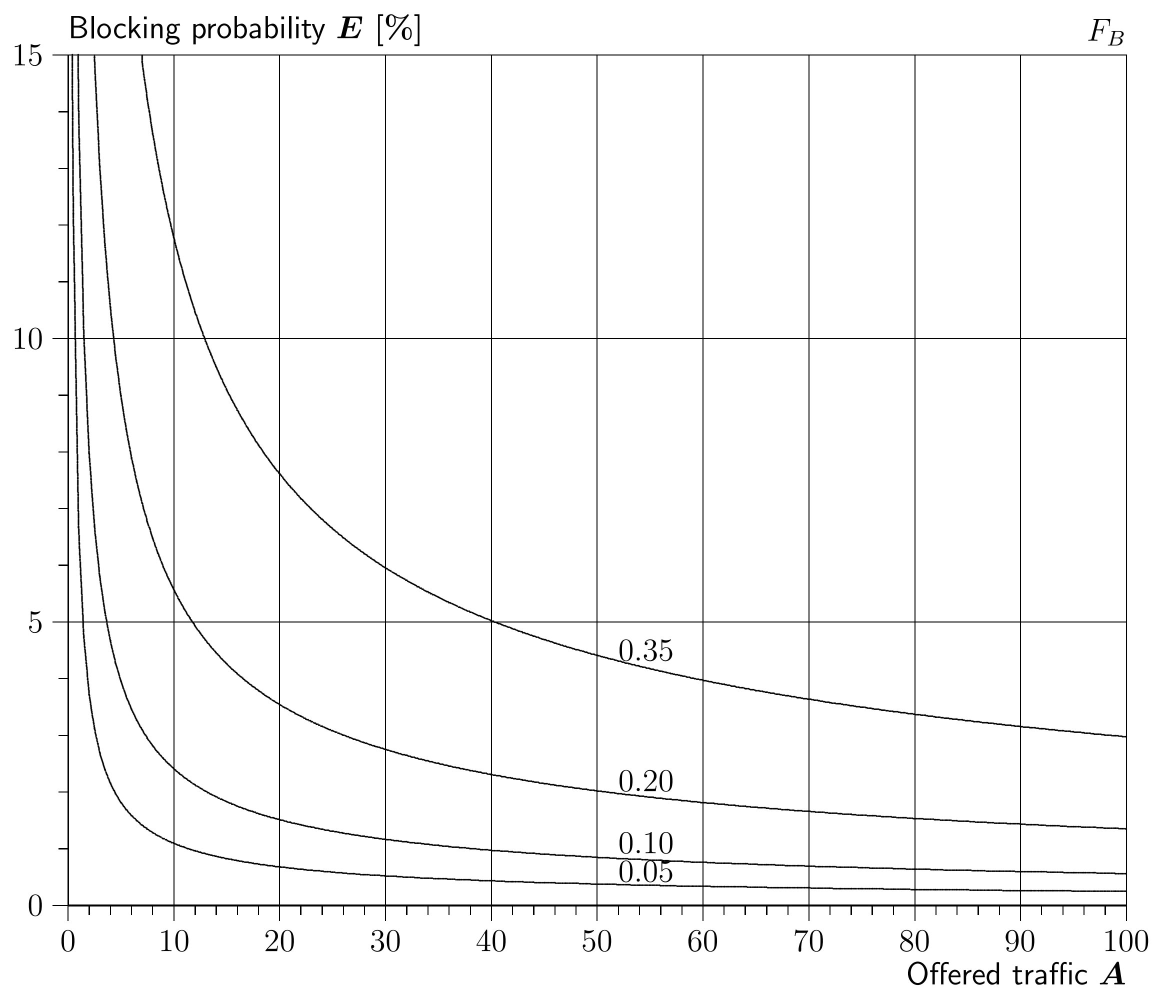


Figure 7.7: *When dimensioning with a fixed value of the improvement value F_B the blocking probabilities for small values of the offered traffic become large (cf. Tab. 7.2).*

Chapter 8

Loss systems with full accessibility

In this chapter we generalise Erlang's classical loss system to state-dependent Poisson-arrival processes, which include the so-called *BPP*-traffic models:

- *B*inomial case: *Engset's model*,
- *P*oisson case: *Erlang's model*, and
- *P*ascal (Negative Binomial) case: *Palm–Wallström's model.*

After an introduction in Sec. 8.1 we go through the basic classical theory in Secs. 8.2– 8.7. In Sec. 8.2 we consider the Binomial case, where the number of sources S (subscribers, customers, jobs) is limited and the number of channels n always is sufficient ($S \leq n$). This system is dealt with by balance equations in the same way as the Poisson case (Sec. 7.2). We consider the strategy *Lost-Calls-Cleared (LCC)*. In Sec. 8.3 we restrict the number of channels so that it becomes less than the number of sources ($n < S$). We may then get blocking and we obtain the truncated Binomial distribution, which also is called the *Engset distribution*. The probability of time congestion E is given by *Engset's formula*. With a limited number of sources, time congestion, call congestion, and traffic congestion differ, and the *PASTA*–property is replaced by the general *arrival theorem*, which tells that the state of the system observed by a customer (call average), is equal to the state probability of the system without this customer (time average). Engset's formula is computed numerically by a formula recursive in the number of channels n derived in the same way as for Erlang's B-formula. Also formulæ recursive in number of sources S and in both n & S are derived.

In Sec. 8.6 we consider the Negative Binomial case, also called the Pascal case, where the arrival intensity increases linearly with the state of the system. If the number of channels is limited, then we get *the truncated Negative Binomial distribution* (Sec. 8.7).

Figure 8.1: *A full accessible loss system with S sources, which generate traffic to n channels. The system is shown by a so-called chicko-gram. The beak of a source symbolises a selector which points upon the channels (servers) among which the source may choose.*

8.1 Introduction

We consider a system with same structure (full accessibility group) and strategy (Lost-Calls-Cleared) as in Chap. 7, but with more general traffic processes. In the following we assume the service times are exponentially distributed with intensity μ (mean value $1/\mu$); the traffic process then becomes a *birth & death process*, a special Markov process, which is easy to deal with mathematically. Usually we define the state of the system as the number of busy channels. We shall later see that the processes considered all are *insensitive* to the service time distribution, i.e. only the mean service time is of importance for the state probabilities, the service time distribution itself has no influence.

Definition of offered traffic: In Sec. 2.1 we define the offered traffic A as the traffic carried when the number of servers is unlimited, and this definition is used for both the Engset-case and the Pascal-case. The offered traffic is thus independent of the number of servers.
Only for stationary renewal processes as the Poisson arrival process in the Erlang case this definition is equivalent to the average number of calls attempts per mean service time. In the Engset and Pascal case we don't have renewal arrival processes.

Peakedness is defined as the ratio between variance and mean value of the state probabilities. For the offered traffic the peakedness is considered for an infinite number of channels.

We consider the following arrival processes, where the first case already has been dealt with in Chap. 7:

1. *Erlang-case* (P – Poisson-case):
   The arrival process is a Poisson process with intensity λ. This type of traffic is called *random traffic* or *Pure Chance Traffic type One, PCT–I*. We consider two cases:

   a. $n = \infty$: Poisson distribution (Sec. 7.2).
      The peakedness is in this case equal to one $Z = 1$.

   b. $n < \infty$: Truncated Poisson distribution (Sec. 7.3).

2. *Engset-case* (B – Binomial-case):
   There is a limited number of sources S. The individual source has a constant call (arrival) intensity γ when it is idle. When it is busy the call intensity is zero. The arrival process is thus state-dependent. If at a given point of time i sources are busy, then the arrival intensity is equal to $(S-i)\,\gamma$.

This type of traffic is called *Pure Chance Traffic type Two, PCT–II*. We consider the following two cases:

a. $n \geq S$: Binomial distribution (Sec. 8.2).
The peakedness is in this case less than one: $Z < 1$.

b. $n < S$: Truncated Binomial distribution (Sec. 8.3).

3. *Palm-Wallström–case* (P – Pascal-case):
There is a limited number of sources (customers) S. If at a given instant we have i busy sources, then the arrival intensity equals $(S+i)\gamma$. Again we have two cases:

a. $n = \infty$: Pascal distribution = Negative Binomial distribution (Sec. 8.6).
In this case peakedness is greater than one: $Z > 1$.

b. $n < \infty$: Truncated Pascal distribution (truncated negative Binomial distribution) (Sec. 8.7).

As the Poisson process may be obtained by an infinite number of sources with a total arrival intensity λ, the Erlang-case may be considered as a special case of the two other cases:

$$\lim_{\{S \to \infty\,,\ \gamma \to 0\}} S \cdot \gamma = \lambda\,.$$

For any finite state i we then have a constant arrival intensity: $(S \pm i) \cdot \gamma \simeq S \cdot \gamma = \lambda$.

The three traffic types are referred to as *BPP-traffic* according to the abbreviations given above (Binomial & Poisson & Pascal). As these models include all values of peakedness $Z > 0$, they can be used for modelling traffic with two parameters: mean value A and peakedness Z. For arbitrary values of Z the number of sources S in general becomes non-integral.

Performance–measures: The most important traffic characteristics for loss systems are time congestion E, Call congestion B, traffic congestion C, and the utilisation of the channels. These measures are derived for each of the above-mentioned models.

8.2 Binomial Distribution

We consider a system with a limited number of sources S (subscribers). The individual source switches between the states idle and busy. A source is idle during a time interval which is exponentially distributed with intensity γ, and the source is busy during an exponentially distributed time interval (service time, holding time) with intensity μ (Fig. 8.2). This kind of source is called a *sporadic source* or an *on/off* source. This type of traffic is called *Pure Chance Traffic type Two* (*PCT–II*) or *pseudo-random traffic*.

The number of channels/trunks n is in this section assumed to be greater than or equal to the number of sources ($n \geq S$), so that no calls are lost. Both n and S are assumed to be integers, but it is possible to deal with non–integral values (Iversen & Sanders, 2001 [43]).

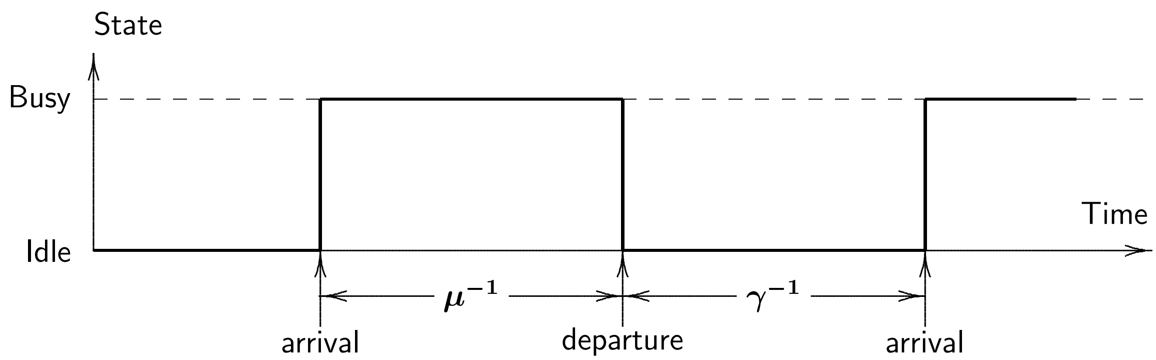


Figure 8.2: *Every individual source is either idle or busy, and behaves independent of all other sources.*

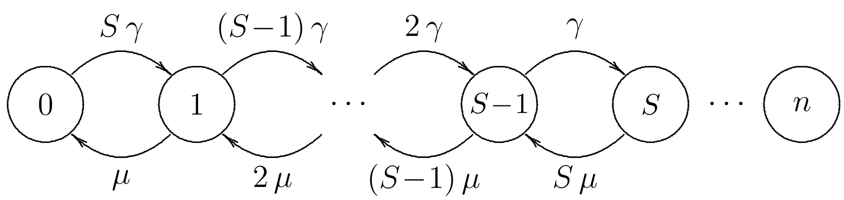


Figure 8.3: *State transition diagram for the Binomial case (Sec. 8.2). The number of sources S is less than the number of circuits n ($n \geq S$).*

8.2.1 Equilibrium equations

We are only interested in the steady state probabilities $p(i)$, which is the proportion of time the process is in state $[\,i\,]$. We base our calculations on the state transition diagram in Fig. 8.3. We consider cuts between neighbour states and find:

$$\begin{aligned}
S \cdot \gamma \cdot p(0) &= \mu \cdot p(1)\,, \\
(S-1) \cdot \gamma \cdot p(1) &= 2\mu \cdot p(2)\,, \\
\cdots & \quad \cdots \\
(S-i-1) \cdot \gamma \cdot p(i-1) &= i\,\mu \cdot p(i)\,, \\
(S-i) \cdot \gamma \cdot p(i) &= (i+1)\mu \cdot p(i+1)\,, \\
\cdots & \quad \cdots \\
1 \cdot \gamma \cdot p(S-1) &= S\mu \cdot p(S)\,.
\end{aligned} \tag{8.1}$$

All state probabilities are expressed by $p(0)$:

$$
\begin{array}{lclcl}
p(1) & = & \dfrac{S\,\gamma}{\mu} \cdot p(0) & = & p(0) \cdot \dbinom{S}{1} \cdot \left(\dfrac{\gamma}{\mu}\right)^{1} , \\[2ex]
p(2) & = & \dfrac{(S-1)\,\gamma}{2\,\mu} \cdot p(1) & = & p(0) \cdot \dbinom{S}{2} \cdot \left(\dfrac{\gamma}{\mu}\right)^{2} , \\[2ex]
\ldots & & \ldots & & \ldots \qquad \ldots \\[2ex]
p(i) & = & \dfrac{(S-i-1)\,\gamma}{i\,\mu} \cdot p(i-1) & = & p(0) \cdot \dbinom{S}{i} \cdot \left(\dfrac{\gamma}{\mu}\right)^{i} , \\[2ex]
p(i+1) & = & \dfrac{(S-i)\,\gamma}{(i+1)\,\mu} \cdot p(i) & = & p(0) \cdot \dbinom{S}{i+1} \cdot \left(\dfrac{\gamma}{\mu}\right)^{i+1} , \\[2ex]
\ldots & & \ldots & & \ldots \qquad \ldots \\[2ex]
p(S) & = & \dfrac{\gamma}{S\,\mu} \cdot p(S-1) & = & p(0) \cdot \dbinom{S}{S} \cdot \left(\dfrac{\gamma}{\mu}\right)^{S} .
\end{array}
$$

The total sum of all probabilities must be equal to one:

$$
\begin{aligned}
1 &= p(0) \cdot \left\{ 1 + \binom{S}{1} \cdot \left(\frac{\gamma}{\mu}\right) + \binom{S}{2} \cdot \left(\frac{\gamma}{\mu}\right)^{2} + \cdots + \binom{S}{S} \cdot \left(\frac{\gamma}{\mu}\right)^{S} \right\} \\
&= p(0) \cdot \left\{ 1 + \frac{\gamma}{\mu} \right\}^{S} ,
\end{aligned}
\tag{8.2}
$$

where we have used the Binomial expansion.

By letting $\beta = \gamma/\mu$ we get:

$$p(0) = \frac{1}{(1+\beta)^S} . \tag{8.3}$$

The parameter β is *the offered traffic per idle source* (number of call attempts per time unit for an idle source) (the offered traffic from a busy source is zero), and we find:

$$
\begin{aligned}
p(i) &= \binom{S}{i} \cdot \beta^{i} \cdot \frac{1}{(1+\beta)^S} \\
&= \binom{S}{i} \cdot \left(\frac{\beta}{1+\beta}\right)^{i} \cdot \left(\frac{1}{1+\beta}\right)^{S-i} ,
\end{aligned}
$$

$$p(i) = \binom{S}{i} \cdot \alpha^{i} \cdot (1-\alpha)^{S-i} , \quad i = 0, 1, \ldots, S , \quad 0 \le S \le n , \tag{8.4}$$

where

$$\alpha = \frac{\beta}{1+\beta} = \frac{\gamma}{\mu+\gamma} = \frac{1/\mu}{1/\gamma + 1/\mu} .$$

In this case, when a call attempt from an idle source never is blocked, the parameter α is equal to the carried traffic a per source ($a = \alpha$), which is equivalent to the probability that a source is busy at a random instant (the proportion of time the source is busy). This is also observed from Fig. 8.2, as all arrival and departure points on the time axes are regeneration points (equilibrium points). A cycle from start of a busy state (arrival) till start of the next busy state is representative for the whole time axes, and time averages are obtained by averaging over one cycle. Notice that for a systems with blocking we have $a \neq \alpha$ (cf. Sec. 8.3).

Formula (8.4) is the *Binomial distribution*. In teletraffic theory it is sometimes called the Bernoulli distribution, but this should be avoided as we in statistics use this name for a two-point distribution.

Formula (8.4) can be derived by elementary considerations. All subscribers can be split into two groups: idle subscribers and busy subscribers. The probability that an arbitrary subscriber belongs to class *busy* is $a = \alpha$, which is independent of the state of all other subscribers as the system has no blocking and call attempts always are accepted. There are in total S subscribers (sources) and the probability $p(i)$ that i sources are busy at an arbitrary instant is given by the Binomial distribution (8.4) & Tab. 6.1.

8.2.2 Traffic characteristics of Binomial traffic

We repeat the above mentioned definitions of the parameters:

$$\gamma \quad = \quad \text{call intensity per idle source,} \tag{8.5}$$

$$1/\mu \quad = \quad \text{mean service (holding) time,} \tag{8.6}$$

$$\beta = \gamma/\mu \quad = \quad \text{offered traffic per idle source.} \tag{8.7}$$

Offered traffic per idle source is a difficult concept to deal with as the proportion of time a source is idle depends on the congestion. The number of calls offered by a source becomes dependent of the number of channels (feed-back): a high congestion results in more idle time for a source and thus in more call attempts. By definition the offered traffic of a source is equal to the carried traffic in a system with no congestion, where the source freely changes between the states *idle* and *busy*. Therefore, we have the following definition of the offered traffic:

$$\alpha = \frac{\beta}{1+\beta} \quad = \quad \text{offered traffic per source,} \tag{8.8}$$

$$A = S \cdot \alpha \quad = \quad \text{total offered traffic,} \tag{8.9}$$

$$a \quad = \quad \text{carried traffic per source} \tag{8.10}$$

$$Y = S \cdot a \quad = \quad \text{total carried traffic.} \tag{8.11}$$

Time congestion:

$$\begin{aligned} E &= 0 && S < n\,, \\ E &= p(n) = \alpha^n && S = n\,. \end{aligned} \tag{8.12}$$

Carried traffic:

$$\begin{aligned} Y &= S \cdot a = \sum_{i=0}^{S} i \cdot p(i) \\ &= S \cdot \alpha = A\,, \end{aligned} \tag{8.13}$$

which is the mean value of the Binomial distribution (8.4). In this case with no blocking we therefore have $\alpha = a$.

Traffic congestion:

$$C = \frac{A - Y}{A} = 0\,. \tag{8.14}$$

Number of call attempts per time unit:

$$\begin{aligned} \Lambda &= \sum_{i=0}^{S} p(i) \cdot (S - i)\,\gamma \\ &= S\,\gamma - \gamma \cdot \sum_{i=0}^{S} i \cdot p(i) \\ &= S\,\gamma \cdot (1 - \alpha)\,, \\ \Lambda &= S\,\alpha\,\mu = S\,a\,\mu\,. \end{aligned} \tag{8.15}$$

The expression $(S\,a\,\mu)$ is the number calls carried per time unit, and thus we get:
Call congestion:

$$B = 0\,. \tag{8.16}$$

Traffic carried by channel ν:

Random hunting:

$$a_\nu = \frac{Y}{n} = \frac{S \cdot a}{n}\,. \tag{8.17}$$

Sequential hunting: complex expression derived by L.A. Joys (1971 [55]).

Improvement function:

$$F_n(A) = Y_{n+1} - Y_n = 0\,. \tag{8.18}$$

Peakedness (Tab. 6.1):

$$Z = \frac{\sigma^2}{\mu} = \frac{S \cdot \alpha \cdot (1-\alpha)}{S \cdot \alpha},$$

$$Z = 1-\alpha\,, \tag{8.19}$$

$$Z = \frac{1}{1+\beta} < 1\,. \tag{8.20}$$

We observe that the peakedness Z is independent of the number of sources and always less than one so that it corresponds to *smooth traffic*. As $A = \alpha\, S$, we therefore get the following simple relations between A, S and Z:

$$Z = 1 - \frac{A}{S}\,, \tag{8.21}$$

$$A = S \cdot (1-Z)\,, \tag{8.22}$$

$$S = \frac{A}{1-Z}\,. \tag{8.23}$$

Duration of state i: This is exponentially distributed with rate:

$$\gamma(i) = (S-i)\cdot\gamma + i\cdot\mu\,, \qquad 0 \le i \le S \le n\,. \tag{8.24}$$

8.3 Engset distribution

The only difference in comparison with Sec. 8.2 is that the number of sources S now is greater than or equal to the number of trunks (channels), $S \ge n$. Therefore, call attempts may experience congestion.

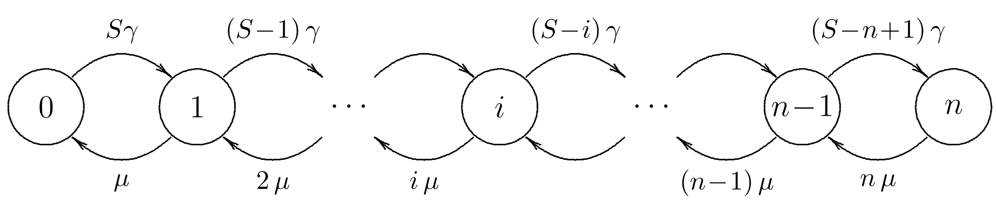


Figure 8.4: *State transition diagram for the Engset case with $S > n$, where S is the number of sources and n is the number of channels.*

8.3.1 State probabilities

The cut equations are identical to (8.1), but they only exist for $0 \leq i \leq n$ (Fig. 8.4). The normalisation equation (8.2) becomes:

$$1 = p(0) \cdot \left\{ 1 + \binom{S}{1} \cdot \left(\frac{\gamma}{\mu}\right) + \cdots + \binom{S}{n} \cdot \left(\frac{\gamma}{\mu}\right)^n \right\} .$$

Letting $\beta = \gamma/\mu$ the state probabilities become:

$$p(i) = \frac{\binom{S}{i} \cdot \beta^i}{\sum_{j=0}^{n} \binom{S}{j} \cdot \beta^j} . \tag{8.25}$$

In the same way as above we may by using (8.8) rewrite this expression to a form, which is analogue to (8.4):

$$p(i) = \frac{\binom{S}{i} \cdot \alpha^i \cdot (1-\alpha)^{S-i}}{\sum_{j=0}^{n} \binom{S}{j} \cdot \alpha^j \cdot (1-\alpha)^{S-j}} , \qquad 0 \leq i \leq n , \tag{8.26}$$

from which we directly observe why it is called a *truncated Binomial distribution* (cf. truncated Poisson distribution (7.10)).

The distribution (8.25) & (8.26) is called the *Engset–distribution*. T. Engset (1865–1943) was a Norwegian who first published the model with a finite number of sources (1918 [26]).

8.3.2 Traffic characteristics of Engset traffic

The Engset-distribution results in more complicated calculations than the Erlang loss system. The essential issue is to understand how to find the performance measures directly from the state probabilities using the definitions. The Engset system is characterised by the parameters $\beta = \gamma/\mu$ = offered traffic per idle source, S = number of sources, and n = number of channels.

Time congestion E: this is by definition equal to the proportion of time the system is blocked for new call attempts, i.e. $p(n)$ (8.25):

$$E_{n,S}(\beta) = p(n) = \frac{\binom{S}{n} \cdot \beta^n}{\sum_{j=0}^{n} \binom{S}{j} \cdot \beta^j} , \qquad S \geq n . \tag{8.27}$$

Call congestion B: this is by definition equal to the proportion of call attempts which are lost. Only call attempts arriving at the system in state n are blocked. During one unit of time we get the following ratio between the number of blocked call attempts and the total number of call attempts:

$$
\begin{aligned}
B_{n,S}(\beta) &= \frac{p(n)\cdot(S-n)\cdot\gamma}{\displaystyle\sum_{j=0}^{n} p(j)\cdot(S-j)\cdot\gamma} \\
&= \frac{\binom{S}{n}\cdot\beta^{n}\cdot(S-n)\cdot\gamma}{\displaystyle\sum_{j=0}^{n}\binom{S}{j}\cdot\beta^{j}\cdot(S-j)\cdot\gamma}
\end{aligned}
$$

Using

$$
\binom{S}{i}\cdot\frac{S-i}{S} = \binom{S-1}{i},
$$

we get:

$$
\begin{aligned}
B_{n,S}(\beta) &= \frac{\binom{S-1}{n}\cdot\beta^{n}}{\displaystyle\sum_{j=0}^{n}\binom{S-1}{j}\cdot\beta^{j}} \\
B_{n,S}(\beta) &= E_{n,S-1}(\beta)\,,\quad S\geq n\,.
\end{aligned}
\tag{8.28}
$$

This result may be interpreted as follows: The probability that a call attempt from a random source (subscriber) is rejected is equal to the probability that the remaining $S-1$ sources occupy all n channels. This is called *the arrival theorem*, and it can be shown to be valid for both loss and delay systems with a limited number of sources. The result is based on the product form among sources and the convolution of sources.
As E increases when S increases, we have $B_{n,S}(\beta)=E_{n,S-1}(\beta)<E_{n,S}(\beta)$.

Theorem 8.1 Arrival-theorem: *For all systems with a limited number of sources a random source upon arrival will observe the state of the system as if the source itself is not belonging to the system.*

Carried traffic: By applying the cut equation between state $i-1$ and state i we get:

$$\begin{aligned} Y &= \sum_{i=1}^{n} i \cdot p(i) && (8.29) \\ &= \sum_{i=1}^{n} \frac{\gamma}{\mu} \cdot (S-i+1) \cdot p(i-1) && (8.30) \\ &= \sum_{i=0}^{n-1} \beta \cdot (S-i) \cdot p(i) && (8.31) \\ &= \sum_{i=0}^{n} \beta \cdot (S-i) \cdot p(i) - \beta \cdot (S-n) \cdot p(n)\,, \\ Y &= \beta \cdot (S-Y) - \beta \cdot (S-n) \cdot E\,, && (8.32) \end{aligned}$$

as $E = E_{n,S}(\beta) = p(n)$. This is solved with respect to Y:

$$Y = \frac{\beta}{1+\beta} \cdot \{S - (S-n) \cdot E\}\ . \tag{8.33}$$

The offered traffic is given by (8.9), and we thus get as follows.

Traffic congestion $C = C_{n,S}(A)$. This is the most important congestion measure:

$$\begin{aligned} C &= \frac{A-Y}{A} \\ &= \frac{\dfrac{S\,\beta}{1+\beta} - \dfrac{\beta}{1+\beta} \cdot \{S-(S-n)\cdot E\}}{\dfrac{S\,\beta}{1+\beta}}\,, \\ C &= \frac{S-n}{S} \cdot E\,. && (8.34) \end{aligned}$$

This gives a simple relation between C and E.

We may also find the carried traffic if we know the call congestion B. The number of accepted call attempts per idle source per mean service time is $\gamma \cdot (1-B)$, and the carried traffic per source becomes:

$$a = \frac{\gamma\,(1-B)}{1+\gamma\,(1-B)}\,.$$

The total carried traffic becomes:

$$Y = S \cdot a = S \cdot \frac{\gamma\,(1-B)}{1+\gamma\,(1-B)}\,. \tag{8.35}$$

From (8.33) and (8.35) we get a simple relation between E and B which is identical to (8.53) & (8.54).

Number of call attempts per time unit:

$$\begin{aligned}\Lambda &= \sum_{i=0}^{n} p(i) \cdot (S-i) \cdot \gamma \\ &= S \cdot \gamma - Y\,\gamma \\ &= (S-Y) \cdot \gamma\,,\end{aligned}$$

where Y is the carried traffic. Thus $(S-Y)$ is the average number of idle sources.

Historically, the total offered traffic was earlier defined as Λ/μ. This is, however, misleading because we cannot assign every repeated call attempt a mean holding time $1/\mu$. Also it has caused a lot of confusion because the offered traffic by this definition depends upon the system (number of channels).

Lost traffic:

$$\begin{aligned}A_\ell &= A \cdot C \\ &= S\,\frac{\beta}{1+\beta} \cdot \frac{S-n}{S}\,E \\ &= \frac{(S-n)\,\beta}{1+\beta} \cdot E\,. \end{aligned} \tag{8.36}$$

Duration of state i: This is exponentially distributed with the intensity:

$$\gamma(i) = (S-i) \cdot \gamma + i \cdot \mu\,, \qquad 0 \le i < n\,, \tag{8.37}$$

$$\gamma(n) = n\,\mu\,, \qquad i = n\,. \tag{8.38}$$

Improvement function:

$$F_{n,S}(A) = Y_{n+1} - Y_n\,. \tag{8.39}$$

Example 8.3.1: Call average and time average
Above we have defined the state probabilities $p(i)$ under the assumption of statistical equilibrium as the proportion of time the system spends in state i, i.e. as a time average. We may also study how the state of the system looks when it is observed by an arriving or departing source (user) (call average). If we consider one time unit, then on the average $(S-i)\gamma \cdot p(i)$ sources will observe the system in state $[\,i\,]$ just before the arrival epoch, and if they are accepted they will bring the system into state $[\,i+1\,]$. Sources observing the system in state n are blocked and remain idle. Therefore, arriving sources observe the system in state $[\,i\,]$ with probability:

$$\pi_{n,S,\beta}(i) = \frac{(S-i)\,\gamma \cdot p(i)}{\sum_{j=0}^{n} (S-j)\,\gamma \cdot p(j)}\,, \quad i = 0, 1, \ldots n\,. \tag{8.40}$$

In a way analogue to the derivation of (8.28) we may show that in agreement with the arrival theorem (Theorem 8.1) we have as follows:

$$\pi_{n,S,\beta}(i) = p_{n,S-1,\beta}(i-1)\,, \quad i = 0, 1, \ldots, n\,. \tag{8.41}$$

When sources leaving the system looks back they observe the system in state $[\,i-1\,]$ with probability:

$$\psi_{n,S,\beta}(i-1) = \frac{i\,\mu \cdot p(i)}{\sum_{j=1}^{n} j\,\mu \cdot p(j)}\,, \quad i = 1, 2, \ldots, n\,. \tag{8.42}$$

By applying cut equations we immediately get that this is identical with (8.40), if we include the blocked customers. On the average, sources thus depart from the system in the same state as they arrive to the system. The process will be reversible and insensitive to the service time distribution. If we make a film of the system, then we are unable to determine whether it runs forward or backward. □

8.4 Evaluation of Engset's formula

If we try to calculate numerical values of Engset's formula directly from (8.27) (time congestion E), then we will experience numerical problems for large values of S and n. In the following we derive various numerically stable recursive formulæ for E and its reciprocal $I = 1/E$. When the time congestion E is known, it is easy to obtain the call congestion B and the traffic congestion C by using the formulæ (8.54) and (8.34). Numerically it is also simple to find any of the four parameters n, S, β, E when we know three of them. Mathematically we may assume that n and eventually S are non-integral.

8.4.1 Recursion formula on n

From the general formula (7.25) recursive in n we get using $\gamma_x = (S-x)\,\gamma$ and $\beta = \gamma/\mu$:

$$E_{x,S}(\beta) \;=\; \frac{\gamma_{x-1} \cdot E_{x-1,S}(\beta)}{x\,\mu + \gamma_{x-1} \cdot E_{x-1,S}(\beta)}\,, \tag{8.43}$$

$$E_{x,S}(\beta) \;=\; \frac{(S-x+1)\beta \cdot E_{x-1,S}(\beta)}{x + (S-x+1)\beta \cdot E_{x-1,S}(\beta)}\,, \quad E_{0,S}(\beta) = 1\,. \tag{8.44}$$

Introducing the reciprocal time congestion $I_{n,S}(\beta) = 1/E_{n,S}(\beta)$, we find the recursion formula:

$$I_{x,S}(\beta) = 1 + \frac{x}{(S-x+1)\,\beta}\, I_{x-1,S}(\beta)\,, \quad I_{0,S}(\beta) = 1\,. \tag{8.45}$$

Both (8.44) and (8.45) are analytically exact, numerically stable and accurate recursions for increasing values of n. However, for decreasing values of n the numerical errors accumulate and the recursions are not accurate. The number of iterations is n.

8.4.2 Recursion formula on S

If we denote the normalised state probabilities of a system with n channels and $S-1$ sources by $p_{n,S-1}(i)$, then we get the state probabilities for a system with S sources and n channels by convolving these state probabilities with the state probability of a single source

$\{p_{1,1}(0) = 1-\alpha,\ p_{1,1}(1) = \alpha\}$, truncate the state space at n, and normalise the state probabilities (cf. example 3.2.1) ($p(-1) = 0$):

$$q_{n,S}(i) = (1-\alpha)\, p_{n,S-1}(i) + \alpha\, p_{n,S-1}(i-1)\,, \quad i = 0, 1, \ldots, n\,. \tag{8.46}$$

The state probabilities $q_{n,S}(i)$ are not normalised, because we truncate at state $[\,n\,]$ and exclude the last term $q_{n,S}(n+1) = \alpha\, p_{n,S-1}(n)$ (state $[\,n+1\,]$). The normalised state probabilities $p_{n,S}(i)$ for a system with S sources and n channels are thus obtained from the normalised state probabilities $p_{n,S-1}(i)$ for a system with $S-1$ sources by:

$$p_{n,S}(i) = \frac{q_{n,S}(i)}{1-\alpha\, p_{n,S-1}(n)}\,, \quad i = 0, 1, \ldots, n. \tag{8.47}$$

The time congestion $E_{n,S}(\beta)$ for a system with S sources can be expressed by the time congestion $E_{n,S-1}(\beta)$ for a system with $S-1$ sources by using (8.47) and (8.46):

$$\begin{aligned} E_{n,S}(\beta) &= p_{n,S}(n) \qquad\qquad (8.48)\\ &= \frac{(1-\alpha)\, p_{n,S-1}(n) + \alpha\, p_{n,S-1}(n-1)}{1-\alpha\, p_{n,S-1}(n)} \\ &= \frac{(1-\alpha)\, E_{n,S-1}(\beta) + \alpha\, \dfrac{n\,\mu}{(S-n)\,\gamma}\, E_{n,S-1}(\beta)}{1-\alpha\, E_{n,S-1}(\beta)}\,, \end{aligned}$$

where we have used the balance equation between state $[n-1, S-1]$ and state $[n, S-1]$. Replacing α by using (8.8) we get:

$$E_{n,S}(\beta) = \frac{E_{n,S-1}(\beta) + \frac{n}{S-n}\, E_{n,S-1}(\beta)}{1+\beta-\beta\, E_{n,S-1}(\beta)}\,.$$

Thus we obtain the following recursive formula:

$$E_{n,S}(\beta) = \frac{S}{S-n} \cdot \frac{E_{n,S-1}(\beta)}{1+\beta\, \{1-E_{n,S-1}(\beta)\}}\,, \quad S > n\,, \quad E_{n,n}(\beta) = \alpha^n\,. \tag{8.49}$$

The initial value is obtained from (8.12). Using the reciprocal blocking probability $I = 1/E$ we get:

$$I_{n,S}(\beta) = \frac{S-n}{S\,(1-\alpha)} \cdot \{I_{n,S-1}(\beta) - \alpha\}\,, \quad S > n\,, \quad I_{n,n}(\beta) = \alpha^{-n}\,. \tag{8.50}$$

For increasing S the number of iterations is $S-n$. However, numerical errors accumulate due to the normalisation, where we divide with a number less than one, (8.47) and the applicability is limited. Therefore, it is recommended to use the recursion (8.52) given in the next section for increasing S.

For decreasing S the above formula is analytically exact, numerically stable, and accurate. However, the initial value should be known beforehand.

8.4.3 Recursion formula on both n and S

If we insert (8.44) into (8.49), respectively (8.45) into (8.50), we find:

$$E_{n,S}(\beta) = \frac{S\,\alpha \cdot E_{n-1,S-1}(\beta)}{n + (S-n)\alpha \cdot E_{n-1,S-1}(\beta)}\,, \qquad E_{0,S-n}(\beta) = 1\,, \tag{8.51}$$

$$I_{n,S}(\beta) = \frac{n}{S\,\alpha} \cdot I_{n-1,S-1}(\beta) + \frac{S-n}{S}\,, \qquad I_{0,S-n}(\beta) = 1\,, \tag{8.52}$$

which are recursive in both the number of servers and the number of sources. Both of these recursions are numerically accurate for increasing indices and the number of iterations is n (Joys, 1967 [53]).

From the above we have the following conclusions for recursion formulæ for the Engset formula. For increasing values of the parameter, recursion formulæ (8.44) & (8.45) are the best, and formulæ (8.51) & (8.52) are almost as good. Recursion formulæ (8.49) &(8.50) are numerically unstable for increasing values, but unlike the others stable for decreasing values. In general, we have that a recursion, which is stable in one direction, will be unstable in the opposite direction.

8.5 Relationships between E, B, and C

From (8.49) we get the following relation between $E = E_{n,S}(\beta)$ and $B = B_{n,S}(\beta) = E_{n,S-1}(\beta)$ using (8.28):

$$E = \frac{S}{S-n} \cdot \frac{B}{1+\beta(1-B)} \quad \text{or} \quad \frac{1}{E} = \frac{S-n}{S}\left\{(1+\beta)\cdot\frac{1}{B} - \beta\right\}\,, \tag{8.53}$$

$$B = \frac{(S-n)\cdot E\cdot(1+\beta)}{S+(S-n)\cdot E\cdot\beta} \quad \text{or} \quad \frac{1}{B} = \frac{1}{1+\beta}\left\{\frac{S}{S-n}\cdot\frac{1}{E} + \beta\right\}\,. \tag{8.54}$$

The expressions to the right-hand side are linear in the reciprocal blocking probabilities. In (8.34) we obtained the following simple relation between C and E:

$$C = \frac{S-n}{S} \cdot E\,, \tag{8.55}$$

$$E = \frac{S}{S-n} \cdot C\,. \tag{8.56}$$

If we in (8.55) express E by B (8.53), then we get C expressed by B:

$$C = \frac{B}{1+\beta\cdot(1-B)}\,, \tag{8.57}$$

$$B = \frac{(1+\beta)C}{1+\beta C}\,. \tag{8.58}$$

From (8.57) and (8.28) we get:

$$C_{n,S}(\beta) < B_{n,S}(\beta) < E_{n,S}(\beta)\,. \tag{8.59}$$

Example 8.5.1: Engset's loss system

We consider an Engset loss system having $n = 3$ channels and $S = 4$ sources. The call rate per idle source is $\gamma = 1/3$ calls per time unit, and the mean service time $(1/\mu)$ is 1 time unit. We find the following parameters:

$$\beta = \frac{\gamma}{\mu} = \frac{1}{3} \quad \text{erlang} \quad \text{(offered traffic per idle source)},$$

$$\alpha = \frac{\beta}{1+\beta} = \frac{1}{4} \quad \text{erlang} \quad \text{(offered traffic per source)},$$

$$A = S\cdot\alpha = 1 \quad \text{erlang} \quad \text{(offered traffic)},$$

$$Z = 1 - A/S = 3/4 \quad \text{(peakedness)}.$$

From the state transition diagram we obtain the following table:

i	$\gamma(i)$	$\mu(i)$	$q(i)$	$p(i)$	$i\cdot p(i)$	$\gamma(i)\cdot p(i)$
0	4/3	0	1.0000	0.3176	0.0000	0.4235
1	3/3	1	1.3333	0.4235	0.4235	0.4235
2	2/3	2	0.6667	0.2118	0.4235	0.1412
3	1/3	3	0.1481	0.0471	0.1412	0.0157
Total			3.1481	1.0000	0.9882	1.0039

We find the following blocking probabilities:

$$\begin{aligned}
\text{Time congestion:} \qquad & E_{3,4}(1/3) &=& \; p(3) = 0.0471\,, \\
\text{Traffic congestion:} \qquad & C_{3,4}(1/3) &=& \; \frac{A-Y}{A} = \frac{1-0.9882}{1} = 0.0118\,, \\
\text{Call congestion:} \qquad & B_{3,4}(1/3) &=& \; \{\gamma(3)\cdot p(3)\} \Bigg/ \left\{\sum_{i=0}^{3} \gamma(i)\cdot p(i)\right\} = \frac{0.0157}{1.0039} = 0.0156\;.
\end{aligned}$$

We notice that $E > B > C$, which is a general result for the Engset case (8.59) & (Fig. 8.5). By applying the recursion formula (8.45) we, of course, get the same results:

$$\begin{aligned}
E_{0,4}(1/3) &= 1\,, \\
E_{1,4}(1/3) &= \frac{(4-1+1)\cdot\frac{1}{3}\cdot 1}{1+(4-1+1)\cdot\frac{1}{3}\cdot 1} = \frac{4}{7}\,, \\
E_{2,4}(1/3) &= \frac{(4-2+1)\cdot\frac{1}{3}\cdot\frac{4}{7}}{2+(4-2+1)\cdot\frac{1}{3}\cdot\frac{4}{7}} = \frac{2}{9}\,, \\
E_{3,4}(1/3) &= \frac{(4-3+1)\cdot\frac{1}{3}\cdot\frac{2}{9}}{3+(4-3+1)\cdot\frac{1}{3}\cdot\frac{2}{9}} = \frac{4}{85} = 0.0471\,, \quad \text{q.e.d.}
\end{aligned}$$

□

Example 8.5.2: Limited number of sources
The influence from the limitation in the number of sources can be estimated by considering either the time congestion, the call congestion, or the traffic congestion. The congestion values are shown in Fig. 8.5 for a fixed number of channels n, a fixed offered traffic A, and an increasing value of the peakedness Z corresponding to a number of sources S, which is given by $S = A/(1-Z)$ (8.23). The offered traffic is defined as the traffic carried in a system without blocking ($n = \infty$). Here $Z = 1$ corresponds to a Poisson arrival process (Erlang's B-formula, $E = B = C$). For $Z < 1$ we get the Engset-case, and for this case the time congestion E is larger than the call congestion B, which is larger than the traffic congestion C. For $Z > 1$ we get the Pascal-case (Secs. 8.6 & 8.7 and Example 8.7.1). □

8.6 Pascal Distribution (Negative Binomial)

In the Binomial case the arrival intensity decreases linearly with an increasing number of busy sources. Palm & Wallström introduced a model where the arrival intensity increases linearly with the number of busy sources (servers)

(Wallström, 1964 [101]). The arrival intensity in state i is given by:

$$\gamma_i = \gamma \cdot (S+i), \quad 0 \leq i \leq n\,, \tag{8.60}$$

where γ and S are positive constants. The holding time are still assumed to be exponentially distributed with intensity μ.

In this section we assume the number of channels is infinite. We then set up a state transition diagram (cf. Fig. 8.6 with n infinite) and find the steady state probabilities which only exist for $\gamma < \mu$. We obtain:

$$p(i) = \binom{-S}{i} \cdot \left(-\frac{\gamma}{\mu}\right)^i \left(1-\frac{\gamma}{\mu}\right)^S , \quad 0 \leq i < \infty\,, \quad \gamma < \mu\,, \tag{8.61}$$

where

$$\binom{-S}{i} = (-1)^i \cdot \binom{S+i-1}{i}. \tag{8.62}$$

Formula (8.61) is the Negative Binomial distribution (cf. Pascal distribution, Tab. 6.1). The traffic characteristics of this model are obtained by an appropriate substitution of the parameters of the Binomial distribution. This is dealt with in the following section, which deals with a more realistic case.

8.7 Truncated Pascal distribution

We consider the same traffic process as in Sec. 8.6, but now we restrict the number of servers to a limited number n. The restriction $\gamma < \mu$ is superfluous as we always will obtain statistical equilibrium with a finite number of states. The state transition diagram is shown in Fig. 8.6, and state probabilities are given by:

$$p(i) = \frac{\binom{-S}{i}\left(-\frac{\gamma}{\mu}\right)^i}{\sum_{j=0}^{n}\binom{-S}{j}\left(-\frac{\gamma}{\mu}\right)^j}\,, \qquad 0 \leq i \leq n\,. \tag{8.63}$$

This is the truncated Negative Binomial (Pascal) distribution. Formally it is obtained from the Bernoulli/Engset case by the the following substitutions:

$$S \quad \text{is replaced by} \quad -S\,, \tag{8.64}$$

$$\gamma \quad \text{is replaced by} \quad -\gamma\,. \tag{8.65}$$

By these substitutions all formulae of the Bernoulli/Engset cases are valid for the truncated Pascal distribution.

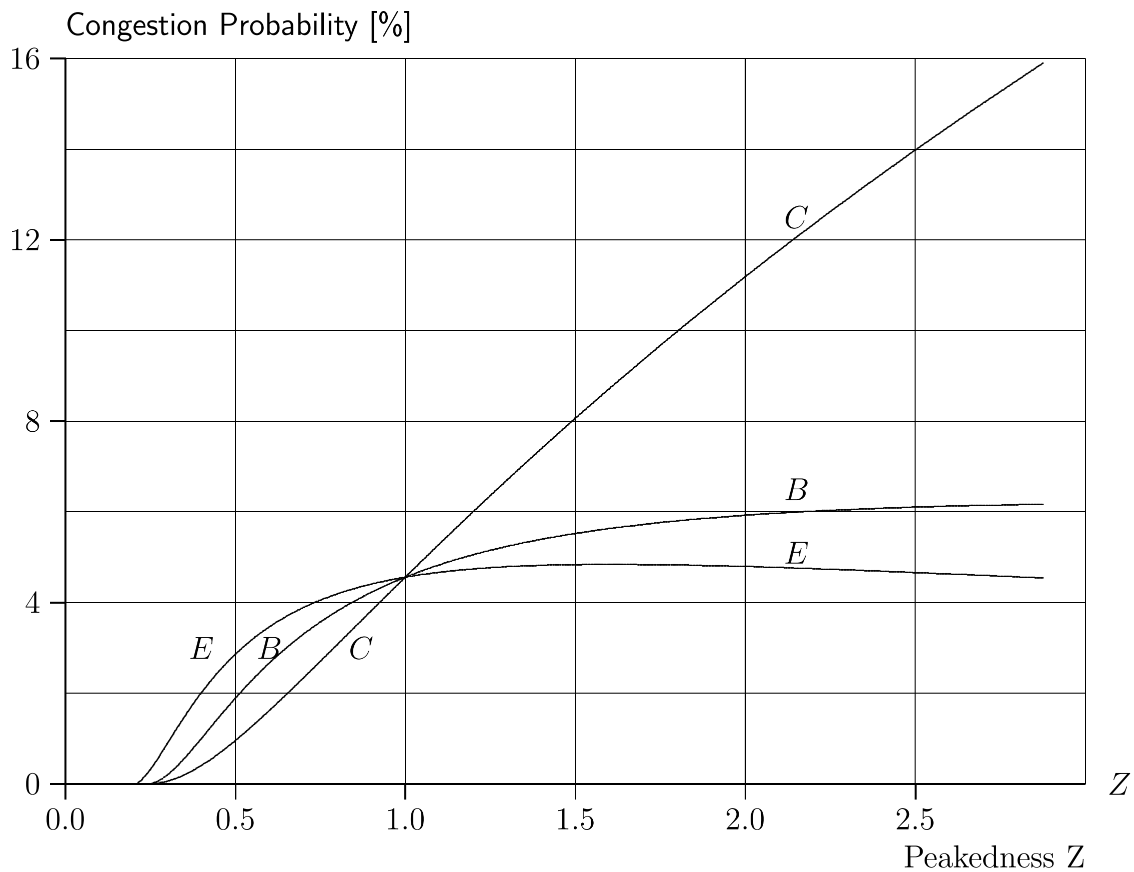


Figure 8.5: *Time congestion E, Call congestion B and Traffic congestion C as a function of peakedness Z for BPP–traffic i a system with $n = 20$ trunks and an offered traffic $A = 15$ erlang. More comments are given in Example 8.5.2 and Example 8.7.1. For applications the traffic congestion C is the most important, as it is almost a linear function of the peakedness.*

It can be shown that (8.63) is valid for arbitrary holding time distribution (Iversen, 1980 [38]). Assuming exponentially distributed holding times, this model has the same state probabilities as Palm's first normal form, i.e. a system with a Poisson arrival process having a stochastic intensity distributed as a gamma-distribution (inter-arrival times become *Pareto distributed*, i.e. heavy-tailed). The model is used for modelling overflow traffic which has a peakedness greater than one.

Example 8.7.1: Peakedness: numerical example
In Fig. 8.5 we keep the number of channels n and the offered traffic A fixed, and calculate the blocking probabilities for increasing peakedness Z. For $Z > 1$ we get the Pascal-case. For this case the time congestion E is less than the call congestion B which is less than the traffic congestion C. We observe that both the time congestion and the call congestion have a maximum value. Only the traffic congestion gives a reasonable description of the performance of the system. □

Example 8.7.2: Pascal loss system
We consider a Pascal loss system with $n = 4$ channels and $S = 2$ sources. The arrival rate is

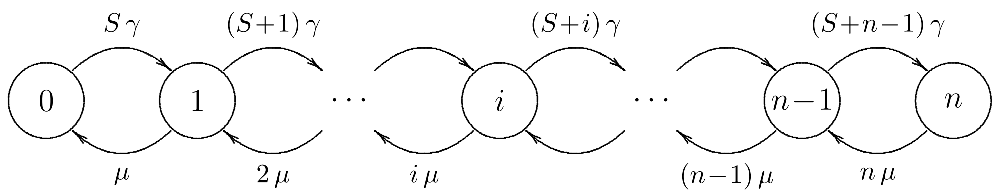


Figure 8.6: *State transition diagram for the Pascal (truncated Negative Binomial) case.*

$\gamma = 1/3$ calls/time unit per idle source, and the mean holding time $(1/\mu)$ is 1 time unit. We find the following parameters when we for the Engset case replace S by $-S$ (8.64) and γ by $-\gamma$ (8.65):

$$\begin{aligned}
\beta &= -\frac{1}{3}\,, \\
\alpha &= -\frac{1}{2}\,, \\
A &= -2 \cdot \left\{-\frac{1}{2}\right\} = 1 \quad \text{erlang}, \\
Z &= 1 - \frac{A}{S} = 1 - \frac{1}{-2} = \frac{3}{2}\,.
\end{aligned}$$

From a state transition diagram we get the following parameters:

i	$\gamma(i)$	$\mu(i)$	$q(i)$	$p(i)$	$i \cdot p(i)$	$\gamma(i) \cdot p(i)$
0	0.6667	0	1.0000	0.4525	0.0000	0.3017
1	1.0000	1	0.6667	0.3017	0.3017	0.3017
2	1.3333	2	0.3333	0.1508	0.3017	0.2011
3	1.6667	3	0.1481	0.0670	0.2011	0.1117
4	2.0000	4	0.0617	0.0279	0.1117	0.0559
Total			2.2099	1.0000	0.9162	0.9721

We find the following blocking probabilities:

Time congestion: $E_{4,-2}(-1/3) = p(4) = 0.0279\,.$

Traffic congestion: $C_{4,-2}(-1/3) = \dfrac{A-Y}{A} = \dfrac{1-0.9162}{1} = 0.0838\,.$

Call congestion: $B_{4,-2}(-1/3) = \{\gamma(4)\cdot p(4)\} \Bigg/ \left\{\sum_{i=0}^{4} \gamma(i)\cdot p(i)\right\} = \dfrac{0.0559}{0.9721} = 0.0575\,.$

We notice that $E < B < C$, which is a general result for the Pascal case. By using the same recursion formula as for the Engset case (8.44), we of course get the same results:

$$
\begin{aligned}
E_{0,-2}(-1/3) &= 1.0000\,, \\
E_{1,-2}(-1/3) &= \frac{\frac{2}{3}\cdot 1}{1+\frac{2}{3}\cdot 1} = \frac{2}{5}\,, \\
E_{2,-2}(-1/3) &= \frac{\frac{3}{3}\cdot \frac{2}{5}}{2+\frac{3}{3}\cdot \frac{2}{5}} = \frac{1}{6}\,, \\
E_{3,-2}(-1/3) &= \frac{\frac{4}{3}\cdot \frac{1}{6}}{3+\frac{4}{3}\cdot \frac{1}{6}} = \frac{2}{29}\,, \\
I_{4,-2}(-1/3) &= \frac{\frac{5}{3}\cdot \frac{2}{29}}{4+\frac{5}{3}\cdot \frac{2}{29}} = \frac{5}{179} = 0.0279 \quad \text{q.e.d.}
\end{aligned}
$$

□

Chapter 9

Overflow theory

In this chapter we consider systems with restricted (limited) accessibility, i.e. systems where a subscriber or a traffic flow only has access to k specific channels from a total of n ($k \leq n$). If all k channels are busy, then a call attempt is blocked even if there are idle channels among the remaining $(n-k)$ channels. An example is shown in Fig. 9.1, where we consider a hierarchical network with traffic from A to B, and from A to C. From A to B there is a direct (primary) route with n_1 channels. If they all are busy, then the call is directed to the alternative (secondary) route via T to B. In a similar way, the traffic from A to C has a first-choice route AC and an alternative route ATC. If we assume the routes TB and TC are without blocking, then we get the accessibility scheme shown to the right in Fig. 9.1. From this we notice that the total number of channels is $(n_1 + n_2 + n_{12})$ and that the traffic AB only has access to (n_1+n_{12}) of these. In this case sequential hunting among the routes should be applied so that a call only is routed via the group n_{12}, when all n_1 primary channels are busy.

It is typical for a hierarchical network that it possesses a certain *service protection*. Independent of how high the traffic from A to C is, then it will never get access to the n_1 channels. On the other hand, we may block calls even if there are idle channels, and therefore the utilisation will always be lower than for systems with full accessibility. The utilisation will, however, be bigger than for separate systems with the same total number of channels. The common channels allows for a certain traffic balancing between the two groups.

Historically, it was necessary to consider restricted accessibility because the electro-mechanical systems had very limited intelligence and limited selector capacity (accessibility). In digital systems we do not have these restrictions, but still the theory of restricted accessibility is important both in networks and in guaranteeing the grade-of-service.

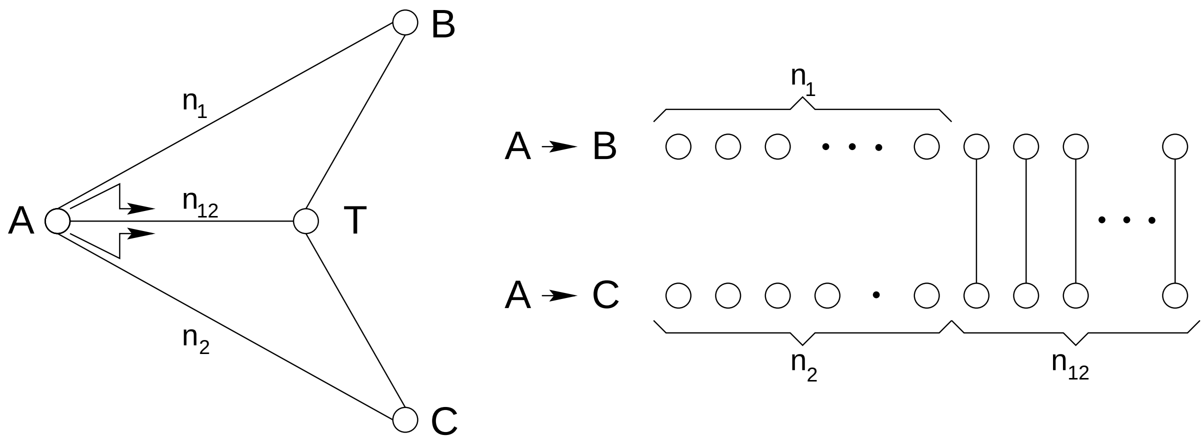


Figure 9.1: *Telecommunication network with alternate routing and the corresponding accessibility scheme, which is called an O'Dell–grading. We assume the links between the transit exchange T and the exchanges B and C are without blocking. The n_{12} channels are common for both traffic streams.*

9.1 Overflow theory

The classical traffic models assume that the traffic offered to a system is pure chance traffic type one or two, *PCT–I or PCT–II*. In communication networks with alternative traffic routing, the traffic which is lost from the primary group is offered to an overflow group, and it has properties different from *PCT* traffic (Sec. 6.4). Therefore, we cannot use the classical models for evaluating blocking probabilities of overflow traffic.

Example 9.1.1: Group divided into two
Let us consider a group with 16 channels which is offered 10 erlang *PCT–I* traffic. By using Erlang's B–formula we find the blocking probability E = 2.23% and the lost traffic 0.2230 erlang.
We now assume sequential hunting and split the 16 channels into a primary group and an overflow group, each of 8 channels. By using Erlang's B–formula we find the overflow traffic from the primary group equal to 3.3832 erlang. This traffic is offered to the overflow group. Using Erlang's B–formula again, we find the lost traffic from the overflow group: $A_\ell = 3.3832 \cdot E_8(3.3832) = 0.0493$ [erlang]. The total blocking probability in this way becomes 0.493%, which is much less than the correct result 2.23%. We have *made an error* by applying the B–formula to the overflow traffic, which is *not PCT–I* traffic, but more bursty. □

In the following we describe two classes of models for overflow traffic. We can in principle study the traffic process either vertically or horizontally. By vertical studies we calculate the state probabilities (Sec. 9.1.1–9.4.3). By horizontal studies we analyse the distance between call arrivals, i.e. the inter-arrival time distribution (9.5).

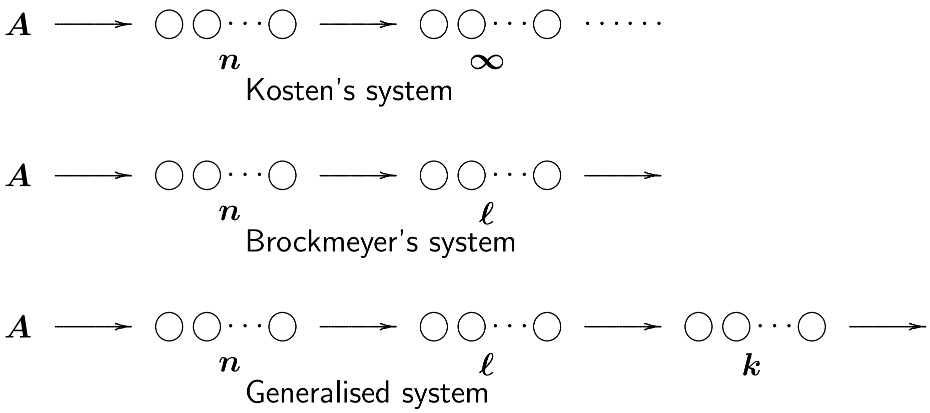


Figure 9.2: *Different overflow systems described in the literature.*

9.1.1 State probability of overflow systems

Let us consider a full accessible group with ordered (sequential) hunting. The group is split into a limited primary group with n channels and an overflow group with infinite capacity. The offered traffic A is assumed to be *PCT–I*. This is called *Kosten's system* (Fig. 9.2). The state of the system is described by a two-dimensional vector:

$$p(i,j), \qquad 0 \leq i \leq n, \quad 0 \leq j \leq \infty\,, \tag{9.1}$$

which is the probability that at a random point of time i channels are occupied in the primary group and j channels in the overflow group. The state transition diagram is shown in Fig. 9.3. Kosten (1937 [67]) analysed this model and derived the marginal state probabilities:

$$p(i,\cdot\,) \;=\; \sum_{j=0}^{\infty} p(i,j), \qquad 0 \leq i \leq n\,, \tag{9.2}$$

$$p(\,\cdot\,,j) \;=\; \sum_{i=0}^{n} p(i,j), \qquad 0 \leq j < \infty\,. \tag{9.3}$$

Riordan (1956 [88]) derived the moments of the marginal distributions. Mean value and peakedness (variance/mean ratio) of the marginal distributions, i.e. the traffic carried by the two groups, become:

Primary group:

$$m \;=\; A\cdot\{1-E_n(A)\}\;, \tag{9.4}$$

$$\frac{v}{m} = Z \;=\; 1 - A\cdot\{E_{n-1}(A)-E_n(A)\}\;, \tag{9.5}$$

$$Z \;=\; 1 - F_{n-1}(A) = 1 - a_n \leq 1\,.$$

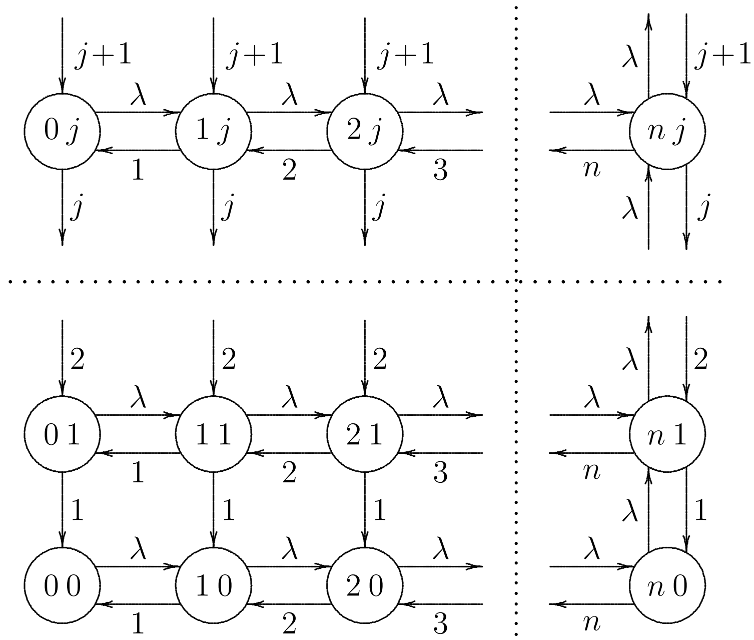


Figure 9.3: *State transition diagram for Kosten's system, which has a primary group with n channels and an unlimited overflow group. The states are denoted by (i, j), where i is the number of busy channels in the primary group, and j is the number of busy channels in the overflow group. The mean holding time is chosen as time unit.*

where $F_{n-1}(A)$ is the improvement function of Erlang's B-formula.

Secondary group = Overflow group:

$$m = A \cdot E_n(A)\,, \tag{9.6}$$

$$\frac{v}{m} = Z = 1 - m + \frac{A}{n+1-A+m} \geq 1\,. \tag{9.7}$$

Experience shows that the peakedness Z is a good measure for the relative blocking probability a traffic stream with a given mean value is subject to. In Fig. 9.4 we notice that the peakedness of overflow traffic has a maximum for a fixed traffic and an increasing number of channels. Peakedness has the dimension [channels]. Peakedness is applicable for theoretical calculations, but difficult to estimate accurately from observations.

For *PCT–I* traffic the peakedness is equal to one, and the blocking is calculated by using the Erlang-B formula. If the peakedness is less than one (9.5), the traffic is called *smooth* and it experiences less blocking than *PCT–I* traffic. If the peakedness is larger than one, then the traffic is called *bursty* and it experiences larger blocking than *PCT–I* traffic. Overflow traffic is usually bursty (9.7).

Brockmeyer (1954 [10]) derived the state probabilities and moments of a system with a limited overflow group (Fig. 9.2), which is called *Brockmeyer's system*. Bech (1954 [6]) did the same by using matrix equations, and obtained more complicated and more general expressions. Brockmeyer's system is further generalised by Schehrer who also derived higher order moments for finite overflow groups.

Wallström (1966 [102]) calculated state probabilities and moments for overflow traffic of a generalised Kosten system, where the arrival intensity depends either upon the total number of calls in the system or the number of calls in the primary group.

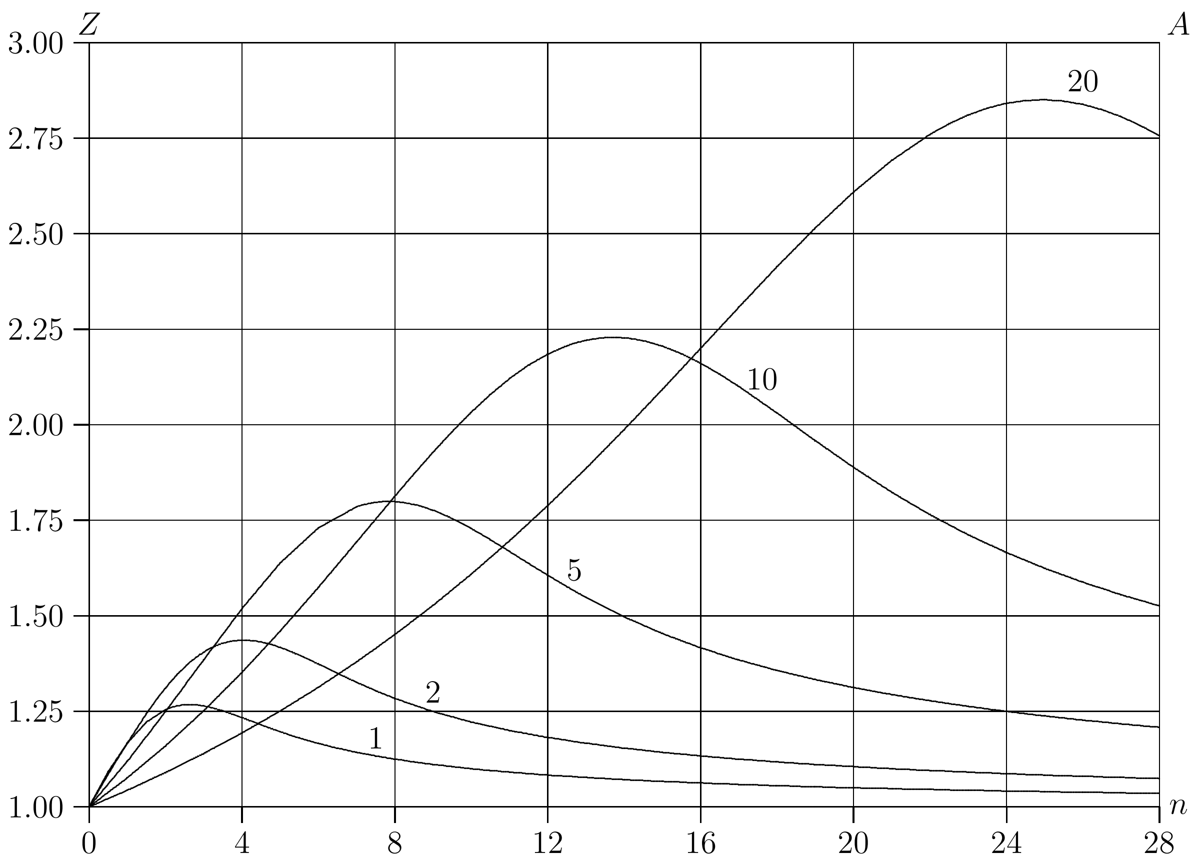


Figure 9.4: *Peakedness Z of overflow traffic as a function of number of channels for a fixed value of the offered traffic. Notice that Z has a maximum. When n becomes large call attempts are seldom blocked and the blocked attempts will be mutually independent. Therefore, the process of overflowing calls converges to a Poisson process (Chap. 6).*

9.2 Equivalent Random Traffic method

This equivalent method is also called the *ERT–method*, *Wilkinson's method* or *Wilkinson-Bretschneider's method*. It was published independently at the same time in USA by Wilkinson (1956 [103]) and in Germany by Bretschneider (1956 [8]). It plays a key role when dimensioning telecommunication networks.

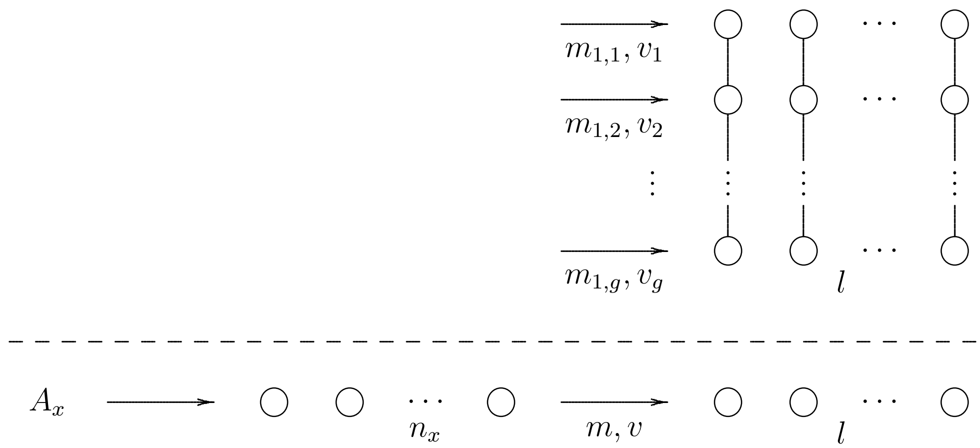


Figure 9.5: *Application of the ERT-method to a system having g independent traffic streams offered to a common group of ℓ channels. The aggregated overflow process of the g traffic streams is said to be equivalent to the traffic overflowing from a full accessible group with the same mean and variance of the overflow traffic. (9.8) & (9.9).*

9.2.1 Preliminary analysis

Let us consider a group with ℓ channels which is offered g traffic streams (Fig. 9.5). The traffic streams may for instance be traffic which is offered from other exchanges to a transit exchange, and therefore they cannot be described by classical traffic models. Thus we do not know the distributions (state probabilities) of the traffic streams, but we are satisfied (as it is often the case in applications of statistics) by characterising the i'th traffic stream by its mean value $m_{1,i}$ and variance v_i. With this simplification *we will consider two traffic streams to be equivalent, if they have same mean value and variance*.

The total traffic offered to the group with ℓ channels has the mean value:

$$m = \sum_{i=1}^{g} m_{1,i} \; . \tag{9.8}$$

We assume that the traffic streams are independent (non-correlated), and thus the variance of the total traffic stream becomes:

$$v = \sum_{i=1}^{g} v_i \; . \tag{9.9}$$

The total traffic is characterised by m and v. So far we assume that $m < v$. We now consider this traffic to be equivalent to a traffic flow, which is lost from a full accessible group and has same mean value m and variance v. In Fig. 9.5 the upper system is replaced by the *equivalent system* at the lower part of Fig. 9.5, which is a full accessible system with $(n_x + \ell)$ channels offered the traffic A_x. For given values of m and v we therefore solve equations (9.6) and (9.7) with respect to n and A. It can be shown there that a unique solution exists, and it will be denoted by (n_x, A_x).

The lost traffic is found from the Erlang's B-formula:

$$A_\ell = A_x \cdot E_{n_x+\ell}(A_x) \, . \tag{9.10}$$

As the offered traffic is m, the traffic congestion of the system becomes:

$$C = \frac{A_\ell}{m} \, . \tag{9.11}$$

Notice: the blocking probability is *not* $E_{n_x+\ell}(A_x)$. We should remember the last step (9.11), where we relate the lost traffic to the originally offered traffic, which in this case is given by m (9.8).

We notice that if the overflow traffic is from a single primary group with *PCT–I* traffic, then the method is exact. In the general case with more traffic streams the method is approximate, and it does not yield the exact mean blocking probability.

Example 9.2.1: Paradox

In Sec. 6.3 we derived Palm's theorem, which states that by superposition of many independent arrival processes, we *locally* get a Poisson process. This is *not* contradictory with (9.8) and (9.9), because these formulæ are valid *globally*. □

9.2.2 Numerical aspects

When applying the *ERT*–method we need to calculate (m, v) for given values of (A, n) and vice versa. It is easy to obtain (m, v) for given (A, n) by using (9.4) & (9.5). To obtain (A, n) for given (m, v), we have to solve two equations with two unknown. It requires an iterative procedure, since $E_n(A)$ cannot be solved explicitly with respect to neither n nor A (Sec. 7.5). However, we can solve (9.7) with respect to n:

$$n = A \cdot \frac{m + \frac{v}{m}}{m + \frac{v}{m} - 1} - m - 1 \, , \tag{9.12}$$

so that we know n for given A. Thus A is the only independent variable. We can use Newton-Raphson's iteration method to solve the remaining equation, introducing the function:

$$f(A) = m - A \cdot E_n(A) = 0 \, .$$

For a proper starting value A_0 we improve this iteratively until the resulting values of m and v/m become close enough to the known values.

Yngvé Rapp (1965 [87]) has proposed a good approximate solution for A, which can be used as initial value A_0 in the iteration:

$$A \approx v + 3 \cdot \frac{v}{m} \cdot \left\{ \frac{v}{m} - 1 \right\} \, . \tag{9.13}$$

From A we get n, using (9.12). *Rapp's approximation* is sufficient accurate for practical applications, except when A_x is very small. The peakedness $Z = v/m$ has a maximum, obtained when n is little larger than A (Fig. 9.4). For some combinations of m and v/m the convergence is critical, but when using computers we can always find the correct solution.

Using computers we operate with non-integral number of channels, and only at the end of calculations we choose an integral number of channels greater than or equal to the obtained results (typical a module of a certain number of channels (8 in *GSM*, 30 in *PCM*, etc.). When using tables of Erlang's B–formula, we should in every step choose the number of channels in a conservative way so that the blocking probability aimed at becomes worst case.

The above-mentioned method presupposes that v/m is larger than one, and so it is only valid for bursty traffic. Individual traffic stream in Fig. 9.5 are allowed to have $v_i/m_i < 1$, provided the total aggregated traffic stream is bursty. Bretschneider ([9], 1973) has extended the method to include a negative number of channels during the calculations. In this way it is possible to deal with smooth traffic (*EERT-method* = *Extended ERT method*).

9.2.3 Parcel blocking probabilities

The individual traffic streams in Fig. 9.5 do not have the same mean value and variance, and therefore they do not experience equal blocking probabilities in the common overflow group with ℓ channels. From the above we calculate the mean blocking (9.11) for all traffic streams aggregated. Experiences show that the blocking probability experienced is proportional to the peakedness $Z = v/m$. We can split the total lost traffic into individual lost traffic parcels by assuming that the traffic lost for stream i is proportional to the mean value m_i and to the peakedness $Z_i = v_i/m_i$ of the stream. We obtain:

$$\begin{aligned} A_\ell &= \sum_{i=1}^{g} A_{\ell,i} \\ &= c \cdot A_\ell \cdot \sum_{i=1}^{g} m_{1,i} \cdot \frac{v_i}{m_{1,i}} \\ &= c \cdot A_\ell \cdot v\,, \end{aligned}$$

from which we find the constant $c = 1/v$.

The (traffic) blocking probability for traffic stream i, which is called the *parcel blocking probability* for stream i, then becomes:

$$C_i = \frac{A_{\ell,i}}{m_i} = \frac{v_i}{v} \cdot A_\ell\,. \tag{9.14}$$

Furthermore, we can divide the blocking among the individual groups (primary, secondary, etc.). Consider the equivalent group at the bottom of Fig. 9.5 with n_x primary channels and ℓ secondary (overflow) channels, we can calculate the blocking probability due to the n_x primary channels, and the blocking probability due to the ℓ secondary channels. The probability that the traffic is lost by the ℓ channels is equal to the probability that the traffic is lost by the $n_x + \ell$ channels, under the condition that the traffic is offered to the ℓ channels:

$$H(l) = \frac{A \cdot E_{n_x+l}(A)}{A \cdot E_{n_x}(A)} = \frac{E_{n_x+l}(A)}{E_{n_x}(A)} \,. \tag{9.15}$$

The total loss probability can therefore be related to the two groups:

$$E_{n_x+l}(A) = E_{n_x}(A) \cdot \frac{E_{n_x+l}(A)}{E_{n_x}(A)} \,. \tag{9.16}$$

By using this expression, we can find the blocking for each channel group and then for example obtain information about which group should be increased by adding more channels.

Example 9.2.2: Example 9.1.1 continued
In example 9.1.1 the blocking probability of the primary group of 8 channels is $E_8(10) = 0.3383$. The blocking of the overflow group is

$$H(8) = \frac{E_{16}(10)}{E_8(10)} = \frac{0.02231}{0.3383} = 0.06592 \,.$$

The total blocking of the system is:

$$E_{16}(10) = E_8(10) \cdot H(8) = 0.3383 \cdot 0.06592 = 0.02231 \,.$$

□

Example 9.2.3: Hierarchical cellular system
We consider a cellular system *HCS* covering three areas. The traffic offered in the areas are 12, 8 and 4 erlang, respectively. In the first two cells we introduce micro-cells with 16, respectively 8 channels, and a common macro-cell covering all three areas is allocated 8 channels. We allow overflow from micro-cells to macro-cells, but do not rearrange the calls from macro- to micro-cells when a channel becomes idle. Furthermore, we look away from hand over traffic. Using (9.6) & (9.7) we find the mean value and the variance of the traffic offered to the macro-cell:

Cell	Offered traffic	Number of channels	Overflow mean	Overflow variance	Peakedness
i	A_i	$n_i(j)$	$m_{1,i}$	v_i	Z_i
1	12	16	0.7250	1.7190	2.3711
2	8	8	1.8846	3.5596	1.8888
3	4	0	4.0000	4.0000	1.0000
Total	24		6.6095	9.2786	1.4038

The total traffic offered to the macro-cell has mean value 6.61 erlang and variance 9.28. This corresponds to the overflow traffic from an equivalent system with 10.78 erlang offered to 4.72 channels. Thus we end up with a system of 12.72 channels offered 10.78 erlang. Using the Erlang-B formula, we find the lost traffic 1.3049 erlang. Originally we offered 24 erlang, so the total traffic blocking probability becomes $B = 5.437\%$.

The three areas have individual blocking probabilities. Using (9.14) we find the approximate lost traffic from the areas to be 0.2434 erlang, 0.5042 erlang, and 0.5664 erlang, respectively. Thus the traffic blocking probabilities become 2.03%, 6.30% and 14.16%, respectively. A computer simulation with 100 million calls yields the blocking probabilities 1.77%, 5.72%, and 15.05%, respectively. This corresponds to a total lost traffic equal to 1.273 erlang and a blocking probability 5.30%. The accuracy of the method of this chapter is sufficient for real applications. □

9.3 Fredericks & Hayward's method

Fredericks (1980 [29]) has proposed an equivalence method which is simpler to use than Wilkinson-Bretschneider's method. The motivation for the method was first put forward by W.S. Hayward.

Fredericks & Hayward's equivalence method also characterises the traffic by mean value A and peakedness Z $(0 < Z < \infty)$ ($Z = 0$ is a trivial case with constant traffic). The peakedness (7.7) is the ratio between the variance v and the mean value m_1 of the state probabilities, and it has the dimension [channels]. For random traffic (*PCT–I*) we have $Z = 1$ and we can apply the Erlang-B formula.

For peakedness $Z \neq 1$ Fredericks & Hayward's method proposes that the system has the same blocking probability as a system with n/Z channels, offered traffic A/Z, and peakedness $Z = 1$. For the latter system we may apply the Erlang–B formula:

$$E(n, A, Z) \sim E\left(\frac{n}{Z},\ \frac{A}{Z},\ 1\right) \sim E_{\frac{n}{Z}}\left(\frac{A}{Z}\right) . \tag{9.17}$$

When $Z = 1$ we assume the traffic is *PCT-I* and apply Erlang's B–formula for calculating the congestion. It is *the traffic congestion* we obtain when using the method (cf. Sec. 9.3.1). For fixed value of the blocking in the Erlang-B formula we know (Fig. 7.4) that the utilisation increases, when the number of channels increases: the larger the system, the higher utilisation for a fixed blocking probability. Fredericks & Hayward's method thus expresses that if the traffic has a larger peakedness Z than *PCT–I* traffic, then we get a lower utilisation than the one obtained by using Erlang's B–formula. If peakedness $Z < 1$, then we get a better utilisation.

We avoid solving the equations (9.6) and (9.7) with respect to (A, n) for given values of (m, v). The method can easily be applied for both peaked and smooth traffic. In general we get an non-integral number of channels and thus need to evaluate the Erlang-B formula for a continuous number of channels.

Basharin & Kurenkov has extended the method to comprise multi-slot (multi-rate) traffic, where a call requires d channels from start to termination. If a call uses d channels in stead of one (change of scale), then the mean value becomes d times bigger and the variance d^2 times bigger. Therefore, the peakedness becomes d times bigger. In stead of reducing number of channels by the factor Z, we may fix the number of channels and make the slot-size Z times bigger:

$$(n,\ A,\ Z,\ d) \sim \left(n,\ \frac{A}{Z},\ 1,\ d \cdot Z\right) \sim \left(\frac{n}{Z},\ \frac{A}{Z},\ 1,\ d\right) . \tag{9.18}$$

If we have more traffic streams offered to the same group, then it may be an advantage to keep the number of channels fixed, but then we get the problem that $c \cdot Z$ in general will not be integral.

Example 9.3.1: Fredericks & Hayward's method
If we apply Fredericks & Hayward's method to example 9.2.3, then the macro-cell has (8/1.4038) channels and is offered (6.6095/1.4038) erlang. The blocking probability is obtained from Erlang's B-formula and becomes 0.19470. The lost traffic is calculated from the *original* offered traffic (6.6095 erlang) and becomes 1.2871 erlang. The blocking probability of the system becomes $E = 1.2871/24$ = 5.36%. This is very close to the result obtained (5.44%) by the *ERT*–method. □

Example 9.3.2: Multi-slot traffic
We shall later consider service-integrated system with multi-rate (multi-slot) traffic. In example 10.4.3 we consider a trunk group with 1536 channels, which is offered 24 traffic streams with individual slot-size and peakedness. The exact total traffic congestion is equal to 5.950%. If we calculate the peakedness of the offered traffic by adding all traffic streams, then we find peakedness $Z = 9.8125$ and a total mean value equal to 1536 erlang. Fredericks & Hayward's method results in a total traffic congestion equal to 6.114%, which thus is a conservative estimate (worst case). □

9.3.1 Traffic splitting

In the following we shall give a natural interpretation of Fredericks & Hayward's method and at the same time discuss splitting of traffic streams. We consider a traffic stream with mean value A, variance v, and peakedness $Z = v/A$. We split this traffic stream into g identical sub-streams. A single sub-stream then has the mean value A/g and peakedness Z/g because the mean value is reduced by a factor g and the variance by a factor g^2 (Example 3.3.2). If we choose the number of sub-streams g equal to Z, then we get the peakedness $Z=1$ for each sub-stream.

Let us assume the original traffic stream is offered to n channels. If we also split the n channels into g sub-group (one for each sub-stream), then each subgroup has n/g channels.

Each sub-group will then have the same blocking probability as the original total system. By choosing $g = Z$ we get peakedness $Z = 1$ in the sub-streams, and we may (approximately) use Erlang's B–formula for calculating the blocking probability.

This is a natural interpretation of Fredericks & Hayward's method. It can easily be extended to comprise multi–slot traffic. If every call requires d channels during the whole connection time, then by splitting the traffic into d sub-streams each call will use a single channel in each of the d sub-groups, and we will get d identical systems with single-slot traffic.

The above splitting of the traffic into g identical traffic streams shows that the blocking probability obtained by Fredericks-Hayward's method is *the traffic congestion*. The equal splitting of the traffic at any point of time implies that all g traffic streams are identical and thus have the mutual correlation one. In reality, we cannot split circuit switched traffic into identical sub-streams. If we have $g = 2$ streams and three channels are busy at a given point of time, then we will for example use two channels in one sub-stream and one in the other, but anyway we obtain the same optimal utilisation as in the total system, because we always will have access to an idle channel in any sub-group (full accessibility). The correlation between the sub-streams becomes smaller than one. The above is an example of using more intelligent strategies so that we maintain the optimal full accessibility.

In Sec. 6.3.2 we studied the splitting of the arrival process when the splitting is done in a random way (Raikov's theorem 6.2). By this splitting we did not reduce the variation of the process when the process is a Poisson process or more regular. The resulting sub-streams converge to Poisson processes. In this section we have considered the splitting of the traffic process, which includes both the arrival process and the holding times. The splitting process depends upon the state. In a sub-process, a long holding time of a single call will result in fewer new calls in this sub-process during the following time interval, and the arrival process will no longer be a renewal process.

Most attempts of improving Fredericks & Hayward's equivalence method are based on reducing the correlation between the sub-streams, because the arrival processes for a single sub-stream is considered as a renewal process, and the holding times are assumed to be exponentially distributed. From the above we see that these approaches are deemed to be unsuccessful, because they will not result in an optimal traffic splitting. In the following example we shall see that the optimal splitting can be implemented for packet switched traffic with constant packet size.

Example 9.3.3: Inverse multiplexing
If we need more capacity in a network than what corresponds to a single channel, then we may combine more channels in parallel. At the originating source we may then distribute the traffic (packets or cells in ATM) in a cyclic way over the individual channels, and at the destination we reconstruct the original information. In this way we get access to higher bandwidth without leasing fixed broadband channels, which are very expensive. If the traffic parcels are of constant size, then the traffic process is split into a number of identical traffic streams, so that we get the same utilisation as in a single system with the total capacity. This principle was first exploited in a Danish

equipment (Johansen & Johansen & Rasmussen, 1991 [52]) for combining up to 30 individual 64 Kbps ISDN connections for transfer of video traffic for maintenance of aircrafts.
Today, similar equipment is applied for combining a number of 2 Mbps connections to be used by ATM-connections with larger bandwidth (*IMA* = Inverse Multiplexing for ATM) (Techguide, 2001 [97]), (Postigo–Boix & García-Haro & Aguilar-Igartua, 2001 [84]). □

9.4 Other methods based on state space

From a blocking point of the view, the mean value and variance do not necessarily characterise the traffic in the optimal way. Other parameters may better describe the traffic. When calculating the blocking with the *ERT*–method we have two equations with two unknown variables (9.6 & 9.7). The Erlang loss system is uniquely defined by the number of channels and the offered traffic A_x. Therefore, it is not possible to generalise the method to take account of more than two moments (mean & variance).

9.4.1 BPP traffic models

The *BPP*–traffic models describe the traffic by two parameters, mean value and peakedness, and are thus natural candidates to model traffic with two parameters. Historically, however, the concept and definition of traffic congestion has due to earlier definitions of offered traffic been confused with call congestion. As seen from Fig. 8.5 only the traffic congestion makes sense for overflow calculations. By proper application of the traffic congestion, the *BPP*–model is very applicable.

Example 9.4.1: BPP traffic model
If we apply the BPP–model to the overflow traffic in example 9.2.3 we have $A = 6.6095$ and $Z = 1.4038$. This corresponds to a Pascal traffic with $S = 16.37$ sources and $\beta = 0.2876$. The traffic congestion becomes 20.52% corresponding to a lost traffic 1.3563 erlang, or a blocking probability for the system equal to $E = 1.3563/24 = 5.65\%$. This result is quite accurate. □

9.4.2 Sanders' method

Sanders & Haemers & Wilcke (1983 [94]) have proposed another simple and interesting equivalence method also based on the state space. We will call it *Sanders' method*. Like Fredericks & Hayward's method, it is based on a change of scale of state probabilities so that the peakedness becomes equal to one. The method transforms a non–Poisson traffic with (mean, variance) = (m, v) into a traffic stream with peakedness one by adding a constant (zero–variance) traffic stream with mean $v - m$ so that the total traffic has mean equal to variance v. The constant traffic stream occupies $v-m$ channels permanently (with no loss) and we increase the number of channels by this amount. In this way we get a system with $n+(v-m)$ channels which are offered the traffic $m+(v-m) = v$ erlang. The peakedness becomes one, and the blocking probability is obtained using Erlang's B-formula, and so we find the traffic lost from the equivalent system.. This lost traffic is divided by the originally offered traffic to obtain the traffic congestion C.

The blocking probability relates to the originally offered traffic m. The method is applicable for both both smooth $m > v$ and bursty traffic $m < v$ and requires only the evaluation of the Erlang–B formula with a continuous number of channels.

Example 9.4.2: Sanders' method
If we apply Sanders' method to example 9.2.3, we increase both the number of channels and the offered traffic by $v - m = 2.6691$ (channels/erlang) and thus have 9.2786 erlang offered to 10.6691 channels. From Erlang's B-formula we find the lost traffic 1.3690 erlang, which is on the safe side, but close to the results obtained above. It corresponds to a blocking probability E = 1.3690/24 = 5.70%. □

9.4.3 Berkeley's method

To get an *ERT*–method based on only one parameter, we can in principle keep either n or A fixed. Experience shows that we obtain the best results by keeping the number of channels fixed $n_x = n$. We are now in the position where we only can ensure that the mean value of the overflow traffic is correct. This method is called *Berkeley's equivalence method* (1934). Wilkinson-Bretschneider's method requires a certain amount of computations (computers), whereas Berkeley's method is based on Erlang's B-formula only. Berkeley's method is applicable only for systems, where the primary groups all have the same number of channels.

Example 9.4.3: Group divided into primary and overflow group
If we apply Berkeley's method two example 9.1.1, then we get the exact solution, and from this special case originates the idea of the method. □

Example 9.4.4: Berkeley's method
We consider example 9.2.3 again. To apply Berkeley's method correctly, we should have the same number of channels in all three micro-cells. Let us assume all micro-cells have 8 channels (and not 16, 8, 0, respectively). To obtain the overflow traffic 6.6095 erlang the equivalent offered traffic is 13.72 erlang to the 8 primary channels. The equivalent system then has a traffic 13.72 erlang offered to $(8+8=)$ 16 channels. The lost traffic obtained from the Erlang-B formula becomes 1.4588 erlang corresponding to a blocking probability 6.08%, which is a value a little larger than the correct value. In general, Berkeley's method will be on the safe side. □

9.5 Methods based on arrival processes

The models in Chaps. 7 & 8 are all characterised by a Poisson arrival process with state dependent intensity, whereas the service times are exponentially distributed with equal mean value for all (homogeneous) servers. As these models all are independent of the service time distribution (insensitive, i.e. the state probabilities only depend on the mean value of the service time distribution), then we may only generalise the models by considering more general arrival processes. By using general arrival processes the insensitivity property is lost and the service time distribution becomes important. As we only have one arrival process, but many service processes (one for each of the n servers), then we in general assume exponential service times to avoid complex models.

9.5.1 Interrupted Poisson Process

In Sec. 6.4 we considered Kuczura's Interrupted Poisson Process (*IPP*) (Kuczura, 1977 [71]), which is characterised by three parameters and has been widely used for modelling overflow traffic. If we consider a full accessible group with n servers, which are offered calls arriving according to an *IPP* (cf. Fig. 6.7) with exponentially distributed service times, then we can construct a state transition diagram as shown in Fig. 9.6. The diagram is two-dimensional. State (i,j) denotes that there are i calls being served ($i = 0, 1, \ldots, n$), and that the arrival process is in phase j ($j = a$: arrival process *on*, $j = b$: arrival process *off*). By using the node balance equations we find the equilibrium state probabilities $p(i,j)$.

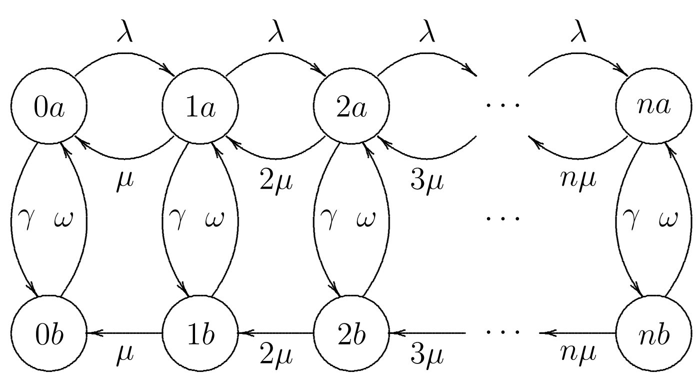


Figure 9.6: *State transition diagram for a full accessible loss system with n servers, IPP arrival process (cf. Fig. 6.7) and exponentially distributed service times (μ).*

Time congestion E becomes:

$$E = p(na) + p(nb)\,. \tag{9.19}$$

Call congestion B becomes:

$$B = \frac{p(na)}{\displaystyle\sum_{i=0}^{n} p(ia)} \geq E\,. \tag{9.20}$$

Traffic congestion C is defined as the proportion of the offered traffic which is lost. The offered traffic is equal to:

$$A = \frac{p(\text{on})}{p(\text{on}) + p(\text{off})} \cdot \frac{\lambda}{\mu} = \frac{\omega}{\omega+\gamma} \cdot \frac{\lambda}{\mu}\,.$$

The carried traffic is:

$$Y = \sum_{i=0}^{n} i \cdot \{p(ia) + p(ib)\}\,. \tag{9.21}$$

From this we obtain $C = (A - Y)/A$.

9.5.2 Cox–2 arrival process

In Sec. 6.4 we noticed that a *Cox–2* arrival process is more general than an *IPP* (Kuczura, 1977 [71]). If we consider *Cox–2* arrival processes as shown in Fig. 4.10, then we get the state transition diagram shown in Fig. 9.7. From this we find under the assumption of statistical

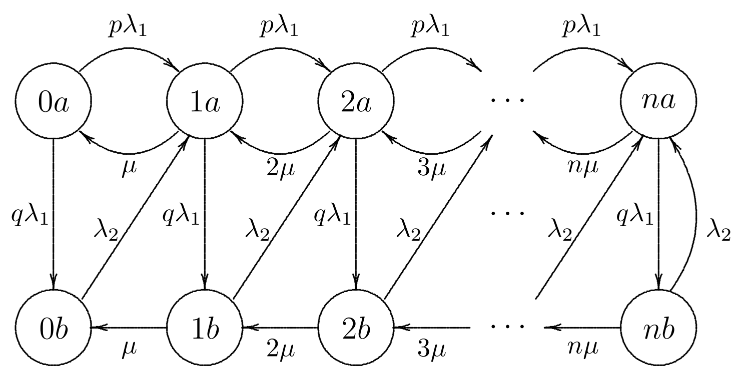


Figure 9.7: *State transition diagram for a full accessible loss system with n servers, Cox–2 arrival processes (cf. Fig. 4.10) and exponentially distributed service times (μ).*

equilibrium the state probabilities and the following performance measures.

Time congestion E:

$$E = p(na) + p(nb)\,. \tag{9.22}$$

Call congestion B:

$$B = \frac{p\,\lambda_1 \cdot p(na) + \lambda_2 \cdot p(nb)}{p\,\lambda_1 \cdot \displaystyle\sum_{i=0}^{n} p(ia) + \lambda_2 \cdot \sum_{i=0}^{n} p(ib)}\,. \tag{9.23}$$

Traffic congestion C. The offered traffic is the average number of call attempts per mean service time. The mean inter-arrival time is:

$$m_a = \frac{1}{\lambda_1} + (1-p) \cdot \frac{1}{\lambda_2} = \frac{\lambda_2 + (1-p)\lambda_1}{\lambda_1\,\lambda_2}\,.$$

The offered traffic then becomes $A = (m_a \cdot \mu)^{-1}$. The carried traffic is given by (9.21) applied to Fig. 9.7 and thus we can find the traffic congestion C.

If we generalise the arrival process to a *Cox–k* arrival process, then the state-transition diagram is still two-dimensional. By the application of *Cox*–distributions we can in principle take any number of parameters into consideration.

If we generalise the service time to a *Cox–k* distribution, then the state transition diagram becomes much more complex for $n > 1$, because we have a service process for each server, but only one arrival process. Therefore, in general we always generalise the arrival process and assume exponentially distributed service times.

Chapter 10

Multi-Dimensional Loss Systems

In this chapter we generalise the classical teletraffic theory to deal with service-integrated systems (*ISDN* and *B-ISDN*). Every class of service corresponds to a traffic stream. Several traffic streams are offered to the same trunk group.

In Sec. 10.1 we consider the classical multi-dimensional Erlang-B loss formula. This is an example of a reversible Markov process which is considered in more details in Sec. 10.2. In Sec. 10.3 we look at more general loss models and strategies, including service-protection (maximum allocation) and multi-slot *BPP*–traffic. The models all have the so-called *product-form* property, and the numerical evaluation is very simple by using the convolution algorithm for loss systems, implemented in the tool *ATMOS* (Sec. 10.4). In Sec. 10.4.2 we review other algorithms for the same problem.

The models considered are all based on flexible channel/slot allocation. They can be generalised to arbitrary circuit switched networks with direct routing, where we calculate the end-to-end blocking probabilities (Chap. 11). All models considered are insensitive to the service time distribution, and thus they are robust for applications. At the end of the chapter we consider other algorithms.

10.1 Multi-dimensional Erlang-B formula

We consider a group of n trunks (channels, slots), which is offered two independent *PCT–I* traffic streams: (λ_1, μ_1) and (λ_2, μ_2). The offered traffic becomes $A_1 = \lambda_1/\mu_1$, respectively $A_2 = \lambda_2/\mu_2$.

Let (i, j) denote the state of the system, i.e. i is the number of calls from stream 1 and j is the number of calls from stream 2. We have the following restrictions:

$$\begin{aligned} 0 \leq \; i \; &\leq n\,, \\ 0 \leq \; j \; &\leq n\,, \\ 0 \leq i+j &\leq n\,. \end{aligned} \tag{10.1}$$

The state transition diagram is shown in Fig. 10.1. Under the assumption of statistical equilibrium the state probabilities are obtained by solving the global balance equations for each node (node equations), in total $(n+1)(n+2)/2$ equations.

As we shall see in the next section, this diagram corresponds to a reversible Markov process, which has *local balance*, and furthermore the solution has product form. We can easily show that the global balance equations are satisfied by the following state probabilities which may be written on product form:

$$\begin{aligned} p(i,j) &= p(i) \cdot p(j) \\ &= Q \cdot \frac{A_1^i}{i!} \cdot \frac{A_2^j}{j!}\,, \end{aligned} \tag{10.2}$$

where $p(i)$ and $p(j)$ are one-dimensional truncated Poisson distributions, Q is a normalisation constant, and (i,j) fulfil the above restrictions (10.1). As we have Poisson arrival processes, which have the *PASTA*-property (*P*oisson *A*rrivals *S*ee *T*ime *A*verages), the time congestion, call congestion, and traffic congestion are all equal for both traffic streams, and they equals $P(i+j=n)$.

By the Binomial expansion or by convolving two Poisson distributions we find the following aggregated state probabilities, where Q is obtained by normalisation:

$$p(i+j=x) = Q \cdot \frac{(A_1+A_2)^x}{x!}\,, \tag{10.3}$$

$$Q^{-1} = \sum_{\nu=0}^{n} \frac{(A_1+A_2)^\nu}{\nu!}\,. \tag{10.4}$$

This is the Truncated Poisson distribution (7.9) with the offered traffic:

$$A = A_1 + A_2\,. \tag{10.5}$$

We may also interpret this model as an Erlang loss system with one Poisson arrival process and hyper-exponentially distributed holding times in the following way. The total arrival process is a superposition of two Poisson processes with the total arrival rate:

$$\lambda = \lambda_1 + \lambda_2\,, \tag{10.6}$$

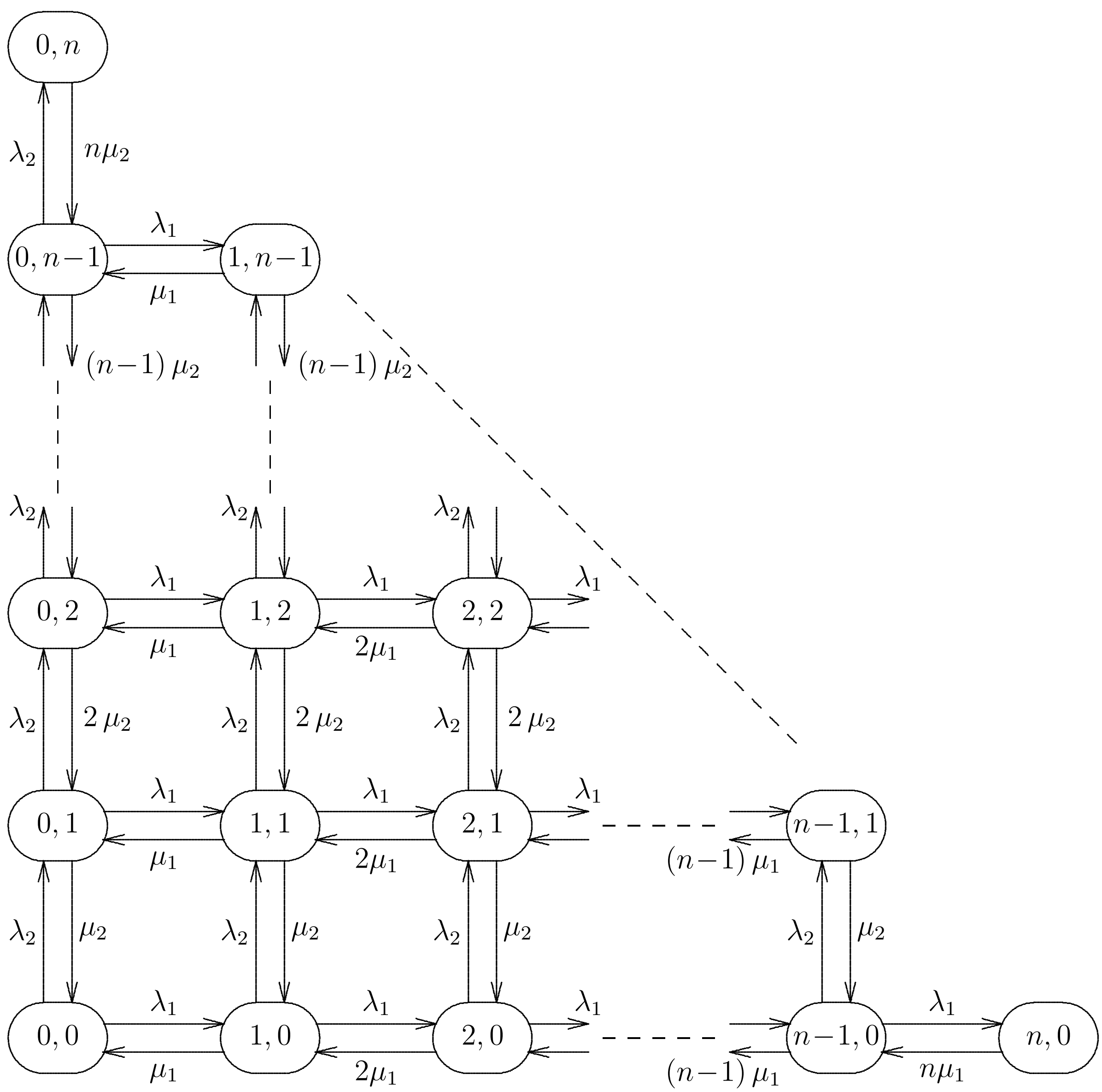


Figure 10.1: *Two-dimensional state transition diagram for a loss system with n channels which are offered two PCT–I traffic streams. This is equivalent to a state transition diagram for the loss system M/H$_2$/n, where the hyper-exponential distribution H$_2$ is given by (10.7).*

and the holding time distribution is hyper-exponentially distributed:

$$f(t) = \frac{\lambda_1}{\lambda_1+\lambda_2} \cdot \mu_1 \cdot \mathrm{e}^{-\mu_1 t} + \frac{\lambda_2}{\lambda_1+\lambda_2} \cdot \mu_2 \cdot \mathrm{e}^{-\mu_2 t} . \tag{10.7}$$

We weight the two exponential distributions according to the relative number of calls per time unit. The mean service time is

$$\begin{aligned} m_1 &= \frac{\lambda_1}{\lambda_1+\lambda_2} \cdot \frac{1}{\mu_1} + \frac{\lambda_2}{\lambda_1+\lambda_2} \cdot \frac{1}{\mu_2} = \frac{A_1+A_2}{\lambda_1+\lambda_2} , \\ m_1 &= \frac{A}{\lambda} , \end{aligned} \tag{10.8}$$

which is in agreement with the offered traffic.

Thus we have shown that Erlang's loss model is valid for hyper-exponentially distributed holding times. This is a special case of the general insensitivity property of Erlang's B–formula.

We may generalise the above model to N traffic streams:

$$p(i_1, i_2, \cdots, i_N) = Q \cdot \frac{A_1^{i_1}}{i_1!} \cdot \frac{A_2^{i_2}}{i_2!} \cdot \cdots \cdot \frac{A_N^{i_N}}{i_N!} , \quad 0 \leq i_j \leq n , \quad \sum_{j=1}^{N} i_j \leq n , \tag{10.9}$$

which is the general multi-dimensional Erlang-B formula. By a generalisation of (10.3) we notice that the global state probabilities can be calculated by the following recursion, where $q(x)$ denotes the relative state probabilities, and $p(x)$ denotes the absolute state probabilities:

$$q(x) = \frac{1}{x} \sum_{j=1}^{N} A_j \cdot q(x-1) , \quad q(0) = 1 , \tag{10.10}$$

$$Q(n) = \sum_{i=0}^{n} q(i) ,$$

$$p(x) = \frac{q(x)}{Q(n)} , \quad 0 \leq x \leq n . \tag{10.11}$$

If we use the recursion with normalisation (Sec. 7.4), then we get the recursion formula for Erlang–B. Formula (10.10) is similar to the balance equations for the Poisson case when:

$$A = \sum_{j=1}^{N} A_j .$$

The time congestion is $E = p(n)$, and as the *PASTA*-property is valid, this is also equal to the call congestion and the traffic congestion. The numerical evaluation is dealt with in detail in Sec. 10.4.

10.2 Reversible Markov processes

In the previous section we considered a two-dimensional state transition diagram. For an increasing number of traffic streams the number of states (and thus equations) increases very rapidly. However, we may simplify the problem by exploiting the structure of the state transition diagram. Let us consider the two-dimensional state transition diagram shown in Fig. 10.2. The process is reversible if there is no circulation flow in the diagram. Thus, if we consider four neighbouring states, then the flow in clockwise direction must equal the flow in the opposite direction (Kingman, 1969 [63]), (Sutton, 1980 [96]). From Fig. 10.2 we have:

Clockwise:

$$\begin{array}{rcll} [i,j] & \rightarrow & [i,j+1]: & p(i,j)\cdot\lambda_2(i,j) \\ [i,j+1] & \rightarrow & [i+1,j+1]: & p(i,j{+}1)\cdot\lambda_1(i,j{+}1) \\ [i+1,j+1] & \rightarrow & [i+1,j]: & p(i{+}1,j{+}1)\cdot\mu_2(i{+}1,j{+}1) \\ [i+1,j] & \rightarrow & [i,j]: & p(i{+}1,j)\cdot\mu_1(i{+}1,j)\,, \end{array}$$

Counter clockwise:

$$\begin{array}{rcll} [i,j] & \rightarrow & [i+1,j]: & p(i,j)\cdot\lambda_1(i,j) \\ [i+1,j] & \rightarrow & [i+1,j+1]: & p(i{+}1,j)\cdot\lambda_2(i{+}1,j) \\ [i+1,j+1] & \rightarrow & [i,j+1]: & p(i{+}1,j{+}1)\cdot\mu_1(i{+}1,j{+}1) \\ [i,j+1] & \rightarrow & [i,j]: & p(i,j{+}1)\cdot\mu_2(i,j{+}1)\,. \end{array}$$

We can reduce both expressions by the state probabilities and then obtain the condition given by (10.12). It can be shown that a necessary and sufficient condition for reversibility is that the following two expressions are equal:
Clockwise:

$$\lambda_2(i,j)\cdot\lambda_1(i,j{+}1)\cdot\mu_2(i{+}1,j{+}1)\cdot\mu_1(i{+}1,j) \tag{10.12}$$

Counter clockwise:

$$\lambda_1(i,j)\cdot\lambda_2(i{+}1,j)\cdot\mu_1(i{+}1,j{+}1)\cdot\mu_2(i,j{+}1)\,.$$

If these two expressions are equal, then there is *local or detailed balance*. A necessary condition for reversibility is thus that if there is a flow (an arrow) from state i to state j, then there must also be a flow (an arrow) from j to i. We may locally apply cut equations between any two connected states. Thus from Fig. 10.2 we get:

$$p(i,j)\cdot\lambda_1(i,j) = p(i+1,j)\cdot\mu_1(i+1,j)\,. \tag{10.13}$$

We can express any state probability $p(i,j)$ by state probability $p(0,0)$ by choosing any path between the two states *(Kolmogorov's criteria).* We may for instance choose the path:

$$(0,0),(1,0),\ldots,(i,0),(i,1),\ldots,(i,j)\,,$$

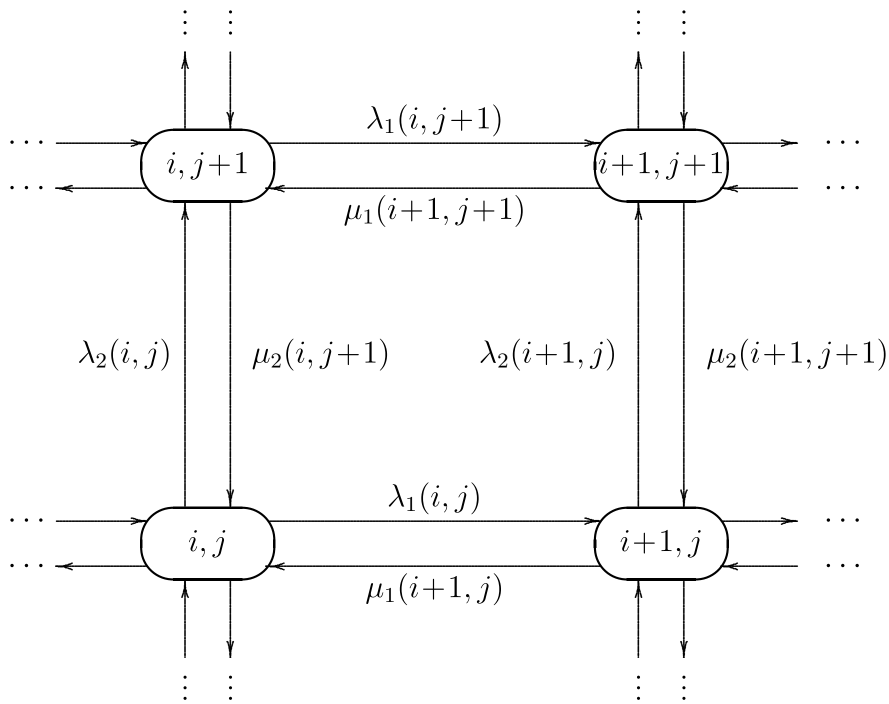


Figure 10.2: *Kolmogorov's criteria: a necessary and sufficient condition for reversibility of a two-dimensional Markov process is that the circulation flow among four neighbouring states in a square equals zero: Flow clockwise = flow counter-clockwise (10.12).*

and we then obtain the following balance equation:

$$p(i,j) = \frac{\lambda_1(0,0)}{\mu_1(1,0)} \cdot \frac{\lambda_1(1,0)}{\mu_1(2,0)} \cdots \frac{\lambda_1(i-1,0)}{\mu_1(i,0)} \cdot \frac{\lambda_2(i,0)}{\mu_2(i,1)} \cdot \frac{\lambda_2(i,1)}{\mu_2(i,2)} \cdots \frac{\lambda_2(i,j-1)}{\mu_2(i,j)} \cdot p(0,0)$$

We find $p(0,0)$ by normalisation of the total probability mass.

The condition for reversibility will be fulfilled in many cases, for example for:

$$\lambda_1(i,j) = \lambda_1(i)\,, \qquad \mu_1(i,j) = i \cdot \mu_1\,, \tag{10.14}$$

$$\lambda_2(i,j) = \lambda_2(j)\,, \qquad \mu_2(i,j) = j \cdot \mu_2\,. \tag{10.15}$$

If we consider a multi-dimensional loss system with N traffic streams, then any traffic stream may be a state-dependent Poisson process, in particular *BPP* (Bernoulli, Poisson, Pascal) traffic streams. For N–dimensional systems the conditions for reversibility are analogue to (10.12). Kolmogorov's criteria must still be fulfilled for all possible paths. In practice, we experience no problems, because the solution obtained under the assumption of reversibility will be the correct solution if and only if the node balance equations are fulfilled. In the following section we use this as the basis for introducing a very general multi-dimensional traffic model.

10.3 Multi-Dimensional Loss Systems

In this section we consider generalisations of the classical teletraffic theory to cover several traffic streams offered to a single channel/trunk group. Each traffic stream may have individual parameters and may be state-dependent Poisson arrival processes with multi-slot traffic and class limitations. This general class of models is insensitive to the holding time distribution, which may be class dependent with individual parameters for each class. We introduce the generalisations one at a time and present a small case-study to illustrate the basic ideas.

10.3.1 Class limitation

In comparison with the case considered in Sec. 10.1 we now restrict the number of simultaneous calls for each traffic stream (class). Thus, we do not have full accessibility, but unlike overflow systems where we physically only have access to specific channels, then we now have access to all channels, but at any instant we may only occupy a limited number. This may be used for the purpose of service protection (virtual circuit protection = class limitation = threshold priority policy). We thus introduce restrictions to the number of simultaneous calls in class j as follows:

$$0 \;\leq\; i_j \;\leq\; n_j \;\leq n\,, \qquad j = 1, 2, \ldots, N\,, \tag{10.16}$$

where

$$\sum_{j=1}^{N} n_j > n\,.$$

If the latter restriction is not fulfilled, then we get separate groups corresponding to N ordinary independent one-dimensional loss systems. Due to the restrictions the state transition diagram is truncated. This is shown for two traffic streams in Fig. 10.3.

We notice that the truncated state transition diagram still is reversible and that the value of $p(i, j)$ relative to the value $p(0, 0)$ is unchanged by the truncation. Only the normalisation constant is modified. In fact, due to the local balance property we can remove any state without changing the above properties. We may consider more general class limitations to sets of traffic streams so that any traffic stream has a minimum (guaranteed) number of allocated channels.

10.3.2 Generalised traffic processes

We are not restricted to consider *PCT–I* traffic only as in Sec. 10.1. Every traffic stream may be a state-dependent Poisson arrival process with a linear state-dependent death (departure)

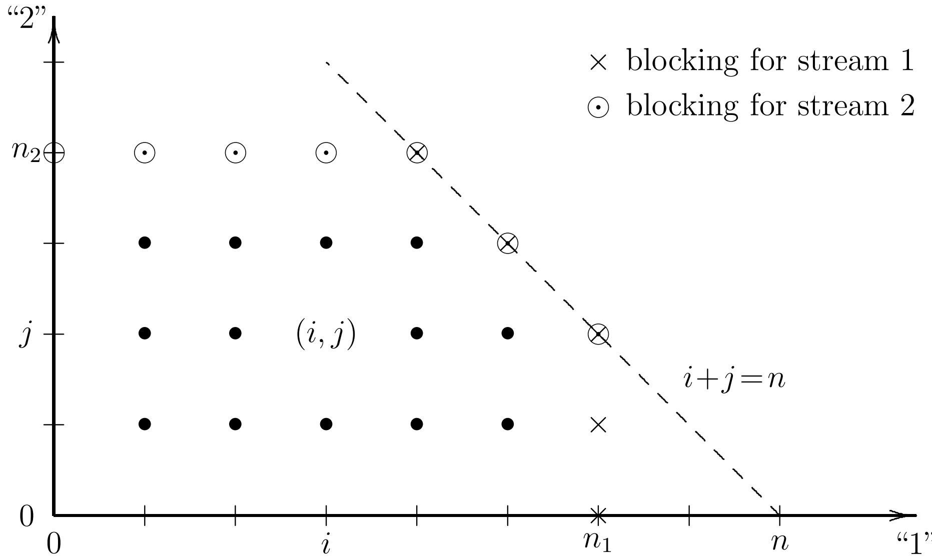


Figure 10.3: *Structure of the state transition diagram for two-dimensional traffic processes with class limitations (cf. 10.16). When calculating the equilibrium probabilities, state (i, j) can be expressed by state $(i, j-1)$ and recursively by state $(i, 0)$, $(i-1, 0)$, and finally by $(0, 0)$ (cf. (10.14)).*

rate (cf. (10.14) and (10.15)). The system still fulfils the reversibility conditions given by (10.12). Thus, the product form still exists for *BPP* traffic streams and more general state-dependent Poisson processes. If all traffic streams are Engset– (Binomial–) processes, then we get the multi-dimensional Engset formula (Jensen, 1948 [48]). As mentioned above, the system is insensitive to the holding time distributions. Every traffic stream may have its own individual holding time distribution.

10.3.3 Multi-slot traffic

In service-integrated systems the bandwidth requested may depend on the type of service. Thus a voice telephone call requires one channel (slot) only, whereas for example a video service may require d channels simultaneously. We get the additional restrictions:

$$0 \leq d_j \cdot i_j \ \leq \ n_j \ \leq \ n \,, \quad j = 1, 2, \ldots, N \,, \tag{10.17}$$

and

$$0 \leq \sum_{j=1}^{N} d_j \cdot i_j \ \leq n \,, \tag{10.18}$$

where i_j is the actual number of type j calls. The resulting state transition diagram will still be reversible and have product form. The restrictions correspond for example to the physical model shown in Fig. 10.5.

Stream 1: *PCT–I* traffic	Stream 2: *PCT–II* traffic
$\lambda_1 = 2$ calls/time unit	$S_2 = 4$ sources
	$\gamma_2 = 1/3$ calls/time unit/idle source
$\mu_1 = 1$ (time units^{-1})	$\mu_2 = 1$ (time units^{-1})
	$\beta_2 = \gamma_2/\mu_2 = 1/3$ erlang per idle source
$Z_1 = 1$ (peakedness)	$Z_2 = 1/(1+\beta_2) = 3/4$ (peakedness)
$d_1 = 1$ channel/call	$d_2 = 2$ channels/call
$A_1 = \lambda_1/\mu_1 = 2$ erlang	$A_2 = S_2 \cdot \beta_2/(1+\beta_2) = 1$ erlang
$n_1 = 6 = n$	$n_2 = 6 = n$

Table 10.1: *Two traffic streams: a Poisson traffic process (Example 7.5.1) and a Binomial traffic process (Example 8.5.1) are offered to the same trunk group.*

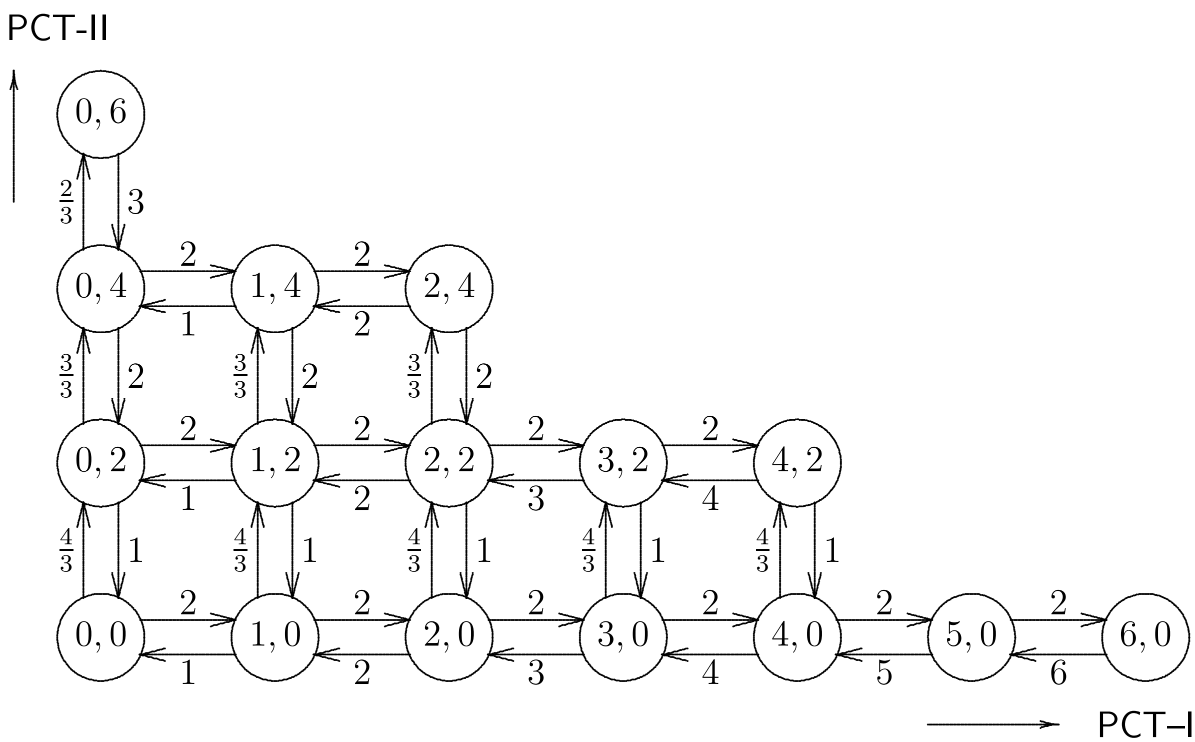
Offered traffic A_j is usually defined as the average number of call attempts per mean holding time. If we measure the carried traffic Y_j as the average number of busy channels, then the lost traffic measured in channels becomes:

$$A_\ell = \sum_{j=1}^{N} A_j \, d_j - \sum_{j=1}^{N} Y_j \,. \tag{10.19}$$

Example 10.3.1: Rönnblom's model
The first example of a multi-slot traffic model was published by Rönnblom (1958 [93]). The paper considers external (outgoing and incoming) traffic and internal traffic in a *PABX* telephone exchange with both-way channels. The external traffic occupies only one channel per call. The internal traffic occupies both an outgoing channel and an incoming channel and thus requires two channels simultaneously. Rönnblom showed that this model has product form. □

Example 10.3.2: Two traffic streams
Let us illustrate the above models by a small case-study. We consider a trunk group of 6 channels which is offered two traffic streams specified in Tab. 10.1. We notice that the second traffic stream is a multi-slot traffic stream. We may at most have three type-2 calls in our system. We only need to specify the offered traffic, not the absolute values of arrival rate and service rate. The offered traffic is as usually defined as the traffic carried by an infinite trunk group.

We get the two-dimensional state transition diagram shown in Fig. 10.4. The total sum of all relative state probabilities equals 20.1704. So by normalisation we find $p(0,0) = 0.0496$ and thus the following state probabilities and marginal state probabilities $p(i,\cdot)$ and $p(\cdot,j)$.



$\frac{4}{27}$						
$\frac{2}{3}$	$\frac{4}{3}$	$\frac{4}{3}$				
$\frac{4}{3}$	$\frac{8}{3}$	$\frac{8}{3}$	$\frac{16}{9}$	$\frac{8}{9}$		
1	2	2	$\frac{4}{3}$	$\frac{2}{3}$	$\frac{4}{15}$	$\frac{4}{45}$

Figure 10.4: *Example 10.3.2: Six channels are offered both a Poisson traffic stream (PCT–I) (horizontal states) and an Engset traffic stream (PCT–II) (vertical states). The parameters are specified in Tab. 10.1. If we allocate state $(0,0)$ the relative value one, then we find by exploiting local balance the relative state probabilities $q(i,j)$ shown below.*

$\boldsymbol{p(i,j)}$	$i=0$	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$	$i=6$	$\boldsymbol{p(\cdot\,,j)}$
$j=6$	0.0073							0.0073
$j=4$	0.0331	0.0661	0.0661					0.1653
$j=2$	0.0661	0.1322	0.1322	0.0881	0.0441			0.4627
$j=0$	0.0496	0.0992	0.0992	0.0661	0.0331	0.0132	0.0044	0.3647
$\boldsymbol{p(i,\cdot)}$	0.1561	0.2975	0.2975	0.1542	0.0771	0.0132	0.0044	1.0000

The global state probabilities become:

$$\begin{array}{lcl} p(0)=p(0,0) & = & 0.0496 \\ p(1)=p(1,0) & = & 0.0992 \\ p(2)=p(0,2)+p(2,0) & = & 0.1653 \\ p(3)=p(1,2)+p(3,0) & = & 0.1983 \\ p(4)=p(0,4)+p(2,2)+p(4,0) & = & 0.1983 \\ p(5)=p(1,4)+p(3,2)+p(5,0) & = & 0.1675 \\ p(6)=p(0,6)+p(2,4)+p(4,2)+p(6,0) & = & 0.1219 \end{array}$$

Performance measures for traffic stream 1:

Due to the *PASTA*-property the time congestion (E_1), the call congestion (B_1), and the traffic congestion (C_1) are identical. We find the time congestion E_1:

$$\begin{aligned} E_1 &= p(6,0)+p(4,2)+p(2,4)+p(0,6) \\ &= p(6)\,, \\ E_1 &= B_1 \;=\; C_1 = 0.1219\,, \\ Y_1 &= 1.7562\,. \end{aligned}$$

Performance measures for stream 2:

Time congestion E_2 (proportion of time the system is blocked for stream 2):

$$\begin{aligned} E_2 &= p(0,6)+p(1,4)+p(2,4)+p(3,2)+p(4,2)+p(5,0)+p(6,0) \\ &= p(5)+p(6)\,, \\ E_2 &= 0.2894\,. \end{aligned}$$

Call congestion B_2 (Proportion of call attempts blocked for stream 2):

The total number of call attempts per time unit is obtained from the marginal distribution:

$$\begin{aligned} x_t &= \frac{4}{3} \cdot 0.3647 + \frac{3}{3} \cdot 0.4627 + \frac{2}{3} \cdot 0.1653 + \frac{1}{3} \cdot 0.0073 \\ &= 1.0616\,. \end{aligned}$$

The number of blocked call attempts per time unit becomes:

$$\begin{aligned} x_\ell &= \frac{4}{3} \cdot \{p(5,0)+p(6,0)\} + \frac{3}{3} \cdot \{p(3,2)+p(4,2)\} + \frac{2}{3} \cdot \{p(1,4)+p(2,4)\} + \frac{1}{3} \cdot p(0,6) \\ &= 0.2462\,. \end{aligned}$$

Hence:

$$B_2 = \frac{x_\ell}{x_t} = 0.2320\,.$$

Traffic congestion C_2 (Proportion of offered traffic blocked):

The carried traffic, measured in the unit [*channel*], is obtained from the marginal distribution:

$$\begin{aligned} Y_2 &= \sum_{j=0}^{6} j \cdot p(\cdot, j)\,, \\ Y_2 &= 2 \cdot 0.4627 + 4 \cdot 0.1653 + 6 \cdot 0.0073\,, \\ Y_2 &= 1.6306 \quad \text{erlang}\,. \end{aligned}$$

The offered traffic, measured in the unit [*channel*], is $d_2 \cdot A_2 = 2$ erlang (Tab. 10.1). Hence we get:

$$C_2 = \frac{2 - 1.6306}{2} = 0.1848\,.$$

□

The above example has only 2 streams and 6 channels and the total number of states equals 14 (Fig. 10.4). When the number of traffic streams and channels increase, then the number of states increases very fast and we are unable to evaluate the system by calculating the individual state probabilities. In the following section we introduce the convolution algorithm for loss systems which eliminates this problem by aggregation of states.

10.4 Convolution Algorithm for loss systems

We now consider a trunk group with a total of n homogeneous trunks. Being homogeneous means that they have the same service rate. The trunk group is offered N different types of calls, also called streams, or classes. A call of type i requires d_i trunks (channels, slots) during the whole service time, i.e. all d_i channels are occupied and released simultaneously. The arrival processes are general state-dependent Poisson processes. For the i'th arrival process the arrival intensity in state $x_i \cdot d_i$, that is, when x_i calls of type i are being served, is $\lambda_i(x_i)$. We may restrict the number x_i of simultaneous calls of type i so that:

$$0 \;\leq\; x_i \cdot d_i \;\leq\; n_i \;\leq\; n\,.$$

It will be natural to require that n_i is an integral multiple of d_i. This model describes for example the system shown in Fig. 10.5.

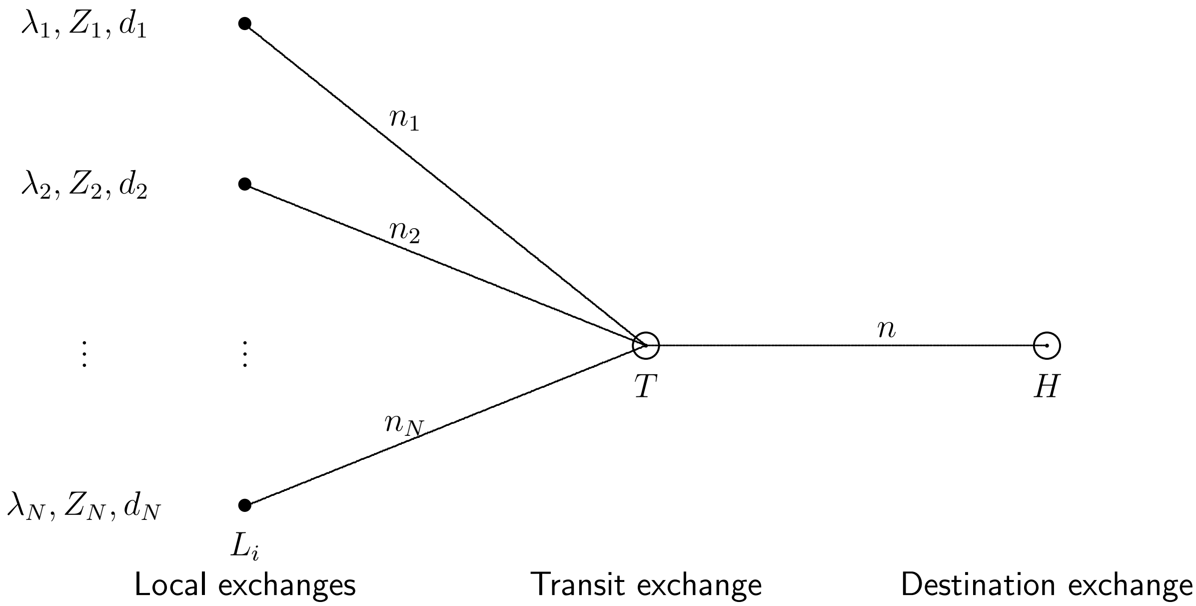


Figure 10.5: *Generalisation of the classical teletraffic model to BPP–traffic and multi-slot traffic. The parameters λ_i and Z_i describe the BPP–traffic, whereas d_i denotes the number of slots required.*

The system mentioned above can be evaluated in an efficient way by the convolution algorithm first introduced in (Iversen, 1987 [40]). We first describe the algorithm, and then explain it in further detail by an example. The convolution algorithm is closely related to the product-form.

10.4.1 The algorithm

The algorithm is described by the following three steps:

- **Step 1:** Calculate the state probabilities of each traffic stream as if it is alone in the system, i.e. we consider classical loss systems as described in Chaps. 7 & 8. For traffic stream i we find:

$$\underline{P_i} = \{p_i(0),\ p_i(1),\ \ldots\ , p_i(n_i)\}\,, \quad i = 1, 2, \ldots, N\,. \tag{10.20}$$

Only the relative values of $p_i(x)$ are of importance, so we may choose $q_i(0) = 1$ and calculate the values of $q_i(x)$ relative to $q_i(0)$. If a term $q_i(x)$ becomes greater than K (e.g. 10^{10}), then we may divide all values $q_i(j)$, $0 \le j \le x$, by K. To avoid any numerical problems in the following it is advisable to normalise the relative state probabilities so that:

$$p_i(j) = \frac{q_i(j)}{Q_i}\,, \quad j = 0, 1 \ldots, n_i\,, \quad Q_i = \sum_{j=0}^{n_i} q_i(j)\,.$$

As described in Sec. 7.4 we may normalise at each step to avoid any numerical problems.

- **Step 2:** By successive convolutions (convolution operator $*$) we calculate the aggregated state probabilities for the total system excepting traffic stream number i:

$$\underline{Q_{N/i}} = \underline{P_1} * \underline{P_2} * \cdots * \underline{P_{i-1}} * \underline{P_{i+1}} * \cdots * \underline{P_N}. \tag{10.21}$$

We first convolve $\underline{P_1}$ and $\underline{P_2}$ and obtain $\underline{P_{12}}$ which is convolved with $\underline{P_3}$, etc. Both the commutative and the associative laws are valid for the convolution operator, defined in the usual way (Sec. 3.2):

$$\underline{P_i} * \underline{P_j} = \left\{ p_i(0) \cdot p_j(0),\ \sum_{x=0}^{1} p_i(x) \cdot p_j(1-x),\ \cdots\ ,\ \sum_{x=0}^{u} p_i(x) \cdot p_j(u-x) \right\}, \tag{10.22}$$

where

$$u = \min\{n_i + n_j, n\}\,. \tag{10.23}$$

Notice, that we truncate the state space at state n. Even if $\underline{P_i}$ and $\underline{P_j}$ are normalised, then the result of a convolution is in general not normalised due to the truncation. It is recommended to normalise after every convolution to avoid any numerical problems both during this step and the following.

- **Step 3:** Calculate the time congestion E_i, the call congestion B_i, and the traffic congestion C_i of stream i. This is done during the convolution:

$$\underline{Q_N} = \underline{Q_{N/i}} * \underline{P_i}\,.$$

This convolution results in:

$$Q_N(j) = \sum_{x=0}^{j} Q_{N/i}(j-x) \cdot p_i(x) = \sum_{x=0}^{j} p_x^i(j)\,, \tag{10.24}$$

where for $p_x^i(j)$, i denotes the traffic stream, j the total number of busy channels, and x the number of channels occupied by stream number i. Steps 2 – 3 are repeated for every traffic stream. In the following we derive formulæ for E_i, B_i, and C_i.

Time congestion E_i for traffic stream i becomes:

$$E_i = \sum_{j \in S_{E^i}} p_x^i(j)/Q\,. \tag{10.25}$$

where

$$S_{E^i} = \{(x,j) \mid x \leq j \leq n \wedge (x > n_i - d_i) \vee (j > n - d_i)\}\ ,$$

The summation is extended to all states S_{E^i} where calls belonging to class i are blocked: the set $\{x > n_i - d_i\}$ corresponds to the states where traffic stream i has utilised its quota, and $(j > n - d_i)$ corresponds to states with less than d_i idle channels. Q is the normalisation constant:

$$Q = \sum_{j=0}^{n} Q_N(j)\,.$$

(At this stage we usually have normalised the state probabilities so that $Q = 1$).

Call congestion B_i for traffic stream i is the ratio between the number of blocked call attempts and the total number of call attempts, both for traffic stream i, and for example per time unit. We find:

$$B_i = \frac{\sum_{S_{E^i}} \lambda_i(x) \cdot p_x^i(j)}{\sum_{j=0}^{n_i} \sum_{x=0}^{j} \lambda_i(x) \cdot p_x^i(j)}\,. \tag{10.26}$$

Traffic congestion C_i*:* We define as usual the offered traffic as the traffic carried by an infinite trunk group. The carried traffic for traffic stream i is:

$$Y_i = \sum_{j=0}^{n_i} \sum_{x=0}^{j} x \cdot p_x^i(j)\,. \tag{10.27}$$

Thus we find:

$$C_i = \frac{A_i - Y_i}{A_i}\,.$$

The algorithm is implemented in the PC-tool *ATMOS* (Listov–Saabye & Iversen, 1989 [74]). The storage requirement is proportional to n as we may calculate the state probabilities of a traffic stream when it is needed. In practice we use a storage proportional with $n \cdot N$, because we save intermediate results of the convolutions for later re-use. It can be shown (Iversen & Stepanov, 1997 [42]) that we need $(4 \cdot N - 6)$ convolutions when we calculate traffic characteristics for all N traffic streams. Thus the calculation time is linear in N and quadratic in n.

Example 10.4.1: De-convolution
In principle we may obtain $\underline{Q_{N/i}}$ from $\underline{Q_N}$ by a de-convolution and then during the re-convolution of $\underline{P_i}$ calculate the performance measures. In this way we need not repeat all the convolutions (10.21)

Stream 3: Pascal traffic (Negative Binomial)
$S_3 = -2$ sources
$\gamma_3 = -1/3$ calls/time unit
$\mu_3 = 1$ (time unit^{-1})
$\beta_3 = \gamma_3/\mu_3 = -1/3$ erlang per idle source
$Z_3 = 1/(1+\beta_3) = 3/2$
$d_3 = 1$ channels/call
$A_3 = S_3 \cdot (1 - Z_3) = 1$ erlang
$n_3 = 4$ (max. # of simultaneous calls)

Table 10.2: *A Pascal traffic stream (Example 8.7.2) is offered to the same trunk as the two traffic streams of Tab. 10.1.*

for each traffic stream. However, when implementing this approach we get numerical problems. The convolution is from a numerical point of view very stable, and therefore the de-convolution will be unstable. Nevertheless, we may apply de-convolution in some cases, for instance when the traffic sources are *on/off*–sources. □

Example 10.4.2: Three traffic streams
We first illustrate the algorithm with a small example, where we go through the calculations in every detail. We consider a system with 6 channels and 3 traffic streams. In addition to the two streams in Example 10.3.2 we add a Pascal stream with class limitation as shown in Tab. 10.2 (cf. Example 8.7.2). We want to calculate the performance measures of traffic stream 3.

- **Step 1**: We calculate the state probabilities $p_i(j)$ of each traffic stream i ($i = 1, 2, 3,\ j = 1, 2, \ldots, n_i$) as if it were alone. The results are given in Tab. 10.3.
- **Step 2:** We evaluate the convolution of $p_1(j)$ with $p_2(k)$, $p_1 * p_2$, truncate the state space at $n = 6$, and normalise the probabilities so that we obtain p_{12} shown in the Tab. 10.3. Notice that this is the result obtained in Example 10.3.2.
- **Step 3:** We convolve $p_{12}(j)$ with $p_3(k)$, truncate at n, and obtain $q_{123}(j)$ as shown in Tab. 10.3.

The time congestion E_3 is obtained from the detailed state probabilities. Traffic stream 3 (single–slot traffic) experiences time congestion, both when all six channels are busy and when the traffic stream occupies 4 channels (maximum allocation). From the detailed state probabilities we get:

$$\begin{aligned} E_3 &= \frac{q_{123}(6) + p_3(4) \cdot \{p_{12}(0) + p_{12}(1)\}}{0.8678} \\ &= \frac{0.1535 + 0.0279 \cdot \{0.0496 + 0.0992\}}{0.8678}, \\ E_3 &= 0.1817\,. \end{aligned}$$

State	Probabilities		$q_{12}(j)$	Normal.	Prob.	$q_{123}(j)$	Normal.
j	$p_1(j)$	$p_2(j)$	$p_1 * p_2$	$p_{12}(j)$	$p_3(j)$	$p_{12} * p_3$	$p_{123}(j)$
0	0.1360	0.3176	0.0432	0.0496	0.4525	0.0224	0.0259
1	0.2719	0.0000	0.0864	0.0992	0.3017	0.0599	0.0689
2	0.2719	0.4235	0.1440	0.1653	0.1508	0.1122	0.1293
3	0.1813	0.0000	0.1727	0.1983	0.0670	0.1579	0.1819
4	0.0906	0.2118	0.1727	0.1983	0.0279	0.1825	0.2104
5	0.0363	0.0000	0.1459	0.1675	0.0000	0.1794	0.2067
6	0.0121	0.0471	0.1062	0.1219	0.0000	0.1535	0.1769
Total	1.0000	1.0000	0.8711	1.0000	1.0000	0.8678	1.0000

Table 10.3: *Convolution algorithm applied to Example 10.4.2. The state probabilities for the individual traffic streams have been calculated in the examples 7.5.1, 8.5.1 and 8.7.2.*

Notice that the state $\{p_3(4) \cdot p_{12}(2)\}$ is included in state $q_{123}(6)$. The carried traffic for traffic stream 3 is obtained during the convolution of $p_3(i)$ and $p_{12}(j)$ and becomes:

$$Y_3 = \frac{1}{0.8678} \left\{ \sum_{i=1}^{4} i \cdot p_3(i) \sum_{j=0}^{6-i} p_{12}(j) \right\},$$

$$Y_3 = \frac{0.6174}{0.8678} = 0.7115\,.$$

As the offered traffic is $A_3 = 1$, we get:

Traffic congestion:

$$C_3 = \frac{1 - 0.7115}{1},$$

$$C_3 = 0.2885\,.$$

The call congestion becomes:

$$B_3 = \frac{x_\ell}{x_t},$$

where x_ℓ is the number of lost calls per time unit, and x_t is the total number of call attempts per time unit. Using the normalised probabilities from Tab. 10.3 we get $\{\lambda_3(i) = (S_3 - i)\,\gamma_3\}$:

$$\begin{aligned} x_\ell = {} & \lambda_3(0) \cdot \{p_3(0) \cdot p_{12}(6)\} \\ & + \lambda_3(1) \cdot \{p_3(1) \cdot p_{12}(5)\} \\ & + \lambda_3(2) \cdot \{p_3(2) \cdot p_{12}(4)\} \\ & + \lambda_3(3) \cdot \{p_3(3) \cdot p_{12}(3)\} \\ & + \lambda_3(4) \cdot p_3(4) \cdot \{p_{12}(2) + p_{12}(1) + p_{12}(0)\}, \end{aligned}$$

$$
\begin{aligned}
x_\ell &= 0.2503\,. \\
x_t &= \lambda_3(0)\cdot p_3(0)\cdot \sum_{j=0}^{6} p_{12}(j) \\
&\quad + \lambda_3(1)\cdot p_3(1)\cdot \sum_{j=0}^{5} p_{12}(j) \\
&\quad + \lambda_3(2)\cdot p_3(2)\cdot \sum_{j=0}^{4} p_{12}(j) \\
&\quad + \lambda_3(3)\cdot p_3(3)\cdot \sum_{j=0}^{3} p_{12}(j) \\
&\quad + \lambda_3(4)\cdot p_3(4)\cdot \sum_{j=0}^{2} p_{12}(j)\,, \\
x_t &= 1.1763\,.
\end{aligned}
$$

We thus get:

$$B_3 = \frac{x_\ell}{x_t} = 0.2128\,.$$

In a similar way by interchanging the order of convolving traffic streams we find the performance measures of stream 1 and 2. The total number of micro-states in this example is 47. By the convolution method we reduce the number of states so that we never need more than two vectors of each $n+1$ states, i.e. 14 states.

By using the *ATMOS*–tool we get the following results shown in Tab. 10.4 and Tab. 10.5. The total congestion can be split up into congestion due to class limitation (n_i), and congestion due to the limited number of channels (n). □

Input	Total number of channels $n = 6$						
	Offered traffic	Peaked ness	Maximum allocation	Slot size	Mean holding time	Sources	beta
i	A_i	Z_i	n_i	d_i	μ_i^{-1}	S_i	β_i
1	2.0000	1.00	6	1	1.00	∞	0
2	1.0000	0.75	6	2	1.00	4	0.3333
3	1.0000	1.50	4	1	1.00	-2	-0.5000

Table 10.4: *Input data to ATMOS for Example 10.4.2 with three traffic streams.*

Output	Call congestion	Traffic congestion	Time congestion	Carried traffic
i	B_i	C_i	E_i	Y_i
1	1.769 200E-01	1.769 200E-01	1.769 200E-01	1.646 160
2	3.346 853E-01	2.739 344E-01	3.836 316E-01	1.452 131
3	2.127 890E-01	2.884 898E-01	1.817 079E-01	0.711 510
Total		2.380 397E-01		3.809 801

Table 10.5: *Output data from ATMOS for the input data in Tab. 10.4.*

Example 10.4.3: Large-scale example
To illustrate the tool *"ATMOS"* we consider in Tab. 10.6 and Tab. 10.7 an example with 1536 trunks and 24 traffic streams. We notice that the time congestion is independent of peakedness Z_i and proportional to the slot-size d_i, because we often have:

$$p(j) \approx p(j-1) \approx \ldots \approx p(j-d_i) \quad \text{for} \quad d_i \ll j\,. \tag{10.28}$$

This is obvious as the time congestion only depends on the global state probabilities. The call congestion is almost equal to the time congestion. It depends weakly upon the slot-size. This is also to be expected, as the call congestion is equal to the time congestion with one source removed (*arrival theorem*). In the table with output data we have in the rightmost column shown the relative traffic congestion divided by $(d_i \cdot Z_i)$, using the single-slot Poisson traffic as reference value $(d_i = Z_i = 1)$. We notice that the traffic congestion is proportional to $d_i \cdot Z_i$, which is the usual assumption when using the Equivalent Random Traffic (*ERT*) method (Chap. 9). The mean value of the offered traffic increases linearly with the slot-size, whereas the variance increases with the square of the slot-size. The peakedness (variance/mean) ratio for multi-slot traffic thus increases linearly with the slot-size. We thus notice that the traffic congestion is much more relevant than the time congestion and call congestion for characterising the performance of the system. If we calculate the total traffic congestion using *Fredericks & Hayward's* method (Sec. 9.2), then we get a total traffic congestion equal to 6.114 % (cf. Example 9.3.2 and Tab. 10.7). The exact value is 5.950 %. □

10.4.2 Other algorithms

The convolution algorithm for loss systems was first published in (Iversen, 1987 [40]). A similar approach to a less general model was published in two papers by Ross & Tsang (1990 [91]), (1990 [92]) without reference to this original paper from 1987 though it was known by the authors. In case of Poisson arrival processes the algorithms become very simple, and we find *Fortet & Grandjean's algorithm* (Fortet & Grandjean, 1964 [28]):

$$q(x) \;=\; \frac{1}{x}\sum_{i=1}^{N} d_i\, A_i \cdot q(x-d_i)\,, \quad x = 1, 2, \ldots, n\,, \tag{10.29}$$

$$q(0) = 1\,, \qquad q(x) = 0 \quad \text{whenever} \quad x < 0\,.$$

Input	Total # of channels $n = 1536$						
	Offered traf.	Peakedness	Max. sim. #	Channels/call	mht	Sources	
i	A_i	Z_i	n_i	d_i	μ_i	S	β
1	64.000	0.200	1536	1	1.000	80.000	4.000
2	64.000	0.500	1536	1	1.000	128.000	1.000
3	64.000	1.000	1536	1	1.000	∞	0.000
4	64.000	2.000	1536	1	1.000	-64.000	-0.500
5	64.000	4.000	1536	1	1.000	-21.333	-0.750
6	64.000	8.000	1536	1	1.000	-9.143	-0.875
7	32.000	0.200	1536	2	1.000	40.000	4.000
8	32.000	0.500	1536	2	1.000	64.000	1.000
9	32.000	1.000	1536	2	1.000	∞	0.000
10	32.000	2.000	1536	2	1.000	-32.000	-0.500
11	32.000	4.000	1536	2	1.000	-10.667	-0.750
12	32.000	8.000	1536	2	1.000	-4.571	-0.875
13	16.000	0.200	1536	4	1.000	20.000	4.000
14	16.000	0.500	1536	4	1.000	32.000	1.000
15	16.000	1.000	1536	4	1.000	∞	0.000
16	16.000	2.000	1536	4	1.000	-16.000	-0.500
17	16.000	4.000	1536	4	1.000	-5.333	-0.750
18	16.000	8.000	1536	4	1.000	-2.286	-0.875
19	8.000	0.200	1536	8	1.000	10.000	4.000
20	8.000	0.500	1536	8	1.000	16.000	1.000
21	8.000	1.000	1536	8	1.000	∞	0.000
22	8.000	2.000	1536	8	1.000	-8.000	-0.500
23	8.000	4.000	1536	8	1.000	-2.667	-0.750
24	8.000	8.000	1536	8	1.000	-1.143	-0.875

Table 10.6: *Input data for Example 10.4.3 with 24 traffic streams and 1536 channels. The maximum number of simultaneous calls of type i (n_i) is in this example $n = 1536$ (full accessibility), and mht is an abbreviation for mean holding time.*

Output	Call congestion	Traffic congestion	Time congestion	Carried traffic	Rel. value
i	B_i	C_i	E_i	Y_i	$C_i/(d_i Z_i)$
1	6.187 744E-03	1.243 705E-03	6.227 392E-03	63.920 403	0.9986
2	6.202 616E-03	3.110 956E-03	6.227 392E-03	63.800 899	0.9991
3	6.227 392E-03	6.227 392E-03	6.227 392E-03	63.601 447	1.0000
4	6.276 886E-03	1.247 546E-02	6.227 392E-03	63.201 570	1.0017
5	6.375 517E-03	2.502 346E-02	6.227 392E-03	62.398 499	1.0046
6	6.570 378E-03	5.025 181E-02	6.227 392E-03	60.783 884	1.0087
7	1.230 795E-02	2.486 068E-03	1.246 554E-02	63.840 892	0.9980
8	1.236 708E-02	6.222 014E-03	1.246 554E-02	63.601 791	0.9991
9	1.246 554E-02	1.246 554E-02	1.246 554E-02	63.202 205	1.0009
10	1.266 184E-02	2.500 705E-02	1.246 554E-02	62.399 549	1.0039
11	1.305 003E-02	5.023 347E-02	1.246 554E-02	60.785 058	1.0083
12	1.379 446E-02	1.006 379E-01	1.246 554E-02	57.559 172	1.0100
13	2.434 998E-02	4.966 747E-03	2.497245E-02	63.682128	0.9970
14	2.458 374E-02	1.244 484E-02	2.497 245E-02	63.203 530	0.9992
15	2.497 245E-02	2.497 245E-02	2.497 245E-02	62.401 763	1.0025
16	2.574 255E-02	5.019 301E-02	2.497 245E-02	60.787 647	1.0075
17	2.722 449E-02	1.006 755E-01	2.497 245E-02	57.556 771	1.0104
18	2.980 277E-02	1.972 682E-01	2.497 245E-02	51.374 835	0.9899
19	4.766 901E-02	9.911 790E-03	5.009 699E-02	63.365 645	0.9948
20	4.858 283E-02	2.489 618E-02	5.009 699E-02	62.406 645	0.9995
21	5.009 699E-02	5.009 699E-02	5.009 699E-02	60.793 792	1.0056
22	5.303 142E-02	1.007 214E-01	5.009 699E-02	57.553 828	1.0109
23	5.818 489E-02	1.981 513E-01	5.009 699E-02	51.318 316	0.9942
24	6.525 455E-02	3.583 491E-01	5.009 699E-02	41.065 660	0.8991
Total		5.950 135E-02		1444.605	

Table 10.7: *Output for Example 10.4.3 with input data given in Tab. 10.6. As mentioned earlier in Example 9.3.2, Fredericks-Hayward's method results in a total congestion equal to 6.114 %. The total traffic congestion 5.950 % is obtained from the total carried traffic and the offered traffic.*

The algorithm is usually called *Kaufman & Roberts' algorithm*, as it was re-discovered by these authors in 1981 (Kaufman, 1981 [57]) (Roberts, 1981 [89]). The same model with *BPP*-traffic and without class limitation has been dealt with by several authors.

Delbrouck's algorithm (Delbrouck, 1983 [22]) is the first and most general of these:

$$q(x) \;=\; \frac{1}{x}\sum_{i=1}^{N}\left\{\frac{d_i\,A_i}{Z_i}\right\}\sum_{j=1}^{\lfloor x/d_i\rfloor} q(x-j\,d_i)\left\{\frac{1-Z_i}{Z_i}\right\}^{j-1}, \quad x=1,2,\ldots n\,, \tag{10.30}$$

$$q(0)=1\,, \qquad q(x)=0 \quad \text{whenever} \quad x<0\,.$$

where $\lfloor x/d_i \rfloor$ denote the integer part of x/d_i. The relative state probabilities $q(i)$ are normalised to obtain the absolute state probabilities:

$$p(x)=\frac{q(x)}{Q_n}\,, \quad Q_n=\sum_{i=0}^{n} q(i)\,, \quad x=0,1,\ldots,n\,. \tag{10.31}$$

For Poisson arrival processes we have $Z_i = 1$, and then we get (10.29) as 0^j is one for $j = 0$ and zero for $j > 0$.

We may rewrite Delbrouck's algorithm as follows by observing that the inner summation is a geometric weighting of previous state probabilities:

$$\delta_i(x) \;=\; \frac{d_i\cdot A_i}{Z_i}\,q(x-d_i)+\delta_i(x-d_i)\cdot\frac{1-Z_i}{Z_i}\,, \quad x=0,1,\ldots,n\,. \tag{10.32}$$

$$\delta_i(x)=0 \quad \text{whenever} \quad x<d_i\,,$$

$$q(x) \;=\; \frac{1}{x}\sum_{i=1}^{N}\delta_i(x)\,, \quad x=1,2,\ldots,n\,, \tag{10.33}$$

$$q(0)=1\,, \qquad q(x)=0 \quad \text{whenever} \quad x<0\,.$$

This modified algorithm only requires a limited number of previous state probabilities. Therefore, we may normalise after each step in the iteration to avoid numerical problems as described Sec. 7.4. In each step we have to normalize not only $q(x)$, but also all $\delta_i(j)$, $i=1,\ldots N$, $j=x,x-1,\ldots,x-d_i$.

Kraimeche & Schwartz (1983 [68]) published a similar algorithm. Based on the theory for queueing networks (Chap. 14) Pinsky & Conway (1992 [83]) published a similar algorithm, which calculates the normalisation constant.

The above-mentioned algorithms require in the general case the same number of operations as the convolution algorithm. Delbrouck's algorithm is only more effective than the convolution algorithm for the Poisson case. The algorithms mentioned in this section cannot be applied to general state dependent Poisson processes, only to BPP–traffic models. In principle, we may apply Delbrouck's algorithm for BPP–traffic to calculate the global state probabilities for the first $(N-1)$ traffic streams and then use the convolution algorithm to calculate the performance measures for the N'th traffic stream. This is only more effective, if there is more than one Poisson traffic stream. By a detailed study of Delbrouck's algorithm we are also able to find call congestion and traffic congestion.

Example 10.4.4: Delbrouck's algorithm
We now apply Delbrouck's algorithm (10.30) to Example 10.3.2. We notice that for the Poisson traffic stream the inner summation in (10.30) becomes just one state (cf. (10.29)) as we know the total offered traffic. By direct application of the algorithm we find the same global state probabilities as above.

$$\begin{aligned}
q(0) &= && = 1 \\
q(1) &= \left\{\tfrac{2}{1}\cdot\tfrac{1}{1}\cdot 1\right\} + \left\{\tfrac{1}{1}\cdot\tfrac{2}{3/4}\cdot 0\right\} && = 2 \\
q(2) &= \left\{\tfrac{2}{2}\cdot\tfrac{1}{1}\cdot 2\right\} + \left\{\tfrac{1}{2}\cdot\tfrac{2}{3/4}\cdot\left[(-\tfrac{1}{3})^0\cdot 1\right]\right\} && = \tfrac{10}{3} \\
q(3) &= \left\{\tfrac{2}{3}\cdot\tfrac{1}{1}\cdot \tfrac{10}{3}\right\} + \left\{\tfrac{1}{3}\cdot\tfrac{2}{3/4}\cdot\left[(-\tfrac{1}{3})^0\cdot 2\right]\right\} && = 4 \\
q(4) &= \left\{\tfrac{2}{4}\cdot\tfrac{1}{1}\cdot 4\right\} + \left\{\tfrac{1}{4}\cdot\tfrac{2}{3/4}\cdot\left[(-\tfrac{1}{3})^0\cdot \tfrac{10}{3} + (-\tfrac{1}{3})^1\cdot 1\right]\right\} && = 4 \\
q(5) &= \left\{\tfrac{2}{5}\cdot\tfrac{1}{1}\cdot 4\right\} + \left\{\tfrac{1}{5}\cdot\tfrac{2}{3/4}\cdot\left[(-\tfrac{1}{3})^0\cdot 4 + (-\tfrac{1}{3})^1\cdot 2\right]\right\} && = \tfrac{152}{45} \\
q(6) &= \left\{\tfrac{2}{6}\cdot\tfrac{1}{1}\cdot \tfrac{152}{45}\right\} + \left\{\tfrac{1}{6}\cdot\tfrac{2}{3/4}\cdot\left[(-\tfrac{1}{3})^0\cdot 4 + (-\tfrac{1}{3})^1\cdot \tfrac{10}{3} + (-\tfrac{1}{3})^2\cdot 1\right]\right\} && = \tfrac{332}{135}
\end{aligned}$$

The relative state probabilities add to $\frac{2723}{135}$. The absolute global state probabilities become identical with the results obtained earlier:

$$\begin{aligned}
p(0) &= \tfrac{135}{2723} = 0.0496\,, \\
p(1) &= \tfrac{270}{2723} = 0.0992\,, \\
p(2) &= \tfrac{450}{2723} = 0.1653\,, \\
p(3) &= \tfrac{540}{2723} = 0.1983\,, \\
p(4) &= \tfrac{540}{2723} = 0.1983\,, \\
p(5) &= \tfrac{456}{2723} = 0.1675\,, \\
p(6) &= \tfrac{332}{2723} = 0.1219\,.
\end{aligned}$$

The time congestion is obtained directly from the global state probabilities, and the call congestion can be obtained for a Binomial–Pascal traffic stream by carrying through the same calculations with one source less. □

Chapter 11

Dimensioning of telecom networks

Network planning includes designing, optimising, and operating telecommunication networks. In this chapter we will consider traffic engineering aspects of network planning. In Sec. 11.1 we introduce traffic matrices and the fundamental double factor method (Kruithof's method) for updating traffic matrices according to forecasts. The traffic matrix contains the basic information for choosing the topology (Sec. 11.2) and traffic routing (Sec. 11.3).

In Sec. 11.4 we consider approximate calculation of end-to-end blocking probabilities, and describe the Erlang fix-point method (reduced load method). Sec. 11.5 generalises the convolution algorithm introduced in Chap. 10 to networks with exact calculation of end-to-end blocking in virtual circuit switched networks with direct routing. The model allows for multi-slot *BPP* traffic with minimum and maximum allocation. The same model can be applied to hierarchical cellular wireless networks with overlapping cells and to optical WDM networks. In Sec. 11.6 we consider service-protection mechanisms. Finally, in Sec. 11.7 we consider optimising of telecommunication networks by applying *Moe's principle*.

11.1 Traffic matrices

To specify the traffic demand in an area with K exchanges we should know K^2 traffic values $A_{ij}(i, j = 1, \ldots, K)$, as given in the *traffic matrix* shown in Tab. 11.1. The traffic matrix assumes we know the location areas of exchanges. Knowing the traffic matrix we have the following two interdependent tasks:

- Decide on the topology of the network (which exchanges should be interconnected ?)

- Decide on the traffic routing (how do we exploit a given topology ?)

	TO							
FROM	1	$\cdots$	i	$\cdots$	j	$\cdots$	K	$A_{i\,\cdot} = \sum_{k=1}^{K} A_{ik}$
1	A_{11}	$\cdots$	A_{1i}	$\cdots$	A_{1j}	$\cdots$	A_{1K}	$A_{1\,\cdot}$
$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\cdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$
i	A_{i1}	$\cdots$	A_{ii}	$\cdots$	A_{ij}	$\cdots$	A_{iK}	$A_{i\,\cdot}$
$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\cdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$
j	A_{j1}	$\cdots$	A_{ji}	$\cdots$	A_{jj}	$\cdots$	A_{jK}	$A_{j\,\cdot}$
$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\cdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$
K	A_{K1}	$\cdots$	A_{Ki}	$\cdots$	A_{Kj}	$\cdots$	A_{KK}	$A_{K\,\cdot}$
$A_{\cdot\,j} = \sum_{k=1}^{K} A_{kj}$	$A_{\cdot\,1}$	$\cdots$	$A_{\cdot\,i}$	$\cdots$	$A_{\cdot\,j}$	$\cdots$	$A_{\cdot\,K}$	$\sum_{i=1}^{K} A_{i\,\cdot} = \sum_{j=1}^{K} A_{\cdot\,j}$

The traffic matrix has the following elements:

$$\begin{aligned}
A_{ij} &= \text{is the traffic from } i \text{ to } j.\\
A_{ii} &= \text{is the internal traffic in exchange } i.\\
A_{i\cdot} &= \text{is the total outgoing (originating) traffic from } i.\\
A_{\cdot j} &= \text{is the total incoming (terminating) traffic to } j.
\end{aligned}$$

Table 11.1: *A traffic matrix. The total incoming traffic is equal to the total outgoing traffic.*

11.1.1 Kruithof's double factor method

Let us assume we know the actual traffic matrix and that we have a forecast for future row sums $O(i)$ and column sums $T(i)$, i.e. the total incoming and outgoing traffic for each exchange. This traffic prognosis may be obtained from subscriber forecasts for the individual exchanges. By means of *Kruithof's double factor method* (Kruithof, 1937 [69]) we are able to estimate the future individual values A_{ij} of the traffic matrix. The procedure is to adjust the individual values A_{ij}, so that they agree with the new row/column sums:

$$A_{ij} \Leftarrow A_{ij} \cdot \frac{S_1}{S_0}\,, \tag{11.1}$$

where S_0 is the actual sum and S_1 is the new sum of the row/column considered. If we start by adjusting A_{ij} with respect to the new row sum S_i, then the row sums will agree, but the column sums will not agree with the wanted values. Therefore, next step is to adjust the obtained values A_{ij} with respect to the column sums so that these agree, but this implies that the row sums no longer agree. By alternatively adjusting row and column sums the values obtained will after a few iterations converge towards unique values. The procedure is best illustrated by an example given below.

Example 11.1.1: Application of Kruithof's double factor method
We consider a telecommunication network having two exchanges. The present traffic matrix is given as:

	1	2	Total
1	10	20	30
2	30	40	70
Total	40	60	100

The prognosis for the total originating and terminating traffic for each exchange is:

	1	2	Total
1			45
2			105
Total	50	100	150

The task is then to estimate the individual values of the matrix by means of the double factor method.

Iteration 1: Adjust the row sums. We multiply the first row by (45/30) and the second row by (105/70) and get:

	1	2	Total
1	15	30	45
2	45	60	105
Total	60	90	150

The row sums are now correct, but the column sums are not.

Iteration 2: Adjust the column sums:

	1	2	Total
1	12.50	33.33	45.83
2	37.50	66.67	104.17
Total	50.00	100.00	150.00

We now have the correct column sums, whereas the column sums deviate a little. We continue by alternately adjusting the row and column sums:

Iteration 3:

	1	2	Total
1	12.27	32.73	45.00
2	37.80	67.20	105.00
Total	50.07	99.93	150.00

Iteration 4:

	1	2	Total
1	12.25	32.75	45.00
2	37.75	67.25	105.00
Total	50.00	100.00	150.00

After four iterations both the row and the column sums agree with two decimals. □

There are other methods for estimating the future individual traffic values A_{ij}, but Kruithof's double factor method has some important properties (Bear, 1988 [5]):

- *Uniqueness*. Only one solution exists for a given forecasts.
- *Reversibility*. The resulting matrix can be reversed to the initial matrix with the same procedure.
- *Transitivity*. The resulting matrix is the same independent of whether it is obtained in one step or via a series of intermediate transformations, (for instance one 5-year forecast, or five 1-year forecasts).
- *Invariance* as regards the numbering of exchanges. We may change the numbering of the exchanges without influencing the results.
- *Fractioning*. The single exchanges can be split into sub-exchanges or be aggregated into larger exchanges without influencing the result. This property is nor exactly fulfilled for Kruithof's double factor method, but the deviations are small.

11.2 Topologies

In Chap. 1 we have described the basic topologies as star net, mesh net, ring net, hierarchical net and non-hierarchical net.

11.3 Routing principles

This is an extensive subject including i.a. alternative traffic routing, load balancing, etc. In (Ash, 1998 [3]) there is a detailed description of this subject.

11.4 Approximate end-to-end calculations methods

If we assume the links of a network are independent, then it is easy to calculate the end-to-end blocking probability. By means of the classical formulæ we calculate the blocking probability of each link. If we denote the blocking probability of link i by E_i, then we find the end-to-end blocking probability for a call attempt on route j as follows:

$$E_j = 1 - \prod_{i \in \mathcal{R}} (1 - E_i)\,, \tag{11.2}$$

where $\mathcal{R}$ is the set of links included in the route of the call. This value will be *worst case*, because the traffic is smoothed by the blocking on each link, and therefore experience less congestion on the last link of a route.

For small blocking probabilities we have:

$$E_j \approx \sum_{i \in \mathcal{R}} E_i\,. \tag{11.3}$$

11.4.1 Fix-point method

A call will usually occupy channels on more links, and in general the traffic on the individual links of a network will be correlated. The blocking probability experienced by a call attempt on the individual links will therefore also be correlated. Erlang's fix-point method is an attempt to take this into account.

11.5 Exact end-to-end calculation methods

Circuit switched telecommunication networks with direct routing have the same complexity as queueing networks with more chains. (Sec. 14.9) and Tab. 14.3). It is necessary to keep account of the number of busy channels on each link. Therefore, the maximum number of states becomes:

$$\prod_{i=1}^{K}(n_i + 1)\,. \tag{11.4}$$

Link	Route 1	Route 2	Route $\cdots$	Route N	Number of channels
1	d_{11}	d_{21}	$\cdots$	d_{N1}	n_1
2	d_{12}	d_{22}	$\cdots$	d_{N2}	n_2
$\cdot$	$\cdot$	$\cdot$		$\cdot$	$\cdot$
$\cdots$	$\cdots$	$\cdots$		$\cdots$	$\cdots$
$\cdot$	$\cdot$	$\cdot$		$\cdot$	$\cdot$
K	d_{1K}	d_{2K}	$\cdots$	d_{NK}	n_K

Table 11.2: *In a circuit switched telecommunication network with direct routing d_{ij} denoted the slot-size (bandwidth demand) of route j upon link i (cf. Tab. 14.3).*

11.5.1 Convolution algorithm

The convolution algorithm described in Chap. 10 can directly be applied to networks with direct routing, because there is product form among the routes. The convolution becomes multi-dimensional, the dimension being the number of links in the network. The truncation of the state space becomes more complex, and the number of states increases very much.

11.6 Load control and service protection

In a telecommunication network with many users competing for the same resources (multiple access) it is important to specify service demands of the users and ensure that the *GoS* is fulfilled under normal service conditions. In most systems it can be ensured that preferential subscribers (police, medical services, etc.) get higher priority than ordinary subscribers when they make call attempts. During normal traffic conditions we want to ensure that all subscribers for all types of calls (local, domestic, international) have approximately the same service level, e.g. 1 % blocking. During overload situations the call attempts of some groups of subscribers should not be completely blocked and other groups of subscribers at the same time experience low blocking. We aim at "the collective misery".

Historically, this has been fulfilled because of the decentralised structure and the application of limited accessibility (grading), which from a service protection point of view still are applicable and useful.

Digital systems and networks have an increased complexity and without preventive measures the carried traffic as a function of the offered traffic will typically have a form similar to the Aloha system (Fig. 6.4). To ensure that a system during overload continues to operate at maximum capacity various strategies are introduced. In stored program controlled systems (exchanges) we may introduce call-gapping and allocate priorities to the tasks (Chap. 13). In telecommunication networks two strategies are common: trunk reservation and virtual channels protection.

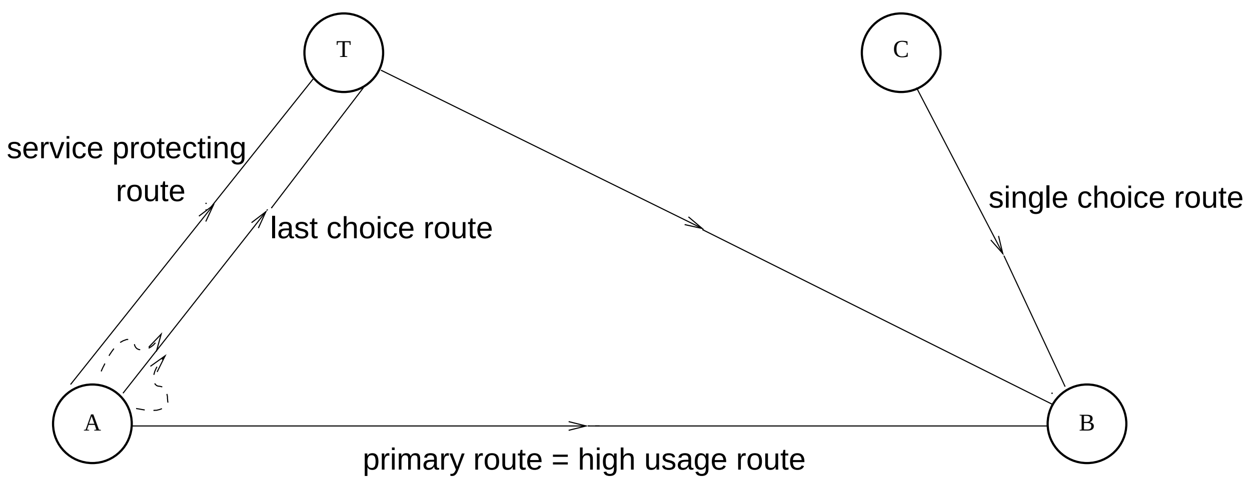


Figure 11.1: *Alternative traffic routing (cf. example 11.6.2). Traffic from A to B is partly carried on the direct route (primary route = high usage route), partly on the secondary route via the transit exchange T.*

11.6.1 Trunk reservation

In hierarchical telecommunication networks with alternative routing we want to protect the primary traffic against overflow traffic. If we consider part of a network (Fig. 11.1), then the direct traffic AT will compete with the overflow traffic from AB for idle channels on the trunk group AT. As the traffic AB already has a direct route, we want to give the traffic AT priority to the channels on the link AT. This can be done by introducing trunk (channel) reservation. We allow the AB–traffic to access the AT–channels only if there are more than r channels idle on AT (r = reservations parameter). In this way, the traffic AT will get higher priority to the AT–channels. If all calls have the same mean holding time ($\mu_1 = \mu_2 = \mu$) and

PCT-I traffic with single slot traffic, then we can easily set up a state transition diagram and find the blocking probability.

If the individual traffic streams have different mean holding times, or if we consider Binomial & Pascal traffic, then we have to set up an N-dimensional state transition diagram which will be non-reversible. Thus we cannot apply the algorithms developed in Chap. 10.

An essential disadvantage by trunk reservation is that it is a local strategy, which only consider one trunk group (link), not the total end-to-end connection. Furthermore, it is a one-way mechanism which protect one traffic stream against the other, but not vice-versa. Therefore, it cannot be applied to mutual protection of connections and services in broadband networks.

Example 11.6.1: Guard channels
In a wireless mobile communication system we may ensure lower blocking probability to hand-over calls than experienced by new call attempts by reserving the last idle channel (called guard channel) to hand-over calls. □

11.6.2 Virtual channel protection

In a service-integrated system it is necessary to protect all services mutually against each other and to guarantee a certain grade-of-service. This can be obtained by (a) a certain minimum allocation of bandwidth which ensures a certain minimum service, and (b) a maximum allocation which both allows for the advantages of statistical multiplexing and ensures that a single service do not dominate. This strategy has the fundamental product form, and the state probabilities are insensitive to the service time distribution. Also, the *GoS* is guaranteed not only on a link basis, but end-to-end.

11.7 Moe's principle

Theorem 11.1 Moe's principle: *the optimal resource allocation is obtained by a simultaneous balancing of marginal incomes and marginal costs over all sectors.*

In this section we present the basic principles published by Moe in 1924. We consider a system with some sectors which consume resources (equipment) for producing items (traffic). The problem can be split into two parts:

a. Given that a limited amount of resources are available, how should we distribute these among the sectors?

b. How many resources should be allocated in total?

The principles are applicable in general for all kind of productions. In our case the resources correspond to cables and switching equipment, and the production consists in carried traffic.

A sector may be a link to an exchange. The problem may be dimensioning of links between a certain exchange and its neighbouring exchanges to which there are direct connections. The problem then is:

a. How much traffic should be carried on each link, when a total fixed amount of traffic is carried?

b. How much traffic should be carried in total?

Question *a* is solved in Sec. 11.7.1 and question *b* in Sec. 11.7.2. We carry through the derivations for continuous variables because these are easier to work with. Similar derivations can be carried through for discreet variables, corresponding to a number of channels. This is Moe's principle (Jensen, 1950 [49]).

11.7.1 Balancing marginal costs

Let us from a given exchange have direct connections to k other exchanges. The cost of a connection to an exchange i is assumed to to be a linear function of the number of channels:

$$C_i = c_{0i} + c_i \cdot n_i, \qquad i = 1, 2, \ldots, k\,. \tag{11.5}$$

The total cost of cables then becomes:

$$C\left(n_1, n_2, \ldots, n_k\right) = C_0 + \sum_{i=1}^{k} c_i \cdot n_i\,, \tag{11.6}$$

where C_0 is a constant.

The total carried traffic is a function of the number of channels:

$$Y = f\left(n_1, n_2, \ldots, n_k\right)\,. \tag{11.7}$$

As we always operate with limited resources we will have:

$$\frac{\partial f}{\partial n_i} = \mathrm{D}_i f > 0\,. \tag{11.8}$$

In a pure loss system $\mathrm{D}_i f$ corresponds to the improvement function, which is always positive for a finite number of channels because of the convexity of Erlang's B–formula.

We want to minimise C for a given total carried traffic Y:

$$\min\{C\} \quad \textit{given} \quad Y = f\,(n_1, n_2, \ldots, n_k)\,. \tag{11.9}$$

By applying the Lagrange multiplier ϑ, where we introduce $G = C - \vartheta \cdot f$, this is equivalent to:

$$\min\left\{G\,(n_1, n_2, \ldots, n_k)\right\} = \min\left\{C\,(n_1, n_2, \ldots, n_k) - \vartheta\,[f\,(n_1, n_2, \ldots, n_k) - Y]\right\} \tag{11.10}$$

A necessary condition for the minimum solution is:

$$\frac{\partial G}{\partial n_i} = c_i - \vartheta\,\frac{\partial f}{\partial n_i} = c_i - \vartheta \mathrm{D}_i f = 0, \qquad i = 1, 2, \ldots, k\,, \tag{11.11}$$

or

$$\frac{1}{\vartheta} = \frac{\mathrm{D}_1 f}{c_1} = \frac{\mathrm{D}_2 f}{c_2} = \cdots = \frac{\mathrm{D}_k f}{c_k}\,. \tag{11.12}$$

A necessary condition for the optimal solution is thus that the marginal increase of the carried traffic when increasing the number of channels (improvement function) divided by the cost for a channel must be identical for all trunk groups (7.31).

It is possible by means of second order derivatives to set up a set of necessary conditions to establish sufficient conditions, which is done in *"Moe's Principle"* (Jensen, 1950 [49]). The improvement functions we deal with will always fulfil these conditions.

If we also have different incomes g_i for the individual trunk groups (directions), then we have to include an additional weight factor, and in the results (11.12) we shall replace c_i by c_i/g_i.

11.7.2 Optimum carried traffic

Let us consider the case where the carried traffic, which is a function of the number of channels (11.7) is Y. If we denote the revenue with $R(Y)$ and the costs with $C(Y)$ (11.6), then the profit becomes:

$$P(Y) = R(Y) - C(Y)\,. \tag{11.13}$$

A necessary condition for optimal profit is:

$$\frac{dP(Y)}{dY} = 0 \qquad \Rightarrow \qquad \frac{dR}{dY} = \frac{dC}{dY}\,, \tag{11.14}$$

i.e. the marginal income should be equal to the marginal cost.

Using:

$$P\,(n_1, n_2, \ldots, n_k) = R\,(f\,(n_1, n_2, \ldots, n_k)) - \left\{C_0 + \sum_{i=1}^{k} c_i \cdot n_i\right\}\,, \tag{11.15}$$

the optimal solution is obtained for:

$$\frac{\partial P}{\partial n_i} = \frac{dR}{dY} \cdot \mathrm{D}_i f - c_i = 0, \qquad i = 1, 2, \ldots, k\,, \tag{11.16}$$

which by using (11.12) gives:

$$\frac{dR}{dY} = \vartheta\,. \tag{11.17}$$

The factor ϑ given by (11.12) is the ratio between the cost of one channel and the traffic which can be carried additionally if the link in extended by one channel. Thus we shall add channels to the link until the marginal income equals the marginal cost ϑ (7.33).

Example 11.7.1: Optimal capacity allocation
We consider two links (trunk groups) where the offered traffic is 3 erlang, respectively 15 erlang. The channels for the two systems have the same cost and there is a total of 25 channels available. How should we distribute the 25 channels among the two links?

From (11.12) we notice that the improvement functions should have the same values for the two directions. Therefore we proceed using a table:

$A_1 = 3$ erlang		$A_2 = 15$ erlang	
n_1	$F_{1,n}(A_1)$	n_2	$F_{1,n}(A_2)$
3	0.4201	17	0.4048
4	0.2882	18	0.3371
5	0.1737	19	0.2715
6	0.0909	20	0.2108
7	0.0412	21	0.1573

For $n_1 = 5$ and $n_2 = 20$ we use all 25 channels. This results in a congestion of 11.0%, respectively 4.6%, i.e. higher congestion for the smaller trunk group. □

Example 11.7.2: Triangle optimisation
This is a classical optimisation of a triangle network using alternative traffic routing (Fig. 11.1). From A to B we have a traffic demand equal to A erlang. The traffic is partly carried on the direct route (primary route) from A to B, partly on an alternative route (secondary route) $A \rightarrow T \rightarrow B$, where T is a transit exchange. There are no other routing possibilities. The cost of a direct connection is c_d, and for a secondary connection c_t.

How much traffic should be carried in each of the two directions? The route $A \rightarrow T \rightarrow B$ already carries traffic to and from other destinations, and we denote the marginal utilisation for a channel on this route by a. We assume it is independent of the additional traffic, which is blocked from A → B.

According to (11.12), the minimum conditions become:

$$\frac{F_{1,n}(A)}{c_d} = \frac{a}{c_t} \, .$$

Here, n is the number of channels in the primary route. This means that the costs should be the same when we route an "additional" call via the direct route and via the alternative route.
If one route were cheaper than the other, then we would route more traffic in the cheaper direction. □

As the traffic values applied as basis for dimensioning are obtained by traffic measurements they are encumbered with unreliability due to a limited sample, limited measuring period, measuring principle, etc. As shown in Chap. 15 the unreliability is approximately proportional to the measured traffic volume. By measuring the same time period for all links we get the highest uncertainty for small links (trunk groups), which is partly compensated by the above-mentioned overload sensitivity, which is smallest for small trunk groups. As a representative value we typically choose the measured mean value plus the standard deviation multiplied by a constant, e.g. 1.0.

To make sure, it should further be emphasised that we dimension the network for the traffic which shall be carried 1–2 years from now. The value used for dimensioning is thus additionally encumbered by a forecast uncertainty. We has not included the fact that part of the equipment may be out of operation because of technical errors.

ITU–T recommends that the traffic is measured during all busy hours of the year, and that we choose n so that by using the mean value of the 30 largest, respectively the 5 largest observations, we get the following blocking probabilities:

$$\begin{aligned} E_n\left(\bar{A}_{30}\right) &\leq 0.01 \, , \\ E_n\left(\bar{A}_{5}\right) &\leq 0.07 \, . \end{aligned} \tag{11.18}$$

The above service criteria can directly be applied to the individual trunk groups. In practise, we aim at a blocking probability from A-subscriber to B-subscriber which is the same for all types of calls. With stored program controlled exchanges the trend is a continuous supervision of the traffic on all expensive and international routes.

In conclusion, we may say that the traffic value used for dimensioning is encumbered with uncertainty. In large trunk groups the application of a non-representative traffic value may result in serious consequences for the grade-of-service level. During later years, there has been an increasing interest for adaptive traffic controlled routing (*traffic network management*), which can be introduce in stored program control digital systems. By this technology we may in principle choose the optimal strategy for traffic routing during any traffic scenario.

Chapter 12

Delay Systems

In this chapter we consider traffic to a system with n identical servers, full accessibility, and an infinite number of waiting positions. When all n servers are busy, an arriving customer joins a queue and waits until a server becomes idle. No customers can be in queue when a server is idle (full accessibility).

We consider the same two traffic cases as in Chaps. 7 & 8.

1. Poisson arrival process (an infinite number of sources) and exponentially distributed service times (*PCT-I*). This is the most important queueing system, called *Erlang's delay system*. Using the notation later introduced in Sec. 13.1, this system is denoted as $M/M/n$. In this system the carried traffic will be equal to the offered traffic as no customers are blocked. The probability of a positive waiting time, mean queue lengths, mean waiting times, carried traffic per channel, and improvement functions will be dealt with in Sec. 12.2. In Sec. 12.3 *Moe's principle* is applied for optimising the system. The waiting time distribution is calculated for the basic service discipline, First–Come First–Served (*FCFS*), in Sec. 12.4.

2. A limited number of sources and exponentially distributed service times (*PCT-II*). This is *Palm's machine repair model* (the machine interference problem) which is dealt with in Sec. 12.5. This model has been widely applied for dimensioning for example computer systems, terminal systems, and flexible manufacturing system (*FMS*). Palm's machine repair model is optimised in Sec. 12.6.

12.1 Erlang's delay system $M/M/n$

Let us consider a queueing system $M/M/n$ with Poisson arrival process (M), exponential service times (M), n servers and an infinite number of waiting positions. The state of the system is defined as the total number of customers in the system (either being served or

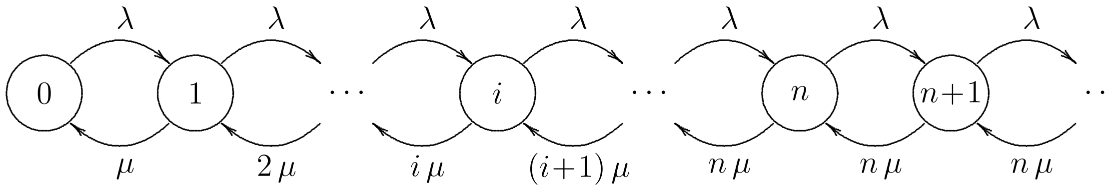


Figure 12.1: *State transition diagram of the M/M/n delay system having n servers and an unlimited number of waiting positions.*

waiting in queue). We are interested in the steady state probabilities of the system. By the procedure described in Sec. 7.4 we set up the state transition diagram shown in Fig. 12.1. Assuming statistical equilibrium, the cut equations become:

$$\begin{aligned}
\lambda \cdot p(0) &= \mu \cdot p(1)\,, \\
\lambda \cdot p(1) &= 2\,\mu \cdot p(2)\,, \\
\vdots \quad & \vdots \quad \vdots \\
\lambda \cdot p(i) &= (i+1)\,\mu \cdot p(i+1)\,, \\
\vdots \quad & \vdots \quad \vdots \\
\lambda \cdot p(n-1) &= n\,\mu \cdot p(n)\,, \\
\lambda \cdot p(n) &= n\,\mu \cdot p(n+1)\,, \\
\vdots \quad & \vdots \quad \vdots \\
\lambda \cdot p(n+j) &= n\,\mu \cdot p(n+j+1)\,.
\end{aligned} \tag{12.1}$$

As $A = \lambda/\mu$ is the offered traffic, we get:

$$p(i) = \begin{cases} p(0) \cdot \dfrac{A^i}{i!}\,, & 0 \le i \le n\,, \\[2ex] p(n) \cdot \left(\dfrac{A}{n}\right)^{i-n} = p(0) \cdot \dfrac{A^i}{n! \cdot n^{\,i-n}}\,, & i \ge n\,. \end{cases} \tag{12.2}$$

By normalisation of the state probabilities we obtain $p(0)$:

$$1 = \sum_{i=0}^{\infty} p(i)\,,$$

$$1 = p(0) \cdot \left\{ 1 + \frac{A}{1} + \frac{A^2}{2!} + \cdots + \frac{A^n}{n!} \left(1 + \frac{A}{n} + \frac{A^2}{n^2} + \ldots \right) \right\}.$$

The innermost brackets have a geometric progression with quotient A/n. The normalisation condition can only be fulfilled for:

$$A < n\,. \tag{12.3}$$

Statistical equilibrium is only obtained for $A < n$. Otherwise, the queue will continue to increase against infinity.

We obtain:

$$p(0) = \frac{1}{\displaystyle\sum_{i=0}^{n-1} \frac{A^i}{i!} + \frac{A^n}{n!}\,\frac{n}{n-A}}\,, \qquad A < n\,. \tag{12.4}$$

Equations (12.2) and (12.4) yield the steady state probabilities.

12.2 Traffic characteristics of delay systems

For evaluation of the capacity and performance of the system, several characteristics have to be considered. They are expressed by the steady state probabilities.

12.2.1 Erlang's C-formula

When the Poisson arrival process is independent of the state of the system, the probability that an arbitrary arriving customer has to wait in the queue is equal to the proportion of time all servers are occupied (*PASTA*–property: Poisson Arrivals See Time Average). The waiting time is a random variable denoted by $\mathcal{W}$. For an arbitrary arriving customer we have:

$$\begin{aligned} E_{2,n}(A) &= p\{\mathcal{W} > 0\} \\ &= \frac{\displaystyle\sum_{i=n}^{\infty} \lambda\, p(i)}{\displaystyle\sum_{i=0}^{\infty} \lambda\, p(i)} = \sum_{i=n}^{\infty} p(i) \\ &= p(n) \cdot \frac{n}{n-A}\,. \end{aligned} \tag{12.5}$$

Erlang's C–formula:

$$E_{2,n}(A) = \frac{\dfrac{A^n}{n!}\,\dfrac{n}{n-A}}{1 + \dfrac{A}{1} + \dfrac{A^2}{2!} + \cdots + \dfrac{A^{n-1}}{(n-1)!} + \dfrac{A^n}{n!}\,\dfrac{n}{n-A}}\,, \qquad A < n\,. \tag{12.6}$$

This delay probability depends only upon A, the product of λ and s, not upon the parameters λ and s individually. The formula has several names: *Erlang's C–formula*, *Erlang's second formula*, or *Erlang's formula for waiting time systems*. It has various notations in literature:

$$E_{2,n}(A) = D = D_n(A) = p\{W > 0\}\,.$$

As customers are either served immediately or put into queue, the probability that a customer is served immediately becomes:

$$S_n = 1 - E_{2,n}(A)\,.$$

The carried traffic Y equals the offered traffic A, as no customers are rejected and the arrival process is a Poisson process:

$$\begin{aligned} Y &= \sum_{i=1}^{n} i\, p(i) + \sum_{i=n+1}^{\infty} n\, p(i) \qquad (12.7)\\ &= \sum_{i=1}^{n} \frac{\lambda}{\mu}\, p(i-1) + \sum_{i=n+1}^{\infty} \frac{\lambda}{\mu}\, p(i-1)\\ &= \frac{\lambda}{\mu} = A\,, \end{aligned}$$

where we have exploited the cut balance equations.

The queue length is a random variable $\mathcal{L}$. The probability of having customers in queue at a random point of time is:

$$\begin{aligned} p\{\mathcal{L} > 0\} &= \sum_{i=n+1}^{\infty} p(i) = \frac{\frac{A}{n}}{1-\frac{A}{n}} \cdot p(n)\,,\\ p\{\mathcal{L} > 0\} &= \frac{A}{n-A}\, p(n) = \frac{A}{n}\, E_{2,n}(A)\,. \qquad (12.8) \end{aligned}$$

where we have used (12.5).

12.2.2 Numerical evaluation

The formula is similar to Erlang's B-formula (7.10) except for the factor $n/(n-A)$ in the last term. As we have very accurate recursive formulæ for numerical evaluation of Erlang's B-formula (7.27) we use the following relationship for obtaining numerical values of the C-formula:

$$\begin{aligned} E_{2,n}(A) &= \frac{n \cdot E_{1,n}(A)}{n - A\,(1 - E_{1,n}(A))} \\ &= \frac{E_{1,n}(A)}{1 - A\,\{1 - E_{1,n}(A)\}\,/n}\,, \qquad A < n\,. \end{aligned} \tag{12.9}$$

We notice that:

$$E_{2,n}(A) > E_{1,n}(A)\,,$$

as the term $A\{1 - E_{1,n}(A)\}/n$ is the average carried traffic per channel in the corresponding loss system. For $A \geq n$, we have $E_{2,n}(A) = 1$ as it is a probability and all customers are delayed.

Erlang's C-formula may in an elegant way be expressed by the B-formula as noticed by B. Sanders:

$$\frac{1}{E_{2,n}(A)} = \frac{1}{E_{1,n}(A)} - \frac{1}{E_{1,n-1}(A)}\,, \tag{12.10}$$

$$I_{2,n}(A) = I_{1,n}(A) - I_{1,n-1}(A)\,, \tag{12.11}$$

where I is the inverse probability (7.28):

$$I_{2,n}(A) = \frac{1}{E_{2,n}(A)}\,.$$

Erlang's C-formula has been tabulated in *Moe's Principle* (Jensen, 1950 [49]) and is shown in Fig. 12.2.

12.2.3 Mean queue lengths

We distinguish between the queue length at an arbitrary point of time and the queue length when there are customers waiting in the queue.

Mean queue length at an arbitrary point of time:

The queue length $\mathcal{L}$ at an arbitrary point of time is called the virtual queue length. This is the queue length experienced by an arbitrary customer as the *PASTA*-property is valid due to the Poisson arrival process (time average = call average). We get the mean queue length

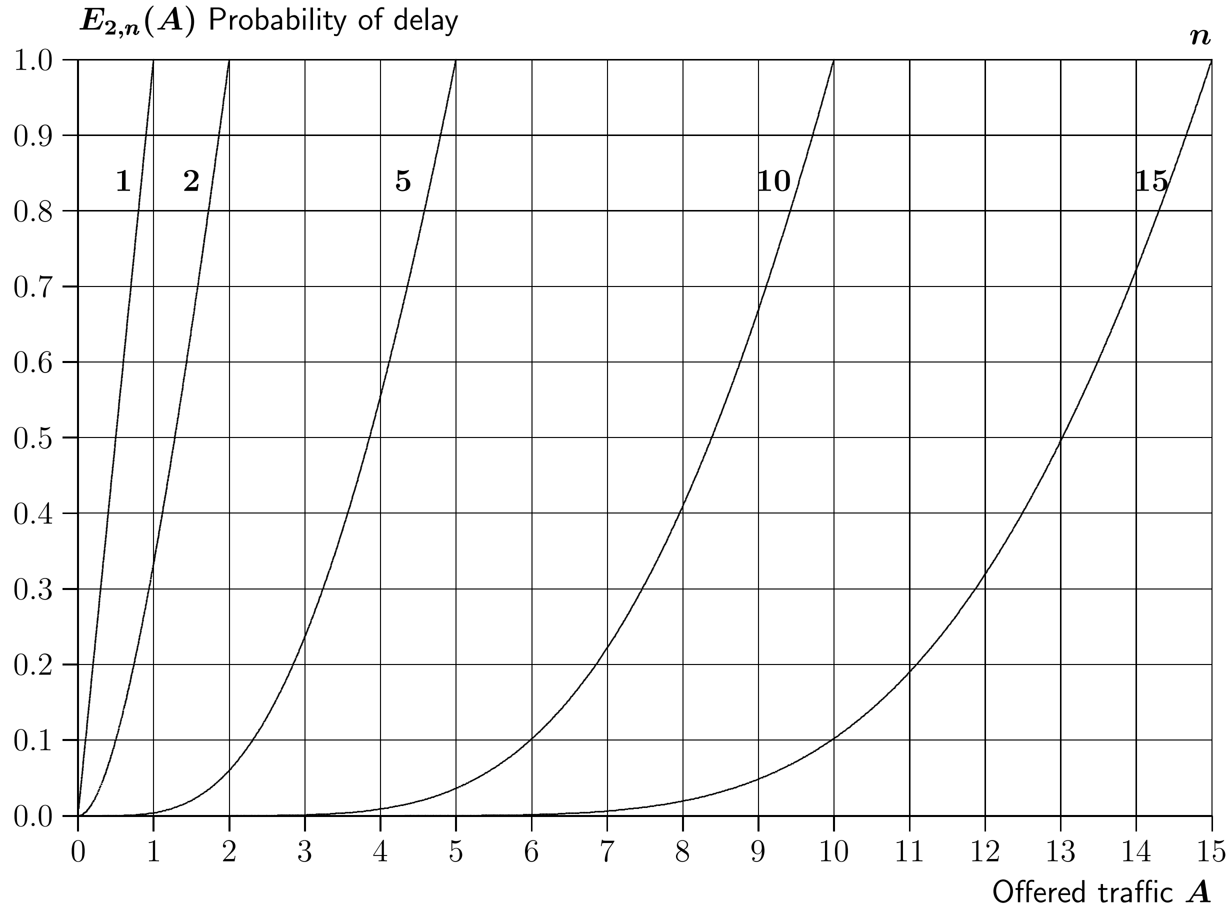


Figure 12.2: *Erlang's C–formula for the delay system M/M/n. The probability $E_{2,n}(A)$ for a positive waiting time is shown as a function of the offered traffic A for different values of the number of servers n.*

$L_n = E\{\mathcal{L}\}$ at an arbitrary point of time:

$$
\begin{aligned}
L_n &= 0 \cdot \sum_{i=0}^{n} p(i) + \sum_{i=n+1}^{\infty} (i-n)\, p(i) \\
&= \sum_{i=n+1}^{\infty} (i-n)\; p(n) \left(\frac{A}{n}\right)^{i-n} \\
&= p(n) \cdot \sum_{i=1}^{\infty} i \left(\frac{A}{n}\right)^{i} \\
&= p(n) \cdot \frac{A}{n} \sum_{i=1}^{\infty} \frac{\partial}{\partial(A/n)} \left\{ \left(\frac{A}{n}\right)^{i} \right\}.
\end{aligned}
$$

As $A/n \leq c < 1$, the series is uniformly convergent, and the differentiation operator may be put outside the summation:

$$\begin{aligned} L_n &= p(n)\,\frac{A}{n}\,\frac{\partial}{\partial(A/n)}\left\{\frac{A/n}{1-(A/n)}\right\} = p(n)\cdot\frac{A/n}{\{1-(A/n)\}^2} \\ &= p(n)\cdot\frac{n}{n-A}\cdot\frac{A}{n-A}\,, \\ L_n &= E_{2,n}(A)\cdot\frac{A}{n-A}\,. \end{aligned} \tag{12.12}$$

The average queue length may be interpreted as the traffic carried by the queueing positions and therefore it is also called the *waiting time traffic*.

Mean queue length, given the queue is greater than zero:

The time average is also in this case equal to the call average. The conditional mean queue length becomes:

$$\begin{aligned} L_{nq} &= \frac{\displaystyle\sum_{i=n+1}^{\infty}(i-n)\,p(i)}{\displaystyle\sum_{i=n+1}^{\infty}p(i)} \\ &= \frac{p(n)\cdot\dfrac{A/n}{(1-A/n)^2}}{p(n)\,\dfrac{A}{n-A}} \\ &= \frac{n}{n-A} \end{aligned} \tag{12.13}$$

By applying (12.8) and (12.12) this is of course the same as:

$$L_{nq} = \frac{L_n}{p\{\mathcal{L}>0\}}\,,$$

where $\mathcal{L}$ is the random variable for queue length.

12.2.4 Mean waiting times

Here also two items are of interest: the mean waiting time W for all customers, and the mean waiting time w for customers experiencing a positive waiting time. The first one is an indicator for the service level of the whole system, whereas the second one is of importance for the customers, which are delayed. Time averages will be equal to call averages because of the *PASTA*-property.

Mean waiting time W for all customers:

Little's theorem tells that the average queue length is equal to the arrival intensity multiplied by the mean waiting time:

$$L_n = \lambda\, W_n\,. \tag{12.14}$$

where $L_n = L_n(A)$, and $W_n = W_n(A)$. From (12.12) we get by considering the arrival process:

$$W_n = \frac{L_n}{\lambda} = \frac{1}{\lambda} \cdot E_{2,n}(A) \cdot \frac{A}{n-A}\,.$$

As $A = \lambda\, s$, where s is the mean service time, we get:

$$W_n = E_{2,n}(A) \cdot \frac{s}{n-A}\,. \tag{12.15}$$

Mean waiting time w for delayed customers:

The total waiting time is constant and may either be averaged over all customers (W_n) or only over customers, which experience positive waiting times (w_n) (3.20):

$$W_n = w_n \cdot E_{2,n}(A)\,, \tag{12.16}$$

$$w_n = \frac{s}{n-A}\,. \tag{12.17}$$

Example 12.2.1: Single server queueing system M/M/1
This is the system appearing most often in the literature. The state probabilities (12.2) are given by a geometric series:

$$p(i) = (1-A) \cdot A^i, \qquad i = 0, 1, 2, \ldots, \tag{12.18}$$

as $p(0) = 1-A$. The probability of delay become:

$$E_{2,1}(A) = A\,.$$

The mean queue length L_n (12.12) and the mean waiting time for all customers W_n (12.15) become:

$$L_1 = \frac{A^2}{1-A}\,, \tag{12.19}$$

$$W_1 = \frac{A\,s}{1-A}\,. \tag{12.20}$$

From this we observe that an increase in the offered traffic results in an increase of L_n by the third power, independent of whether the increase is due to an increased number of customers (λ) or an increased service time (s). The mean waiting time W_n increases by the third power of s, but only by the second power of λ. The mean waiting time w_n for delayed customers increases with the second power of s, and the first power of λ. An increased load due to more customers is thus better than an increased load due to longer service times. Therefore, it is important that the service times of a system do not increase during overload. $\square$

Example 12.2.2: Mean waiting time w when $A \rightarrow 0$
Notice, that as $A \rightarrow 0$, we get $w_n = s/n$ (12.17). If a customer experiences waiting time (which seldom happens when $A \rightarrow 0$), then this customer will be the only one in the queue. The customer must wait until a server becomes idle. This happens after an exponentially distributed time interval with mean value s/n. So w_n never becomes less than s/n. $\square$

12.2.5 Improvement functions for $M/M/n$

The marginal improvement of the traffic carried when we add one server can be expressed in several ways. The decrease in the proportion of total traffic (= the proportion of all customers) that experience delay is given by:

$$F_{2,n}(A) = A\left\{E_{2,n}(A) - E_{2,n+1}(A)\right\} . \tag{12.21}$$

The decrease in mean queue length (= traffic carried by the waiting positions) becomes by using Little's law (12.14):

$$\begin{aligned} F_{L,n}(A) &= L_n(A) - L_{n+1}(A) \\ &= \lambda\left\{W_n(A) - W_{n+1}(A)\right\} , \end{aligned} \tag{12.22}$$

where $W_n(A)$ is the mean waiting time for all customers when the offered traffic is A and the number of servers is n (12.15). Both (12.21) and (12.22) are tabulated in *Moe's Principle* (Jensen, 1950 [49]) and are simple to evaluate by a calculator or computer.

12.3 Moe's principle for delay systems

Moe first derived his principle for queueing systems. He studied the subscribers waiting times for an operator at the manual exchanges in Copenhagen Telephone Company.

Let us consider k independent queueing systems. A customer being served at all k systems has the total average waiting time $\sum_i W_i$, where W_i is the mean waiting time of i'th system which has n_i servers and is offered the traffic A_i. The cost of a channel is c_i, eventually plus a constant cost, which is included in the constant C_0 below. Thus the total costs for channels becomes:

$$C = C_0 + \sum_{i=1}^{k} n_i\, c_i\,. \tag{12.23}$$

If the waiting time also is considered as a cost, then the total costs to be minimised becomes $f = f(n_1, n_2, \ldots, n_k)$. This is to be minimised as a function of number of channels n_i in the individual systems. If the total average waiting time is W, then the allocation of channels to the individual systems is determined by:

$$\min \{f(n_1, n_2, \ldots, n_k)\} = \min \left\{ C_0 + \sum_i n_i \, c_i + \vartheta \cdot \left(\sum_i W_i - W \right) \right\} . \tag{12.24}$$

where ϑ (theta) is Lagrange's multiplier.

As n_i is integral, a necessary condition for minimum, which in this case also can be shown to a be sufficient condition, becomes:

$$\begin{aligned} 0 &< f(n_1, n_2, \ldots, n_i - 1, \ldots, n_k) - f(n_1, n_2, \ldots, n_i, \ldots, n_k) \,, \\ 0 &\geq f(n_1, n_2, \ldots, n_i, \ldots, n_k) - f(n_1, n_2, \ldots, n_i + 1, \ldots, n_k) \,, \end{aligned} \tag{12.25}$$

which corresponds to:

$$\begin{aligned} W_{n_i - 1}(A_i) - W_{n_i}(A_i) &> \frac{c_i}{\vartheta} \,, \\ W_{n_i}(A_i) - W_{n_i + 1}(A_i) &\leq \frac{c_i}{\vartheta} \,, \end{aligned} \tag{12.26}$$

where $W_{n_i}(A_i)$ is given by (12.15).

Expressed by the improvement function for the waiting time $F_{W,n}(A)$ (12.22) the optimal solution becomes:

$$F_{W,n_i-1}(A) > \frac{c_i}{\vartheta} \geq F_{W,n_i}(A) \,, \qquad i = 1, 2, \ldots k \,. \tag{12.27}$$

The function $F_{W,n}(A)$ is tabulated in *Moe's Principle* (Jensen, 1950 [49]). Similar optimisations can be carried out for other improvement functions.

Example 12.3.1: Delay system
We consider two different $M/M/n$ queueing systems. The first one has a mean service time of 100 s and the offered traffic is 20 erlang. The cost-ratio c_1/ϑ is equal to 0.01. The second system has a mean service time equal to 10 s and the offered traffic is 2 erlang. The cost ration equals $c_2/\vartheta = 0.1$. A table of the improvement function $F_{W,n}(A)$ gives:

$$\begin{aligned} n_1 &= 32 \text{ channels and} \\ n_2 &= 5 \text{ channels.} \end{aligned}$$

The mean waiting times are:

$$\begin{aligned} W_1 &= 0.075 \text{ s.} \\ W_2 &= 0.199 \text{ s.} \end{aligned}$$

This shows that a customer, who is served at both systems, experience a total mean waiting time equal to 0.274 s, and that the system with less channels contributes more to the mean waiting time.
□

The cost of waiting is related to the cost ratio. By investing one monetary unit more in the above system, we reduce the costs by the same amount independent of in which queueing system we increase the investment. We should go on investing as long as we make profit.
Moe's investigations during 1920's showed that the mean waiting time for subscribers at small exchanges with few operators should be larger than the mean waiting time at larger exchanges with many operators.

12.4 Waiting time distribution for *M/M/n, FCFS*

Queueing systems, where the service discipline only depends upon the arrival times, all have the same mean waiting times. In this case the strategy has only influence upon the distribution of waiting times for the individual customer. The derivation of the waiting time distribution is simple in the case of ordered queue, *FCFS* = First–Come First–Served. This discipline is also called *FIFO*, First–In First–Out. Customers arriving first to the system will be served first, but if there are multiple servers they may not necessarily leave the server first. So *FIFO* refers to the time for leaving the queue and initiating service.

Let us consider an arbitrary customer. Upon arrival to the system, the customer is either served immediately or has to wait in the queue (12.6).
We now assume that the customer considered has to wait in the queue, i.e. the system may be in state $[n+k]$, $(k = 0, 1, 2, \ldots)$, where k is the number of occupied waiting positions just before the arrival of the customer.

Our customer has to wait until $k+1$ customers have completed their service before an idle server becomes accessible. When all n servers are working, the system completes customers with a *constant* rate $n\,\mu$, i.e. the departure process is a Poisson process with this intensity.

We exploit the relationship between the number representation and the interval representation (5.4): The probability $p\{\mathcal{W} \leq t\} = F(t)$ of experiencing a positive waiting time less than or equal to t is equal to the probability that in a Poisson arrival process with intensity $(n\,\mu)$ at least $(k+1)$ customers arrive during the interval t (6.1):

$$F(t \,|\, k \text{ waiting}) = \sum_{i=k+1}^{\infty} \frac{(n\mu t)^i}{i!} \cdot \mathrm{e}^{-n\mu t} \,. \tag{12.28}$$

The above was based on the assumption that our customer has to wait in the queue. The conditional probability that our customer when arriving observes all n servers busy and k

waiting customers ($k = 0, 1, 2, \cdots$) is:

$$
\begin{aligned}
p_w(k) &= \frac{\lambda\, p(n+k)}{\lambda \displaystyle\sum_{i=0}^{\infty} p(n+i)} = \frac{p(n)\cdot\left(\dfrac{A}{n}\right)^k}{p(n)\cdot \displaystyle\sum_{i=0}^{\infty}\left(\dfrac{A}{n}\right)^i} \\
&= \left(1-\frac{A}{n}\right)\left(\frac{A}{n}\right)^k, \qquad k = 0, 1, \ldots\,. \qquad (12.29)
\end{aligned}
$$

This is a geometric distribution including the zero class (Tab. 6.1). The unconditional waiting time distribution then becomes:

$$
F(t) = \sum_{k=0}^{\infty} p_w(k)\cdot F(t\,|\,k)\,, \qquad (12.30)
$$

$$
\begin{aligned}
F(t) &= \sum_{k=0}^{\infty}\left\{\left(1-\frac{A}{n}\right)\left(\frac{A}{n}\right)^k \cdot \sum_{i=k+1}^{\infty}\frac{(n\mu t)^i}{i!}\,\mathrm{e}^{-n\mu t}\right\} \\
&= \mathrm{e}^{-n\mu t}\sum_{i=1}^{\infty}\left\{\frac{(n\mu t)^i}{i!}\cdot\sum_{k=0}^{i-1}\left(1-\frac{A}{n}\right)\left(\frac{A}{n}\right)^k\right\},
\end{aligned}
$$

as we may interchange the two summations when all terms are positive probabilities. The inner summation is a geometric progression:

$$
\begin{aligned}
\sum_{k=0}^{i-1}\left(1-\frac{A}{n}\right)\left(\frac{A}{n}\right)^k &= \left(1-\frac{A}{n}\right)\cdot\sum_{k=0}^{i-1}\left(\frac{A}{n}\right)^k \\
&= \left(1-\frac{A}{n}\right)\cdot 1\cdot\frac{1-(A/n)^i}{1-(A/n)} \\
&= 1-\left(\frac{A}{n}\right)^i.
\end{aligned}
$$

Inserting this we obtain:

$$\begin{aligned}
F(t) &= \mathrm{e}^{-n\mu t} \cdot \sum_{i=1}^{\infty} \frac{(n\mu t)^i}{i!} \left\{ 1 - \left(\frac{A}{n}\right)^i \right\} \\
&= \mathrm{e}^{-n\mu t} \left\{ \sum_{i=0}^{\infty} \frac{(n\mu t)^i}{i!} - \sum_{i=0}^{\infty} \frac{(n\mu t)^i}{i!} \left(\frac{A}{n}\right)^i \right\} \\
&= i\mathrm{e}^{-n\mu t} \left\{ \mathrm{e}^{n\mu t} - \mathrm{e}^{n\mu t \cdot A/n} \right\} ,
\end{aligned}$$

$$F(t) = 1 - \mathrm{e}^{-(n-A)\mu t} , \tag{12.31}$$

$$F(t) = 1 - \mathrm{e}^{-(n\mu - \lambda)\, t} , \qquad n > A , \quad t > 0 . \tag{12.32}$$

i.e. an exponential distribution. Apparently we have a paradox: when arriving at a system with all servers busy one may:

1. Count the number k of waiting customers ahead. The total waiting time will then be Erlang–$(k+1)$ distributed.
2. Close the eyes. Then the waiting time becomes exponentially distributed.

The interpretation of this is that a weighted sum of Erlang distributions with geometrically distributed weight factors is equivalent to an exponential distribution. In Fig. 12.3 the phase-diagram for (12.30) is shown, and we notice immediately that it can be reduced to a single exponential distribution (Sec. 4.4.2 & Fig. 4.9). Formula (12.31) confirms that the mean waiting time w_n for customers who have to wait in the queue becomes as shown in (12.17). The waiting time distribution for all (an arbitrary customer) becomes (3.19):

$$F_s(t) = 1 - E_{2,n}(A) \cdot \mathrm{e}^{-(n-A)\mu\, t} , \quad A < n , \quad t \geq 0 , \tag{12.33}$$

and the mean value of this distribution is W_n in agreement with (12.15). The results may be derived in an easier way by means of generation functions.

12.4.1 Response time with a single server

When there is only one server, the state probabilities (12.2) are given by a geometric series (12.18), i.e. $p(i) = p(0) \cdot A^i$ for all $i \geq 0$. Every customer spends an exponentially distributed time interval with intensity μ in every state. A customer who finds the system in state $[\,i\,]$ shall stay in the system an Erlang-$(i+1)$ distributed time interval. Therefore, the *total* sojourn

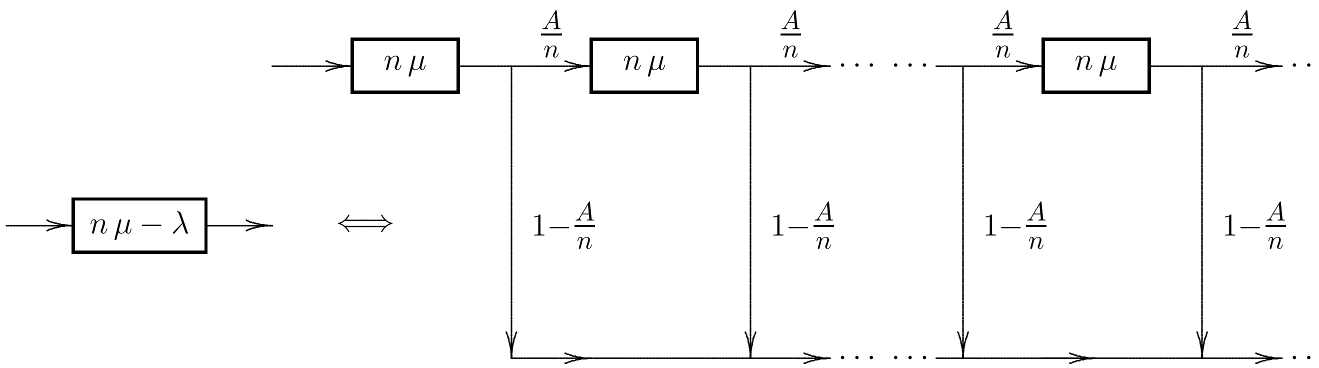


Figure 12.3: *The waiting time distribution for M/M/n–FCFS becomes exponentially distributed with intensity* $(n\mu-\lambda)$. *The phase-diagram to the left corresponds to a weighted sum of Erlang-k distributions (Sec. 4.4.2) as the rate out of all phases is* $n\mu\,(1-A/n) = n\mu-\lambda$.

time in the system (waiting time + service time), i.e. the *response time*, is exponentially distributed with intensity $(\mu - \lambda)$ (cf. Fig. 4.9):

$$F(t) = 1 - \mathrm{e}^{-(\mu - \lambda)t}\,, \quad \mu > \lambda\,, \quad t \geq 0\,. \tag{12.34}$$

This is identical with the waiting time distribution of delayed customers.

12.5 Palm's machine repair model

This model belongs to the class of *cyclic queueing systems* and corresponds to a pure delay system with a limited number of customers (cf. Engset case for loss systems).

The model was first considered by Gnedenko in 1933 and published in 1934. It became widely known when C. Palm published a paper in 1947 [80] in connection with a theoretical analysis of manpower allocation for servicing automatic machines. S machines, which usually run automatically, are serviced by n repairmen. The machines may break down and then they have to be serviced by a repairman before running again. The problem is to adjust the number of repairmen to the number of machines so that the total costs are minimised (or the profit optimised). The machines may be textile machines which stop when they run out of thread; the repairmen then have to replace the empty spool of a machine with a full one.

This *Machine-Repair* model or *Machine Interference* model was also considered in (Feller, 1950 [27]). The model corresponds to a simple closed queueing network and has been successfully applied to solve traffic engineering problems in computer systems. By using Kendall's notation (Chap. 13) the queueing system is denoted by $M/M/n/S/S$, where S is the number of customers, and n is the number of servers.

The model is widely applicable. In the Web the machines corrrespond to *clients* whereas the repairmen correspond to servers. In computer terminal systems the machines correspond to

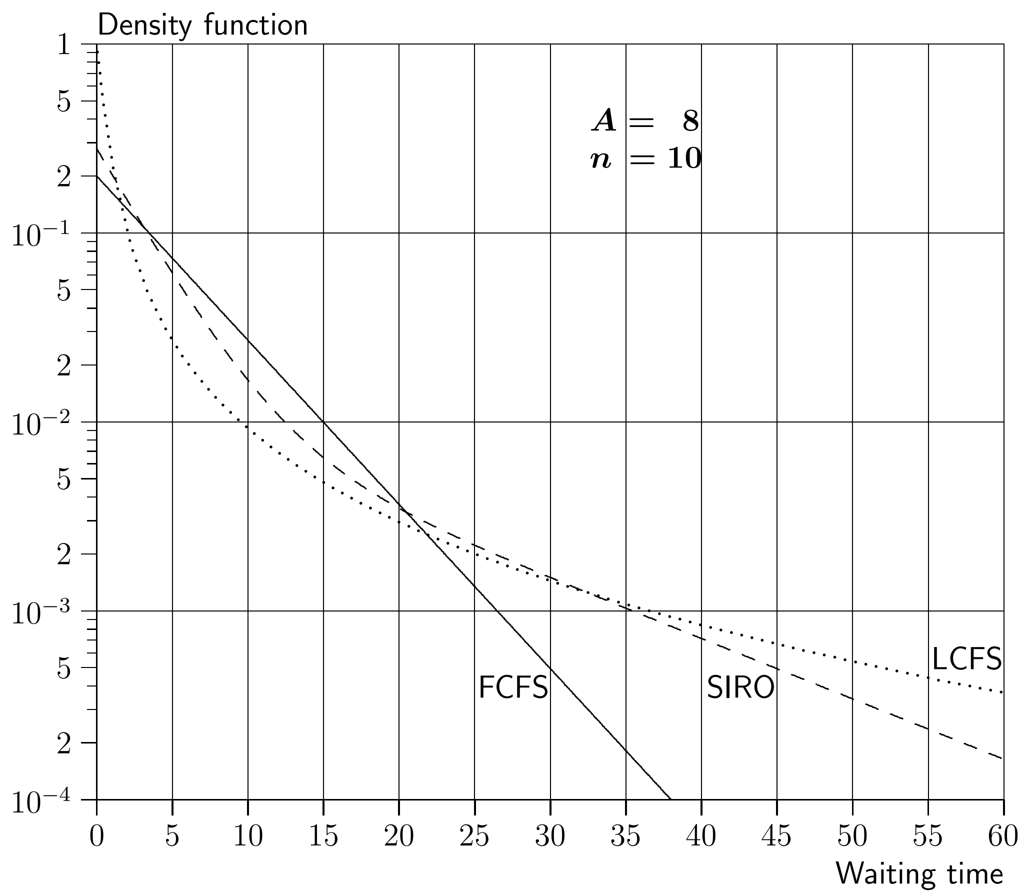


Figure 12.4: *Density function for the waiting time distribution for the queueing discipline FCFS, LCFS, and SIRO (RANDOM). For all three cases the mean waiting time for delayed calls is 5 time-units. The form factor is 2 for FCFS, 3.33 for LCFS, and 10 for SIRO. The number of servers is 10 and the offered traffic is 8 erlang. The mean service time is* $s = 10$ *time-units.*

the terminals and a repairman corresponds to the computer managing the terminals. In a computer system the machine may correspond to a disc storage and the repairmen correspond to input/output (I/O) channels. In the following we will consider a computer terminal system as the background for the development of the theory.

12.5.1 Terminal systems

Time division is an aid in offering optimal service to a large group of customers using for example terminals connected to a mainframe computer. The individual user should feel that he is the only user of the computer (Fig. 12.5).

The individual terminal changes all the time between two states (interactive) (Fig. 12.6):

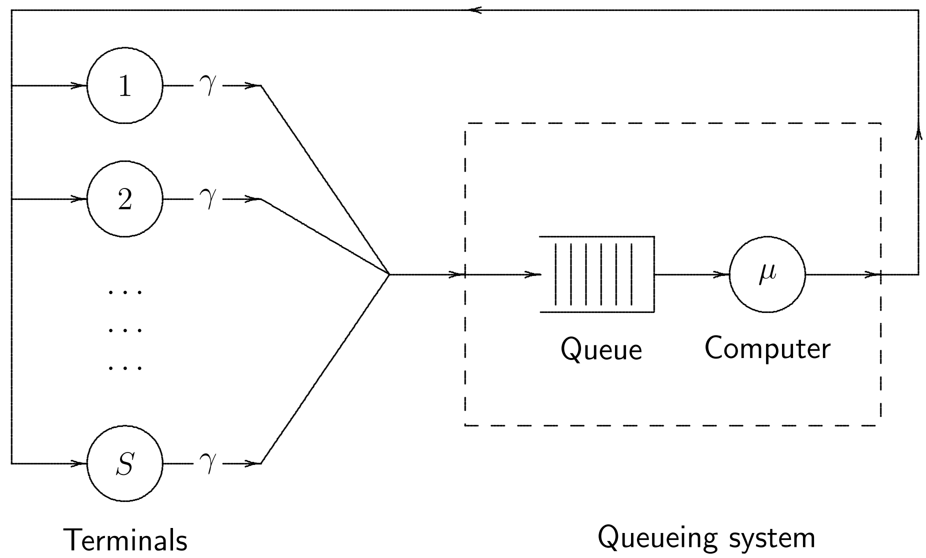


Figure 12.5: *Palm's machine-repair model. A computer system with S terminals (an interactive system) corresponds to a waiting time system with a limited number of sources (cf. Engset-case for loss systems).*

- the user is thinking (working), or
- the user is waiting for a response from the computer.

The time interval the user is thinking is a random variable T_t with mean value m_t. The time interval, when the user is waiting for the response from the computer, is called the *response time* R. This includes both the time interval T_w (mean value m_w), where the job is waiting for getting access to the computer, and the service time itself T_s (mean value m_s).

$T_t + R$ is called the *circulation time* (Fig. 12.6). At the end of this time interval the terminal returns to the same state as it left at the beginning of the interval (recurrent event). In the following we are mainly interested in mean values, and the derivations are valid for all work-conserving queueing disciplines (Sec. 13.4.2).

12.5.2 State probabilities – single server

We now consider a system with S terminals, which are connected to one computer. The thinking times for every thinking terminal are so far assumed to be exponentially distributed with the intensity $\gamma = 1/m_t$, and the service (execution) time m_s at the computer is also assumed to be exponentially distributed with intensity $\mu = 1/m_s$. When there is queue at the computer, the terminals have to wait for service. Terminals being served or waiting un queue have arrival intensity null.

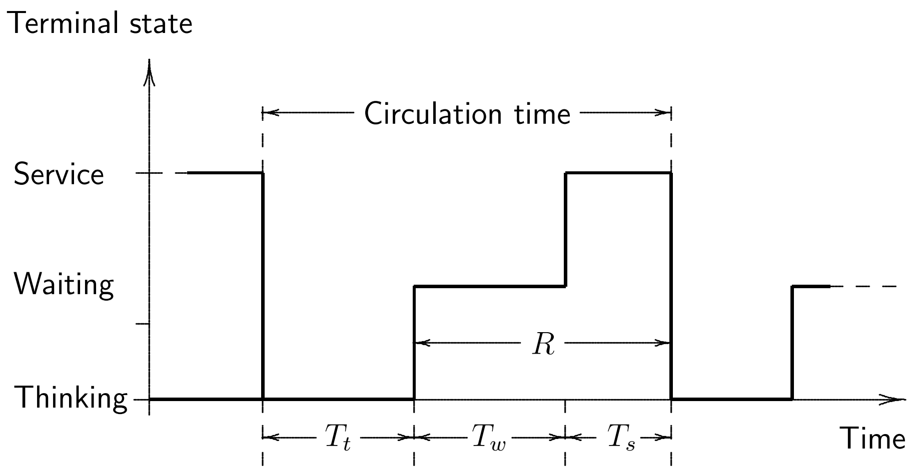


Figure 12.6: *The individual terminal may be in three different states. Either the user is working actively at the terminal (thinking), or he is waiting for response from the computer. The latter time interval (response time) is divided into two phases: a waiting phase and a service phase.*

State $[\,i\,]$ is defined as the state, where there are i terminals in the queueing system (Fig. 12.5), i.e. the computer is either idle ($i = 0$) or working ($i > 0$), and $(i-1)$ terminals are waiting when ($i > 0$).

The queueing system can be modelled by a pure birth and death process, and the state transition diagram is shown in Fig. 12.7. Statistical equilibrium always exists (ergodic system). The arrival intensity decreases as the queue length increases and becomes zero when all terminals are inside the queueing system.

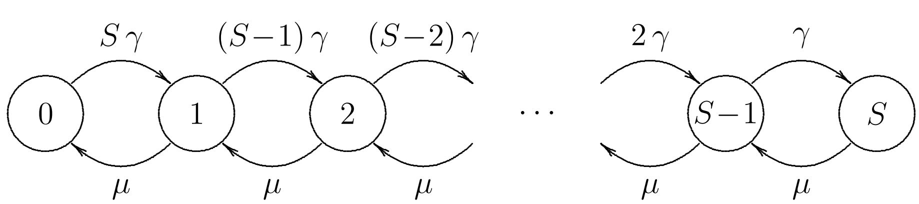


Figure 12.7: *State transition diagram for the queueing system shown in 12.5. State $[\,i\,]$ denotes the number of terminals being either served or waiting, i.e. $S-i$ denotes the number of terminals thinking.*

The steady state probabilities are found by applying cut equations to Fig. 12.7 and expressing all states in terms of state S:

$$(S-i)\,\gamma \cdot p(i) = \mu \cdot p(i+1), \quad i = 0, 1, \ldots, S\,. \tag{12.35}$$

By the additional normalization constraint that the sum of all probabilities must be equal to one we find, introducing $\varrho = \mu/\gamma$:

$$\begin{aligned} p(S-i) &= \frac{\varrho^i}{i!}\, p(S) \\ &= \frac{\dfrac{\varrho^i}{i!}}{\displaystyle\sum_{j=0}^{S} \frac{\varrho^j}{j!}}, \quad i = 0, 1, \ldots, S\,, \qquad (12.36) \\ p(0) &= E_{1,S}(\varrho)\,. \qquad (12.37) \end{aligned}$$

This is the *truncated Poisson distribution* (7.9).

We may interpret the system as follows. A trunk group with S trunks (the terminals) is offered calls from the computer with the exponentially distributed inter-arrival times (intensity μ). When all S trunks are busy (thinking), the computer is idle and the arrival intensity is zero, but we might just as well assume it still generates calls with intensity μ which are lost or overflow to another trunk group (the exponential distribution has no memory). The computer thus offers the traffic $\varrho = \mu/\gamma$ to S trunks, and we have the formula (12.37). Erlang's B-formula is valid for arbitrary holding times (Sec. 7.3.3) and therefore we have:

Theorem 12.1 *The state probabilities of the machine repair model (12.36) & (12.37) with one computer and S terminals is valid for arbitrary thinking times when the service times of the computer are exponentially distributed.*

The ratio $\varrho = \mu/\gamma$ between the time a terminal on average is thinking $1/\gamma$ and the time the computer on average serves a terminal $1/\mu$, is called the *service ratio*. The service ratio corresponds to the offered traffic A in Erlang's B-formula. The state probabilities are thus determined by the number of terminals S and the service ratio ϱ. The numerical evaluation of (12.36) & (12.37) is of course as for Erlang's B-formula (7.27).

Example 12.5.1: Information system
We consider an information system which is organised as follows. All information is kept on 6 discs which are connected to the same input/output data terminal, a multiplexer channel. The average seek time (positioning of the seek-arm) is 3 ms and the average latency time to locate the file is 1 ms, corresponding to a rotation time of 2 ms. The reading time to a file is exponentially distributed with a mean value 0.8 ms. The disc storage is based on rotational positioning sensing, so that the channel is busy only during the reading. We want to find the maximum capacity of the system (number of requests per second).

The thinking time is 4 ms and the service time is 0.8 ms. The service ratio thus becomes 5, and Erlang's B-formula gives the value:

$$1 - p(0) = 1 - E_{1,6}(5) = 0.8082\,.$$

This corresponds to $\gamma_{max} = 0.8082/0.0008 = 1010$ requests per second. □

12.5.3 Terminal states and traffic characteristics

The performance measures are easily obtained from the analogy with Erlang's classical loss system (12.37). The computer is working with the probability $\{1-p(0)\}$. We then have that the average number of terminals being under service by the computer is given by:

$$n_s = 1 - p(0)\,. \tag{12.38}$$

The average number of thinking terminals corresponds to the traffic carried in Erlang's loss system:

$$n_t = \frac{\mu}{\gamma}\,\{1-p(0)\} = \varrho\,\{1-p(0)\}. \tag{12.39}$$

The average number of waiting terminals becomes:

$$\begin{aligned} n_w &= S - n_s - n_t = S - \{1-p(0)\} - \varrho \cdot \{1-p(0)\} \\ &= S - \{1-p(0)\}\{1+\varrho\}\,. \end{aligned} \tag{12.40}$$

If we consider a random terminal at a random point of time, we get:

$$p\{\text{terminal served}\} = p_s = \frac{n_s}{S} = \frac{1-p(0)}{S}\,, \tag{12.41}$$

$$p\{\text{terminal thinking}\} = p_t = \frac{n_t}{S} = \frac{\varrho\,(1-p(0))}{S}\,, \tag{12.42}$$

$$p\{\text{terminal waiting}\} = p_w = \frac{n_w}{S} = 1 - \frac{\{1-p(0)\}\{1+\varrho\}}{S}\,. \tag{12.43}$$

We are also interested in the response time R which has the mean value $m_r = m_w + m_s$. By applying Little's theorem $L = \lambda W$ to terminals, waiting positions and computer, respectively, we obtain (denoting the circulation rate of jobs by λ):

$$\frac{1}{\lambda} = \frac{m_t}{n_t} = \frac{m_w}{n_w} = \frac{m_s}{n_s} = \frac{m_r}{n_w + n_s}\,, \tag{12.44}$$

or

$$m_r = \frac{n_w + n_s}{n_s} \cdot m_s = \frac{S - n_t}{n_s} \cdot m_s\,.$$

Making use of (12.38) and (12.44) $\left\{\dfrac{n_t}{n_s} = \dfrac{m_t}{m_s}\right\}$ we get:

$$\begin{aligned} m_r &= \frac{S}{n_s} \cdot m_s - m_t \\ m_r &= \frac{S}{1-p(0)} \cdot m_s - m_t\,. \end{aligned} \tag{12.45}$$

Thus the mean response time is independent of the time distributions as it is based on (12.38) and (12.44) (Little's Law). However, $p(0)$ will depend on the types of distributions in the same way as the Erlang-B formula. If the service time of the computer is exponentially distributed (mean value $m_s = 1/\mu$), then p(0) will be given by (12.37). Fig. 12.8 shows the response time as a function of the number of terminals in this case.

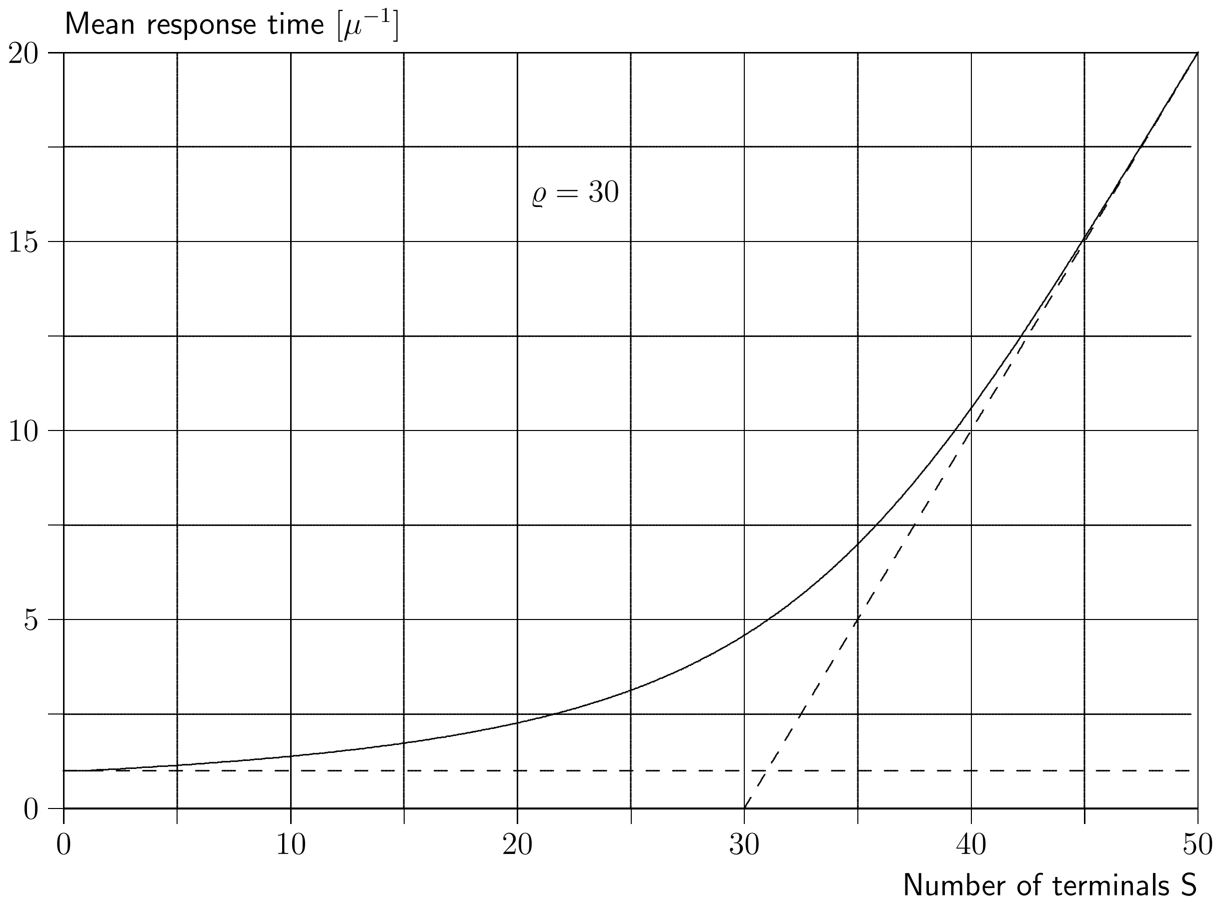


Figure 12.8: *The actual average response time experienced by a terminal as a function of the number of terminals. The service-factor is $\varrho = 30$. The average response time converges to a straight line, cutting the x-axes in $S = 30$ terminals. The average virtual response time for a system with S terminals is equal to the actual average response time for a system with $S+1$ terminals (the Arrival theorem, theorem 8.1).*

If all time intervals are constant, the computer may work all the time serving K terminals without any delay when:

$$\begin{aligned} K &= \frac{m_t + m_s}{m_s} \\ &= \varrho + 1 \,. \end{aligned} \tag{12.46}$$

K is a suitable parameter to describe the point of saturation of the system. The average waiting time for an arbitrary terminal is obtained from (12.45):

$$m_w = m_r - m_s$$

Example 12.5.2: Time sharing computer
In a terminal system the computer sometimes becomes idle (waiting for terminals) and the terminals sometimes wait for the computer. Few terminals result in a low utilisation of the computer, whereas many terminals connected will waste the time of the users.
Fig. 12.9 shows the *waiting time traffic* in erlang, both for the computer and for a single terminal. An appropriate weighting by costs and summation of the waiting times for both the computer and for all terminals gives the total costs of waiting.
For the example in Fig. 12.9 we obtain the minimum total delay costs for about 45 terminals when the cost of waiting for the computer is hundred times the cost of one terminal. At 31 terminals both the computer and each terminal spends 11.4 % of the time for waiting. If the cost ratio is 31, then 31 is the optimal number of terminals. However, there are several other factors to be taken into consideration. □

Example 12.5.3: Traffic congestion
We may define the traffic congestion in the usual way (Sec. 2.3). The offered traffic is the traffic carried when there is no queue. The offered traffic per source is (8.8):

$$\alpha = \frac{\beta}{1+\beta} = \frac{m_s}{m_t + m_s}$$

The carried traffic per source is:

$$a = \frac{m_s}{m_t + m_w + m_s} \,.$$

The traffic congestion becomes:

$$\begin{aligned} C &= \frac{\alpha - a}{\alpha} \\ &= 1 - \frac{m_t + m_s}{m_t + m_w + m_s} = \frac{m_w}{m_t + m_w + m_s}\,, \\ C &= p_w \end{aligned}$$

In this case with finite number of sources the traffic congestion becomes equal to the proportion of time spent waiting. For Erlang's waiting time system the traffic congestion is zero because all offered traffic is carried. □

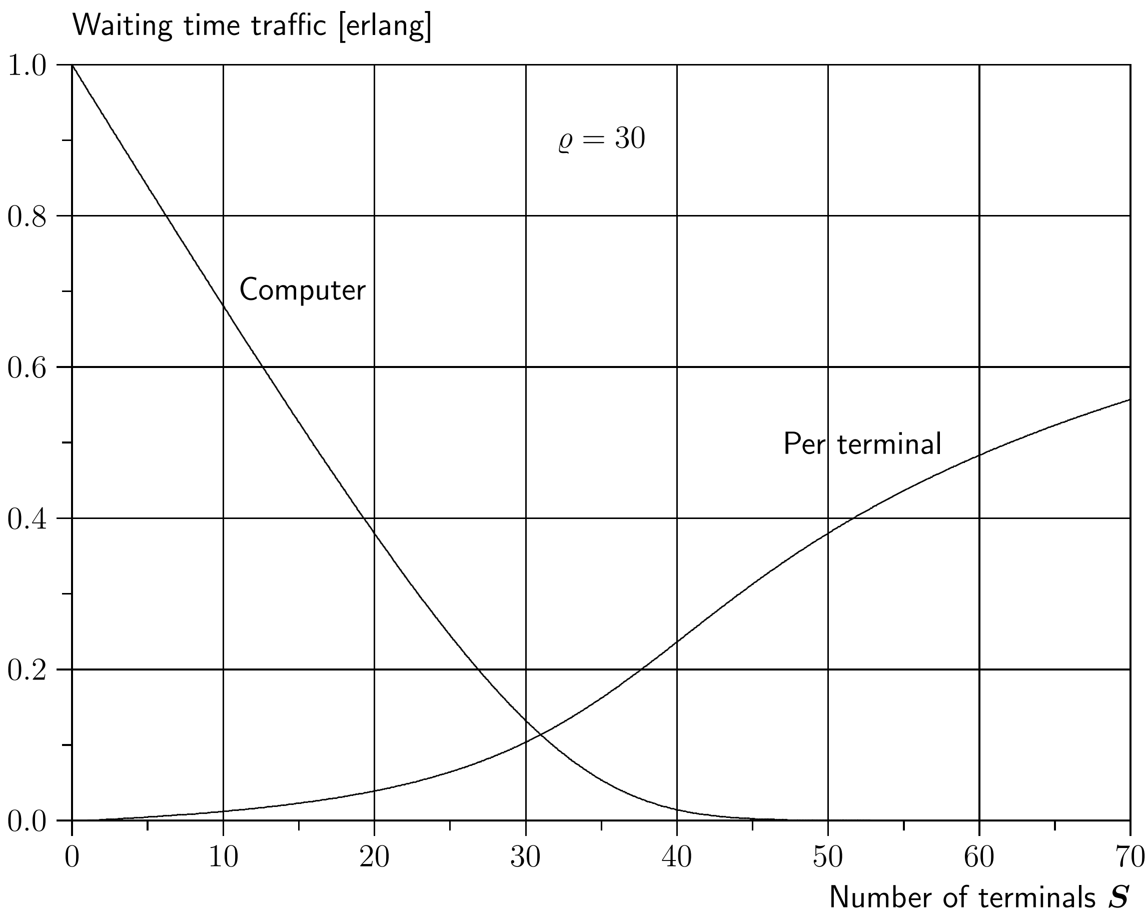


Figure 12.9: *The waiting time traffic (the proportion of time spend waiting) measured in erlang for the computer, respectively the terminals in an interactive queueing system (Service factor $\varrho = 30$).*

12.5.4 Machine–repair model with n servers

The above model is easily generalised to n computers. The transition diagram is shown in Fig. 12.10.

The steady state probabilities become:

$$p(i) = \binom{S}{i} \left(\frac{\gamma}{\mu}\right)^i p(0)\,, \qquad 0 \leq i \leq n\,,$$

$$p(i) = \frac{(S-n)!}{(S-i)!} \left(\frac{\gamma}{n\mu}\right)^{i-n} \cdot p(n)\,, \qquad n \leq i \leq S\,. \tag{12.47}$$

where we have the normalisation constraint:

$$\sum_{i=0}^{S} p(i) = 1\,. \tag{12.48}$$

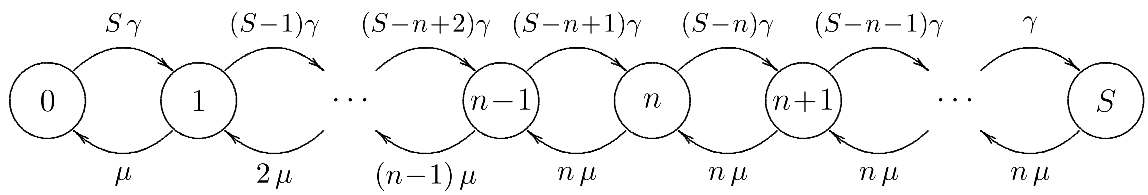


Figure 12.10: *State transition diagram for the machine-repair model with S terminals and n computers.*

We can show that the state probabilities are insensitive to the thinking time distribution as in the case with one computer. (We get a state-dependent Poisson arrival process).

An arbitrary terminal is at a random point of time in one of the three possible states:

$$p_s = p\,\{\textit{the terminal is served by a computer}\},$$

$$p_w = p\,\{\textit{the terminal is waiting for service}\},$$

$$p_t = p\,\{\textit{the terminal is thinking}\}.$$

We have:

$$p_s = \frac{1}{S}\left\{\sum_{i=0}^{n} i \cdot p(i) + \sum_{i=n+1}^{S} n \cdot p(i)\right\}, \tag{12.49}$$

$$p_t = p_s \cdot \frac{\mu}{\gamma}, \tag{12.50}$$

$$p_w = 1 - p_s - p_t\,. \tag{12.51}$$

The mean utilisation of the computers becomes:

$$a = \frac{p_s}{n} \cdot S = \frac{n_s}{n}\,. \tag{12.52}$$

The mean waiting time for a terminal becomes:

$$M = \frac{p_w}{p_s} \cdot \frac{1}{\mu}\,. \tag{12.53}$$

Sometimes p_w is called the loss coefficient of the terminals, and similarly $(1-a)$ is called the loss coefficient of the computers (Fig. 12.9).

Example 12.5.4: Numerical example of scale of economy
The following numerical examples illustrate that we obtain the highest utilisation for large values of n (and S). Let us consider a system with $S/n = 30$ and $\mu/\gamma = 30$ for a increasing number of computers (in this case $p_t = a$).

$\boldsymbol{n}$	1	2	4	8	16
$\boldsymbol{p_s}$	0.0289	0.0300	0.0307	0.0313	0.0316
$\boldsymbol{p_w}$	0.1036	0.0712	0.0477	0.0311	0.0195
$\boldsymbol{p_t}$	0.8675	0.8989	0.9215	0.9377	0.9489
$\boldsymbol{a}$	0.8675	0.8989	0.9215	0.9377	0.9489
$\boldsymbol{W}\ [\boldsymbol{\mu^{-1}}]$	3.5805	2.3754	1.5542	0.9945	0.6155

□

12.6 Optimising the machine-repair model

In this section we optimise the machine/repair model in the same way as Palm did in 1947. We have noticed that the model for a single repair-man is identical with Erlang's loss system, which we optimised in Chap. 7. We will thus see that the same model can be optimised in several ways.

We consider a terminal system with one computer and S terminals, and we want to find an optimal value of S. We assume the following structure of costs:

$$\begin{aligned} c_t &= \textit{cost per terminal per time unit a terminal is thinking}, \\ c_w &= \textit{cost per terminal per time unit a terminal is waiting}, \\ c_s &= \textit{cost per terminal per time unit a terminal is served}, \\ c_a &= \text{cost of the computer per time unit.} \end{aligned}$$

The cost of the computer is supposed to be independent of the utilisation and is split uniformly among all terminals.

The outcome (product) of the process is a certain thinking time at the terminals (production time).

The total costs c_0 per time unit a terminal is thinking (producing) becomes:

$$p_t \cdot c_0 = p_t \cdot c_t + p_s \cdot c_s + p_w \cdot c_w + \frac{1}{S} \cdot c_a \,. \tag{12.54}$$

We want to minimise c_0. The service ratio $\varrho = m_t/m_s$ is equal to p_t/p_s. Introducing the cost

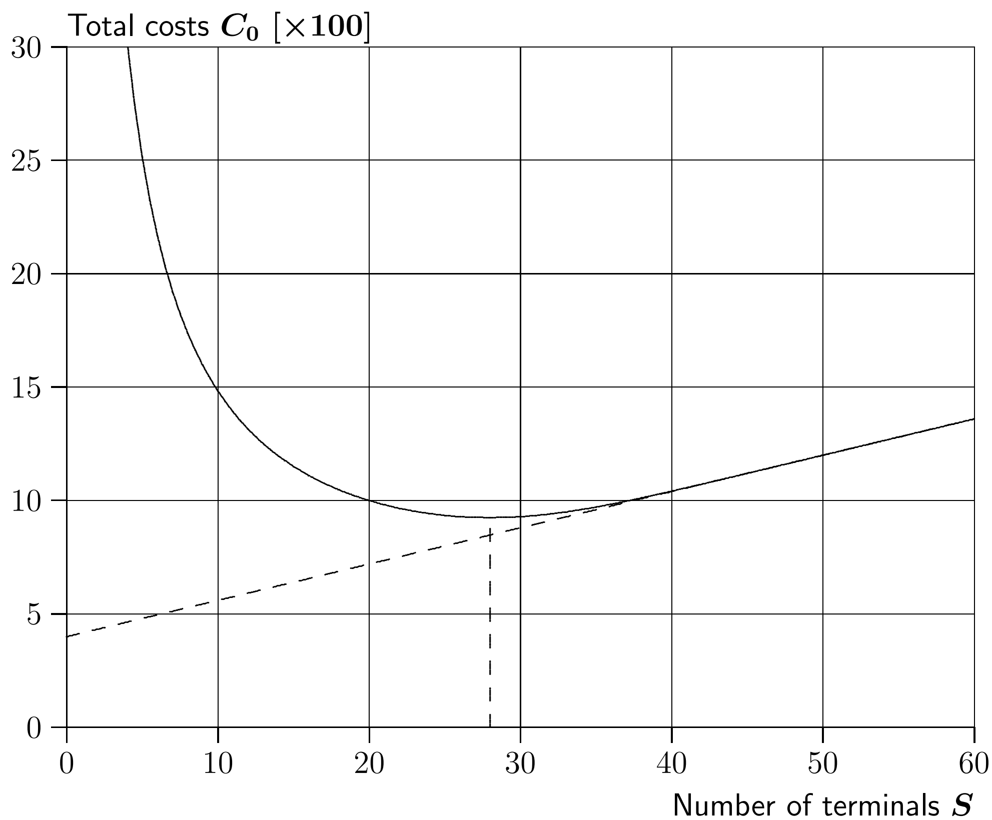


Figure 12.11: *The machine/repair model. The total costs given in (12.57) are shown as a function of number of terminals for a service ratio $\varrho = 25$ and a cost ratio $r = 1/25$ (cf. Fig. 7.6).*

ratio $r = c_w/c_a$, we get:

$$\begin{aligned} c_0 &= c_t + \frac{p_s}{p_t} \cdot c_s + \frac{p_w \cdot c_w + \frac{1}{S} \cdot c_a}{p_t} \\ &= c_t + \frac{1}{\varrho} \cdot c_s + c_a \cdot \frac{r \cdot p_w + (1/S)}{p_t}, \end{aligned} \tag{12.55}$$

which is to be minimised as a function of S. Only the last term depends on the number of terminals and we get:

$$\begin{aligned}
\min_S \{c_0\} &= \min_S \left\{ \frac{r \cdot p_w + (1/S)}{p_t} \right\} \\
&= \min_S \left\{ \frac{r \cdot (n_w/S) + (1/S)}{n_t/S} \right\} \\
&= \min_S \left\{ \frac{r \cdot n_w + 1}{n_t} \right\} \qquad (12.56) \\
&= \min_S \left\{ \frac{r\,[S - \{1 - p(0)\}\,\{1 + \varrho\}] + 1}{\{1 - p(0)\} \cdot \varrho} \right\} \\
&= \min_S \left\{ \frac{r \cdot S + 1}{\{1 - p(0)\} \cdot \varrho} + 1 + \frac{1}{\varrho} \right\}, \qquad (12.57)
\end{aligned}$$

where $p(0)$ is given by Erlang's B-formula (12.36).

We notice that the minimum is independent of c_t and c_s, and that only the ratio $r = c_w/c_a$ appears. The numerator corresponds to (7.29), whereas the denominator corresponds to the carried traffic in the corresponding loss system. Thus we minimise the cost per carried erlang in the corresponding loss system. In Fig. 12.11 an example is shown. We notice that the result deviates from the result obtained by using Moe's Principle for Erlang's loss system (Fig. 7.6), where we optimise the profit.

ng me

Chapter 13

Applied Queueing Theory

Till now we have considered classical queueing systems, where all traffic processes are birth and death processes. The theory of loss systems has been successfully applied for many years within the field of telephony, whereas the theory of delay systems has been applied within the field of data and computer systems. The classical queueing systems play a key role in queueing theory. Usually, we assume that either the inter-arrival time distribution or the service time distribution is exponentially distributed. For theoretical and physical reasons, queueing systems with only one server are often analysed and widely applied.

In this chapter we first concentrate on the single server queue and analyse this system for general service time distributions, various queueing disciplines, and for customers with priorities.

13.1 Classification of queueing models

In this section we shall introduce a compact notations for queueing systems, called Kendall's notation.

13.1.1 Description of traffic and structure

D.G. Kendall (1951 [60]) has introduced the following notation for queueing models:

$$A/B/n$$

where

A = arrival process,
B = service time distribution,

n = number of servers.

For traffic processes we use the following standard notations (cf. Sec. 4.5):

M	$\sim$	Markovian. Exponential time intervals (Poisson arrival process, exponentially distributed service times).
D	$\sim$	Deterministic. Constant time intervals.
E_k	$\sim$	Erlang-k distributed time intervals ($\mathrm{E}_1 = \mathrm{M}$).
H_n	$\sim$	Hyper-exponential of order n distributed time intervals.
Cox	$\sim$	Cox-distributed time intervals.
Ph	$\sim$	Phase-type distributed time intervals.
GI	$\sim$	General Independent time intervals, renewal arrival process.
G	$\sim$	General. Arbitrary distribution of time intervals (may include correlation).

Example 13.1.1: Ordinary queueing models

M/M/n is a pure delay system with Poisson arrival process, exponentially distributed service times and n servers. It is the classical Erlang delay system (Chap. 12).

GI/G/1 is a general delay system with only one server. □

The above mentioned notation is widely used in the literature. For a complete specification of a queueing system more information is required:

$$A/B/n/K/S/X$$

where:

K	=	the total capacity of the system, or only the number of waiting positions,
S	=	the population size (number of customers),
X	=	queueing discipline (Sec. 13.1.2).

$K = n$ corresponds to a loss system, which is often denoted as $A/B/n$–*Loss*.

A superscript b on A, respectively B, indicates group arrival (bulk arrival, batch arrival), respectively group service. C (Clocked) may indicate that the system operates in discrete time. Full accessibility is usually assumed.

13.1.2 Queueing strategy: disciplines and organisation

Customers in a queue waiting to be served can be selected for service according to many different principles. We first consider the three classical queueing disciplines:

FCFS: *First Come – First Served.*
It is also called a fair queue or an ordered queue, and this discipline is often preferred in real-life when customers are human beings. It is also denoted as *FIFO: First In – First Out*. Note that *FIFO* refers to the queue only, not to the total system. If we have more than one server, then a customer with a short service time may overtake a customer with a long waiting time even if we have *FIFO* queue.

LCFS: *Last Come – First Served.*
This corresponds to the stack principle. It is for instance used in storages, on shelves of shops etc. This discipline is also denoted as *LIFO: Last In – First Out*.

SIRO: *Service In Random Order.*
All customers waiting in the queue have the same probability of being chosen for service. This is also called *RANDOM* or *RS* (Random Selection).

The first two disciplines only take *arrival times* into considerations, while the third does not consider any criteria at all and so does not require any memory (contrary to the first two).

They can be implemented in simple technical systems. Within an electro-mechanical telephone exchange the queueing discipline *SIRO* was often used as it corresponds (almost) to sequential hunting without homing.

For the three above-mentioned disciplines the total waiting time for all customers is the same. The queueing discipline only decides how waiting time is allocated to the individual customers. In a program-controlled queueing system there may be more complicated queueing disciplines. In queueing theory we in general assume that the total offered traffic is independent of the queueing discipline.

For computer systems we often try to reduce the total waiting time. It can be done by using the *service time* as criterion:

SJF: *Shortest Job First (SJN = Shortest Job Next, SPF = Shortest Processing time First)*. This discipline assumes that we know the service time in advance and it *minimises the total waiting time* for all customers.

The above mentioned disciplines take account of either the arrival times or the service times. A compromise between these disciplines is obtained by the following disciplines:

RR: *Round Robin.*
A customer served is given at most a fixed service time (time slice or slot). If the service is not completed during this interval, the customer returns to the queue which is *FCFS*.

PS: *Processor Sharing.*
All customers share the service capacity equally.

FB: *Foreground – Background.*
This discipline tries to implement *SJF* without knowing the service times in advance. The server will offer service to the customer who so far has received the least amount of service. When all customers have obtained the same amount of service, *FB* becomes identical with *PS*.

The last mentioned disciplines are dynamic as the queueing disciplines depend on the amount of time spent in the queue.

13.1.3 Priority of customers

In real life customers are often divided into N priority classes, where a customer belonging to class p has higher priority than a customer belonging to class $p+1$. We distinguish between two types of priority:

Non-preemptive = HOL:
A new arriving customer with higher priority than a customer being served waits until a server becomes idle (and all customers with higher priority have been served). This discipline is also called *HOL = Head-Of-the-Line*.

Preemptive:
A customer being served having lower priority than a new arriving customer is interrupted. We distinguish between:

- *Preemptive resume = PR:*
  The service is continued from, where it was interrupted,
- *Preemptive without re-sampling:*
  The service restarts from the beginning with the same service time, and
- *Preemptive with re-sampling:*
  The service starts again with a new service time.

The two latter disciplines are applied in for example manufacturing systems and reliability. Within a single class, we have the disciplines mentioned in Sec. 13.1.2.

In queueing literature we meet many other strategies and symbols. *GD* denotes an arbitrary queueing discipline (general discipline). The behaviour of customers is also subject to modelling:

– *Balking* refers to queueing systems, where customers with a queue dependent probability may give up joining the queue.

– *Reneging* refers to systems with impatient customers which depart from the queue without being served.

– *Jockeying* refers to the systems where the customers may jump from one (e.g. long) queue to another (e.g. shorter) queue.

Thus there are many different possible models. In this chapter we shall only deal with the most important ones. Usually, we only consider systems with one server.

Example 13.1.2: Stored Program Controlled (SPC) switching system
In *SPC*–systems tasks of the processors are divided into for example ten priority classes. The priority is updated for example every $5th$ millisecond. Error messages from a processor have the highest priority, whereas routine tasks of control have the lowest priority. Serving accepted calls has higher priority than detection of new call attempts. □

13.2 General results in the queueing theory

As mentioned earlier there are many different queueing models, but unfortunately there are only few general results in the queueing theory. The literature is very extensive, because many special cases are important in practice. In this section we shall look at the most important general results.

Little's theorem presented in Sec. 5.3 is the most general result which is valid for an arbitrary queueing system. The theorem is easy to apply and very useful in many cases.

In general only queueing systems with Poisson arrival processes are simple to deal with. Concerning queueing systems in series and queueing networks (e.g. computer networks) it is important to know cases, where the departure process from a queueing system is a Poisson process. These queueing systems are called *symmetric queueing systems*, because they are symmetric in time, as the arrival process and the departure process are of same type. If we make a film of the time development, we cannot decide whether the film is run forward or backward (cf. reversibility) (Kelly, 1979 [59]).

The classical queueing models play a key role in the queueing theory, because other systems will often converge to them when the number of servers increases (Palm's theorem 6.1 in Sec. 6.4).

Systems that deviate most from the classical models are the systems with a single server. However, these systems are also the simplest to deal with.

In waiting time systems we also distinguish between call averages and time averages. The *virtual waiting time* is the waiting time, a customer experiences if the customer arrives at a random point of time (time average). The actual waiting time is the waiting time, the real customers experiences (call average). If the arrival process is a Poisson process, then the two averages are identical.

13.3 Pollaczek-Khintchine's formula for $M/G/1$

We have earlier derived the mean waiting time for *M/M/1* (Sec. 12.2.4) and later we consider *M/D/1* (Sec. 13.5). In general the mean waiting time for *M/G/1* is given by:

Theorem 13.1 *Pollaczek-Khintchine's formula (1930–32):*

$$W = \frac{V}{1-A}\,, \tag{13.1}$$

$$W = \frac{A \cdot s}{2(1-A)} \cdot \varepsilon\,, \tag{13.2}$$

as

$$V = A\,\frac{s}{2}\,\varepsilon = \frac{\lambda}{2}\,m_2\,. \tag{13.3}$$

W is the mean waiting time for all customers, s is the mean service time, A is the offered traffic, and ε is the form factor of the holding time distribution (3.10).

The more regular the service process is, the smaller the mean waiting time will become. The corresponding results for the arrival process is studied in Sec. 13.6. In real telephone traffic the form factor will often be $4-6$, in data traffic $10-100$.

Formula(13.2) is one of the most important results in queueing theory, and we will study it carefully.

13.3.1 Derivation of Pollaczek-Khintchine's formula

We consider the queueing system *M/G/1* and we wish to find the mean waiting time for an arbitrary customer. It is independent of the queueing discipline, and therefore we may in the following assume *FCFS*. Due to the Poisson arrival process (*PASTA–property*) the actual waiting time of a customers is equal to the virtual waiting time.

The mean waiting time W for an arbitrary customer can be split up into two parts:

1. The time it takes for a customer under service to be completed. When the new customer we consider arrives at a random point of time, the residual mean service time given by (3.25):
$$m_{1,r} = \frac{s}{2} \cdot \varepsilon\,,$$
where s and ε have the same meaning as in (13.2). When the arrival process is a Poisson process, the probability of finding a customer being served is equal to A because for a single server system we always have $p_0 = 1-A$ (offered traffic = carried traffic).

The contribution to the mean waiting time from a customer under service therefore becomes:

$$V = (1-A)\cdot 0 + A \cdot \frac{s}{2}\cdot \varepsilon$$

$$= \frac{\lambda}{2}\cdot m_2 \,. \tag{13.4}$$

2. The waiting time due to waiting customers in the queue (*FCFS*). On the average the queue length is L. By Little's theorem we have

$$L = \lambda \cdot W \,,$$

where L is the average number of customers in the queue at an arbitrary point of time, λ is the arrival intensity, and W is the mean waiting time which we look for. For every customer in the queue we shall on an average wait s time units. The mean waiting time due to the customers in the queue therefore becomes:

$$L \cdot s = \lambda \cdot W \cdot s = A \cdot W \,. \tag{13.5}$$

We thus have the total waiting time (13.4) & (13.5):

$$W = V + AW \,,$$

$$W = \frac{V}{1-A}$$

$$= \frac{A \cdot s}{2(1-A)} \cdot \varepsilon \,,$$

which is Pollaczek-Khintchine's formula (13.2). W is the mean waiting time for all customers, whereas the mean waiting time for delayed customers w becomes ($A = D =$ the probability of delay) (3.20):

$$w = \frac{W}{D} = \frac{s}{2(1-A)} \cdot \varepsilon \,. \tag{13.6}$$

The above-mentioned derivation is correct since the time average is equal to the call average when the arrival process is a Poisson process (*PASTA–property*). It is interesting, because it shows how ε enters into the formula.

13.3.2 Busy period for *M/G/1*

A busy period of a queueing system is the time interval from the instant all servers become busy until a server becomes idle again. For *M/G/1* it is easy to calculate the mean value of a busy period.

At the instant the queueing system becomes empty, it has lost its memory due to the Poisson arrival process. These instants are regeneration points (equilibrium points), and next event occurs according to a Poisson process with intensity λ.

We need only consider a cycle from the instant the server changes state from idle to busy till the next time it changes state from idle to busy. This cycle includes a busy period of duration T_1 and an idle period of duration T_0. Fig. 13.1 shows an example with constant service time. The proportion of time the system is busy then becomes:

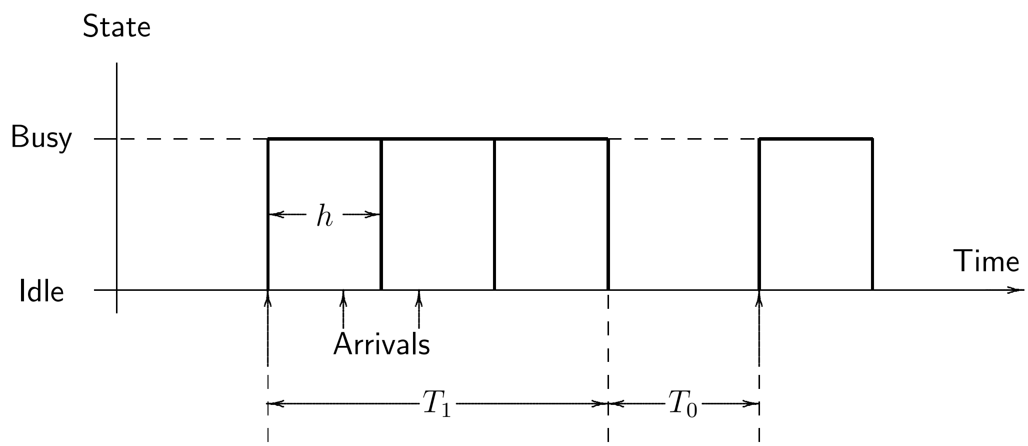


Figure 13.1: *Example of a sequence of events for the system M/D/1 with busy period T_1 and idle period T_0.*

$$\frac{m_{T_1}}{m_{T_0+T_1}} = \frac{m_{T_1}}{m_{T_0}+m_{T_1}} = A = \lambda \cdot s\,.$$

From $m_{T_0} = 1/\lambda$, we get:

$$m_{T_1} = \frac{s}{1-A}\,. \tag{13.7}$$

During a busy period at least one customer is served.

13.3.3 Waiting time for *M/G/1*

If we only consider customers, which are delayed, we are able to find the moments of the waiting time distribution for the classical queueing disciplines (Abate & Whitt, 1997 [1]).

FCFS : Denoting the *i'th* moment of the service time distribution by m_i, we can find the *k'th* moment of the waiting time distribution by the following recursion formula, where the mean service time is chosen as time unit ($m_1 = s = 1$):

$$m_{k,F} = \frac{A}{1-A}\sum_{j=1}^{k}\binom{k}{j}\cdot\frac{m_{j+1}}{j+1}\cdot m_{k-j,F}\,, \quad v_0 = 1\,. \tag{13.8}$$

LCFS : From the above moments $m_{k,F}$ of the *FCFS*–waiting time distribution we can find the moments $m_{k,L}$ of the *LCFS*–waiting time distribution. The three first moments become:

$$m_{1,L} = m_{1,F}\,, \qquad m_{2,L} = \frac{m_{2,F}}{1-A}\,, \qquad m_{3,L} = \frac{m_{3,F} + 3 \cdot m_{1,F} \cdot m_{2,F}}{(1-A)^2}\,. \tag{13.9}$$

13.3.4 Limited queue length: *M/G/1/k*

In real systems the queue length, for example the size of a buffer, will always be finite. Customers arriving when the buffer is full are blocked. For example in the Internet, this strategy is applied in routers and is called the *drop tail* strategy. There exists a simple relation between the state probabilities $p(i)$ $(i = 0, 1, 2, \ldots)$ of the infinite system *M/G/1* and the state probabilities $p_k(i)$, $(i = 0, 1, 2, \ldots, k)$ of *M/G/1/k*, where the total number of positions for customers is k, including the customer being served (Keilson, 1966 [58]):

$$p_k(i) = \frac{p(i)}{(1 - A \cdot Q_k)}\,, \qquad i = 0, 1, \ldots, k-1\,, \tag{13.10}$$

$$p_k(k) = \frac{(1 - A) \cdot Q_k}{(1 - A \cdot Q_k)}\,, \tag{13.11}$$

where $A < 1$ is the offered traffic, and:

$$Q_k = \sum_{j=k}^{\infty} p(j)\,. \tag{13.12}$$

There exists algorithms for calculating $p(i)$ for arbitrary holding time distributions (*M/G/1*) based on imbedded Markov chain analysis (Kendall, 1953 [61]), where the same approach is used for (*GI/M/1*).

We notice that the above is only valid for $A < 1$, but for a finite buffer we also obtain statistical equilibrium for $A > 1$. In this case we cannot use the approach described in this section. For *M/M/1,k* we can use the finite state transition diagram, and for *M/D/1,k* we describe a simple approach in Sec. 13.5.8, which is applicable for general holding time dsitributions.

13.4 Priority queueing systems: *M/G/1*

The time period a customer is waiting usually means an inconvenience or expense to the customer. By different strategies for organising the queue, the waiting times can be distributed among the customers according to our preferences.

13.4.1 Combination of several classes of customers

The customers are divided into N classes (traffic streams). Customers of class i are assumed to arrive according to a Poisson process with intensity λ_i [customers per time unit] and the mean service time is s_i [time units]. The second moment of the service time distribution is denoted m_{2i}, and the offered traffic is $A_i = \lambda_i \cdot s_i$.

In stead of considering the individual arrival processes, we may consider the total arrival process, which also is a Poisson arrival process with intensity:

$$\lambda = \sum_{i=1}^{N} \lambda_i \,. \tag{13.13}$$

The resulting service time distribution then becomes a weighted sum of service time distributions of the individual classes (Sec. 3.2: combination in parallel). The total mean service time becomes:

$$s = \sum_{i=1}^{N} \frac{\lambda_i}{\lambda} \cdot s_i \,, \tag{13.14}$$

and the total second moment is:

$$m_2 = \sum_{i=1}^{N} \frac{\lambda_i}{\lambda} \cdot m_{2i} \,. \tag{13.15}$$

The total offered traffic is:

$$A = \sum_{i=1}^{N} A_i = \sum_{i=1}^{N} \lambda_i \cdot s_i = \lambda\, s \,. \tag{13.16}$$

The remaining mean service time at a random point of time becomes (13.4):

$$\begin{aligned} V &= \frac{1}{2} \cdot \lambda \cdot m_2 \qquad (13.17) \\ &= \frac{1}{2} \cdot A \cdot \frac{1}{s} \cdot m_2 \\ &= \frac{1}{2} \cdot A \cdot \left\{ \sum_{i=1}^{N} \frac{\lambda_i}{\lambda} \cdot s_i \right\}^{-1} \cdot \left\{ \sum_{i=1}^{N} \frac{\lambda_i}{\lambda} \cdot m_{2i} \right\} \\ &= \frac{1}{2} \cdot A \cdot \left\{ \sum_{i=1}^{N} \frac{A_i}{\lambda} \right\}^{-1} \cdot \left\{ \sum_{i=1}^{N} \frac{\lambda_i}{\lambda} \cdot m_{2i} \right\} \end{aligned}$$

$$V = \sum_{i=1}^{N} \frac{\lambda_i}{2} \cdot m_{2i} \tag{13.18}$$

$$= \sum_{i=1}^{N} V_i\,. \tag{13.19}$$

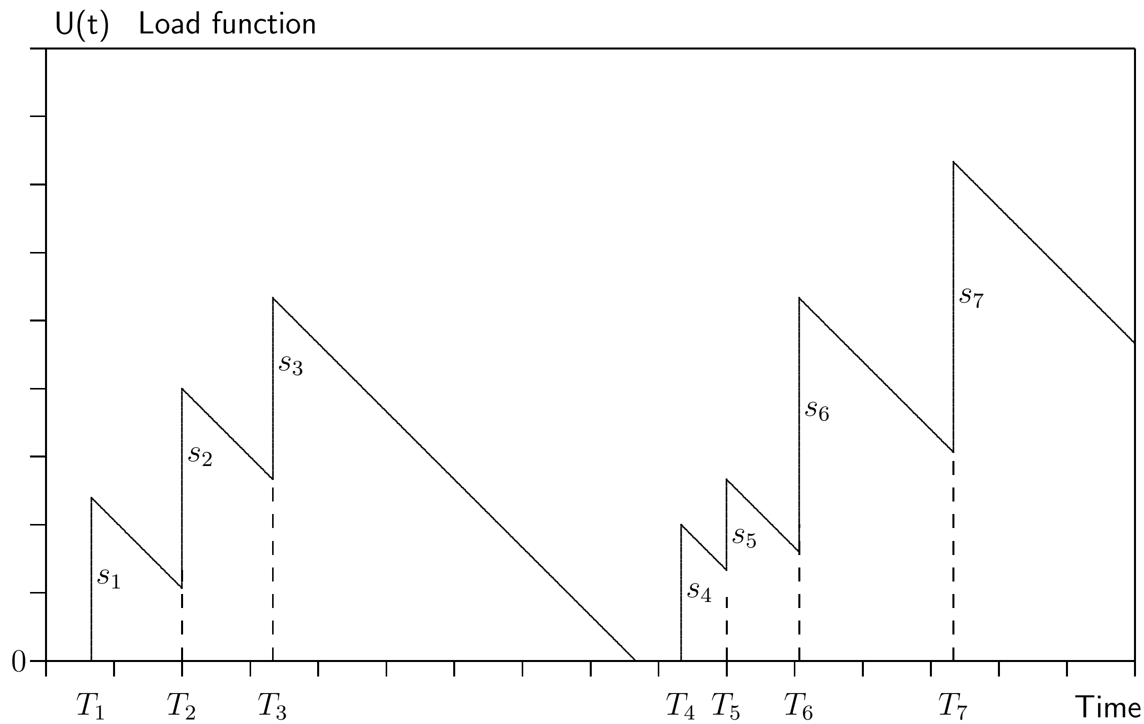


Figure 13.2: *The load function $U(t)$ for the queueing system GI/G/1. If we denote the inter-arrival time $T_{i+1} - T_i$ by a_i, then we have $U_{i+1} = max\{0, U_i + s_i - a_i\}$, where U_i is the value of the load function at time T_i.*

13.4.2 Work conserving queueing disciplines

In the following we shall assume that the service time of a customer is independent of the queueing discipline. The capacity of the server is thus constant and independent of for example the length of the queue. The queueing discipline is said to be *work conserving*. This will not always be the case in practise. If the server is a human being, the service rate will often increase with the length of the queue, and after some time the server may become exhausted and the service rate decreases.

We introduce two functions, which are widely applied in the queueing theory.

Load function $U(t)$ denotes the time, it will require to serve the customers, which has arrived to the system at time t (Fig. 13.2). At a time of arrival $U(t)$ increases with a jump equal to the service time of the arriving customer, and between arrivals $U(t)$ decreases linearly with the slope −1 until 0, where it stays until next arrival time. The mean value of the load function is denoted by $U = E\{U(t)\}$. In a *GI/G/1* queueing system $U(t)$ will be independent of the queueing discipline, if it is work conserving.

The virtual waiting time $W(t)$ denotes the waiting time of a customer, if he arrives at time instant t. The virtual waiting time $W(t)$ depends on the queue organisation. The mean value is denoted by $W = E\{W(t)\}$. If the queue discipline is *FCFS*, then $U(t) = W(t)$. When we consider Poisson arrival processes, the virtual waiting time will be equal to the actual waiting time (*PASTA* property: time average = call average).

We now consider the load function at a random point of time t. It consists of a contribution V from the remaining service time of a customer being served, if any, and a contribution from customers waiting in the queue. The mean value $U = E\{U(t)\}$ becomes:

$$U = V + \sum_{i=1}^{N} L_i \cdot s_i \,.$$

L_i is the queue length for customers of type i. By applying Little's law we get:

$$\begin{aligned} U &= V + \sum_{i=1}^{N} \lambda_i \cdot W_i \cdot s_i \\ &= V + \sum_{i=1}^{N} A_i \cdot W_i \,. \end{aligned} \tag{13.20}$$

As mentioned above, U is the independent of the queueing discipline (the system is assumed to be work conserving), and V is given by (13.17) for *non-preemptive* queueing disciplines. U is obtained by assuming *FCFS*, as we then have $W_i = U$:

$$U = V + \sum_{i=1}^{N} A_i \cdot U = V + A \cdot U \,,$$

$$U = \frac{V}{1-A} \,, \tag{13.21}$$

$$U - V = \frac{A \cdot V}{1-A} \,. \tag{13.22}$$

Under these general assumptions we get by inserting (13.22) into (13.20) Kleinrock's conservation law (1964 [64]):

Theorem 13.2 *Kleinrock's conservation law:*

$$\sum_{i=1}^{N} A_i \cdot W_i = \frac{A \cdot V}{1-A} = constant. \tag{13.23}$$

The average waiting time for all classes weighted by the traffic (load) of the mentioned class, is independent of the queue discipline.

Notice that the above is only valid for non-preemptive queueing disciplines. We may thus give a small proportion of the traffic a very low mean waiting time, without increasing the average waiting time of the remaining customers very much. By various strategies we may allocate waiting times to individual customers according to our preferences.

13.4.3 Non-preemptive queueing discipline

In the following we look at the *M/G/1* priority queueing systems, where customers are divided into N priority classes so that a customer with the priority p has higher priority than customers with priority $p+1$. In a non-preemptive system a service in progress is not interrupted.

The customers in class p are assumed to have the mean service time s_p and the arrival intensity λ_p. In Sec. 13.4.1 we derived parameters for the total process.

The total average waiting time W_p of a class p customers can be derived directly by considering the following three contributions:

a) The residual service time V for the customer under service.

b) The waiting time, due to the customers in the queue with priority p or higher, which are already in the queues (Little's theorem):

$$\sum_{i=1}^{p} s_i \cdot (\lambda_i W_i) \ .$$

c) The waiting time due to customers with higher priority, which overtake the customer we consider while this is waiting:

$$\sum_{i=1}^{p-1} s_i \cdot \lambda_i \cdot W_p \, .$$

In total we get:

$$W_p = V + \sum_{i=1}^{p} s_i \cdot \lambda_i \cdot W_i + \sum_{i=1}^{p-1} s_i \cdot \lambda_i \cdot W_p \, . \tag{13.24}$$

For customers of class 1, which have highest priority we get under the assumption of *FCFS*:

$$\begin{aligned} W_1 &= V + L_1 \cdot s_1 \qquad (13.25) \\ &= V + A_1 \cdot W_1 \,, \\ W_1 &= \frac{V}{1 - A_1} \,. \qquad (13.26) \end{aligned}$$

V is the residual service time for the customer under service when the customer we consider arrives (13.18):

$$V = \sum_{i=1}^{N} \frac{\lambda_i}{2} \cdot m_{2i} \,, \qquad (13.27)$$

where m_{2i} is the second moment of the service time distribution of the i'th class.

For class 2 customers we find:

$$W_2 = V + L_1 \cdot s_1 + L_2 \cdot s_2 + W_2 \cdot (s_1 \lambda_1).$$

Inserting W_1 (13.25), we get:

$$\begin{aligned} W_2 &= W_1 + A_2 \cdot W_2 + A_1 \cdot W_2 \,, \\ W_2 &= \frac{W_1}{1 - A_1 - A_2} \,, \qquad (13.28) \\ W_2 &= \frac{V}{\{1 - A_1\}\,\{1 - (A_1 + A_2)\}} \,. \qquad (13.29) \end{aligned}$$

In general we find (Cobham, 1954 [14]):

$$W_p = \frac{V}{\left\{1 - A'_{p-1}\right\}\left\{1 - A'_p\right\}} \,, \qquad (13.30)$$

where:

$$A'_p = \sum_{i=0}^{p} A_i \,, \qquad A_0 = 0 \,. \qquad (13.31)$$

The structure in formula (13.30) can be directly interpreted. No matter which class all customers wait until the service in progress is completed $\{V\}$.

Furthermore, the waiting time is due to customers who have already arrived and have at least the same priority $\left\{A'_p\right\}$, and customers with higher priority arriving during the waiting time $\left\{A'_{p-1}\right\}$.

Example 13.4.1: SPC-system
We consider a computer which serves two types of customers. The first type has the constant service time of 0.1 second, and the arrival intensity is 1 customer/second. The other type has the exponentially distributed service time with the mean value of 1.6 second and the arrival intensity is 0.5 customer/second.

The load from the two types customers is then $A_1 = 0.1$ erlang, respectively $A_2 = 0.8$ erlang. From (13.27) we find:

$$V = \frac{1}{2} \cdot (0.1)^2 + \frac{0.5}{2} \cdot 2 \cdot (1.6)^2 = 1.2850 \; s \, .$$

Without any priority the mean waiting time becomes by using Pollaczek-Khintchine's formula (13.2):

$$W = \frac{1.2850}{1-(0.8+0.1)} = 12.85 \; s \, .$$

By non-preemptive priority we find:

Type 1 highest priority:

$$\begin{aligned} W_1 &= \frac{1.285}{1-0.1} = 1.43 \; s \, , \\ W_2 &= \frac{W_1}{1-(A_1+A_2)} = 14.28 \; s \, . \end{aligned}$$

Type 2 highest priority:

$$\begin{aligned} W_2 &= 6.43 \; s \, , \\ W_1 &= 64.25 \; s \, . \end{aligned}$$

This shows that we can upgrade type 1 almost without influencing type 2. However the inverse is not the case. The constant in the *Conservation law* (13.23) becomes the same without priority as with non–preemptive priority:

$$0.9 \cdot 12.85 = 0.1 \cdot 1.43 + 0.8 \cdot 14.28 = 0.8 \cdot 6.43 + 0.1 \cdot 64.25 = 11.57 \, .$$

□

13.4.4 *SJF*-queueing discipline

By the SJF-queueing discipline the shorter the service time of a customer is, the higher is the priority. By introducing an infinite number of priority classes, we obtain from the formula (13.30) that a customer with the service time t has the mean waiting time W_t (Phipps 1956):

$$W_t = \frac{V}{(1-A_t)^2} \, , \tag{13.32}$$

where A_t is load from the customers with service time less than or equal to t.

The *SJF* discipline results in the lowest possible total waiting time.

If these different priority classes have different costs per time unit when they wait, so that class j customers have the mean service time s_j and pay c_j per time unit when they wait, then the optimal strategy (minimum cost) is to assign priorities $1, 2, \ldots$ according to increasing ratio s_j/c_j.

Example 13.4.2: $M/M/1$ with SJF queue discipline
We consider the case with exponentially distributed holding times with the mean value $1/\mu$ that is chosen as time unit (*M/M/1*). Even though there are few very long service times, then they contribute significantly to the total traffic (Fig. 3.2).

The contribution to the total traffic A from the customers with service time $\leq t$ is $\{$(3.22) multiplied by $A = \lambda \cdot \mu\}$:

$$
\begin{aligned}
A_t &= \int_0^t x \cdot \lambda \cdot f(x)\,\mathrm{d}x \\
&= \int_0^t x \cdot \lambda \cdot \left(\mu \cdot \mathrm{e}^{-\mu x}\right)\,\mathrm{d}x \\
&= A\left\{1 - \mathrm{e}^{-\mu t}(\mu t + 1)\right\} .
\end{aligned}
$$

Inserting this in (13.32) we find W_t as illustrated in Fig. 13.3, where the *FCFS*–strategy (same mean waiting time as *LCFS* and *SIRO*) is shown for comparison as function of the actual holding time. The round-robin strategy gives a waiting time which is proportional to the service time. The mean waiting time for all customers with *SJF* is less than with *FCFS*, but this is not evident from the figure. The mean waiting time for *SJF* becomes:

$$
\begin{aligned}
W_{SJF} &= \int_0^\infty W_t\, f(t)\,\mathrm{d}t \\
&= \int_0^\infty \frac{V}{(1-A_t)^2} \cdot f(t)\,\mathrm{d}t \\
&= \int_0^\infty \frac{A \cdot \mathrm{e}^{-\mu t}\,\mathrm{d}t}{\{1 - A\,(1-\mathrm{e}^{-\mu t}\,(\mu t + 1))\}^2}
\end{aligned}
$$

which it is not elementary to calculate. □

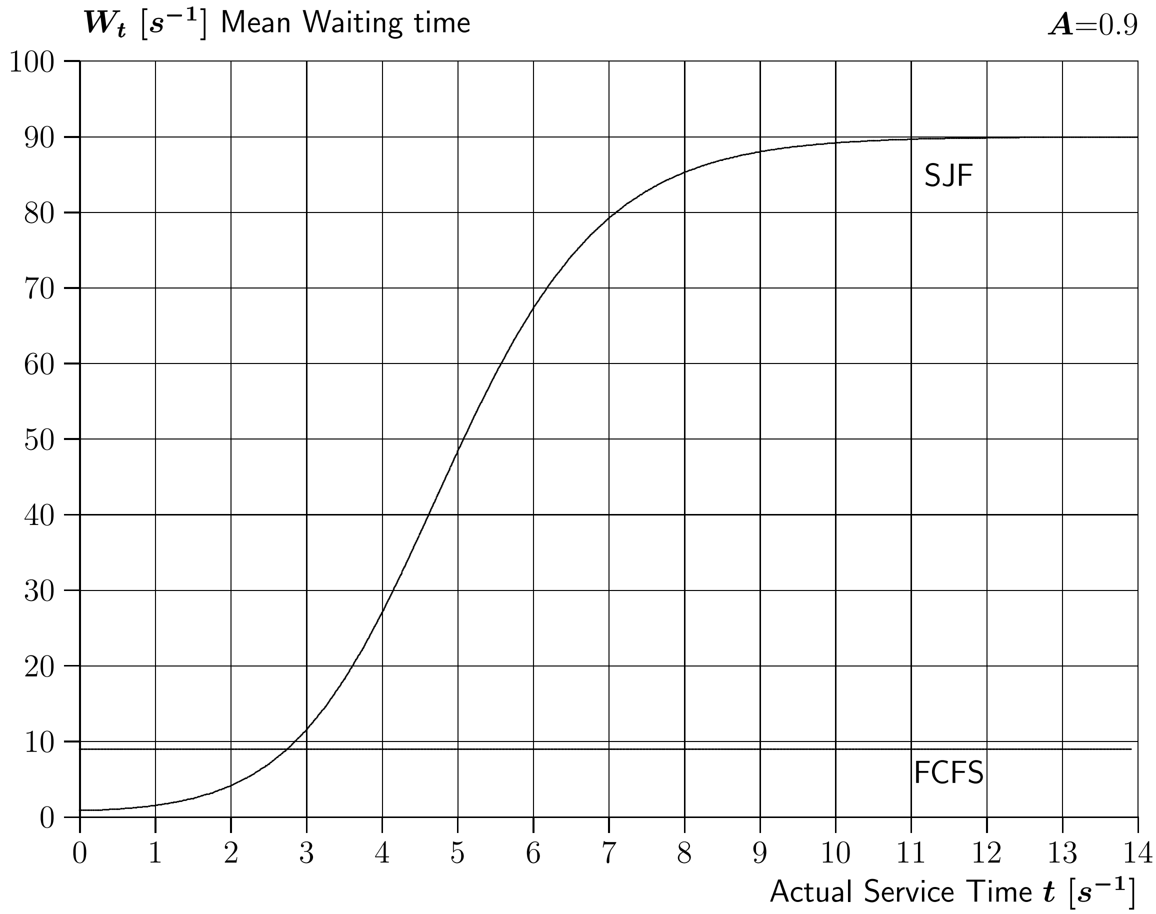


Figure 13.3: *The mean waiting time W_t is a function of the actual service time in a M/M/1–system for SJF and FCFS disciplines, respectively. The offered traffic is 0.9 erlang and the mean service time is chosen as time unit. Notice that for SJF the minimum average waiting time is 0.9 time units, because an eventual job being served must first be finished. The maximum mean waiting time is 90 time units. In comparison with FCFS by using SJF 93.6 % of the jobs get shorter mean waiting time. This corresponds to jobs with a service time less than 2.747 mean service times (time units). The offered traffic may be greater than one erlang, but then only the shorter jobs get a finite waiting time.*

13.4.5 $M/M/n$ with non-preemptive priority

We may also generalise Erlang's classical waiting time system $M/M/n$ with non–preemptive queueing disciplines, when all classes of customers have the same exponentially distributed service time distribution with mean value $s = \mu^{-1}$. Denoting the arrival intensity for class i by λ_i, we have the mean waiting time W_p for class p:

$$\begin{aligned} W_p &= V + \sum_{i=1}^{p} \frac{s}{n} \cdot L_i + W_p \sum_{i=1}^{p-1} \frac{s}{n}\, \lambda_i \,, \\ W_p &= E_{2,n}(A) \cdot \frac{s}{n} + \sum_{i=1}^{p} \frac{s\,\lambda_i}{n} \cdot W_i + W_p \sum_{i=1}^{p-1} \frac{s}{n}\, \lambda_i \,. \end{aligned}$$

A is the total offered traffic for all classes. The probability $E_{2,n}(A)$ for waiting time is given by Erlang's C-formula, and customers are terminated with the mean inter-departure time s/n when all servers are busy. For highest priority class $p = 1$ we find:

$$\begin{aligned} W_1 &= E_{2,n}(A)\, \frac{s}{n} + \frac{1}{n}\, A_1 W_1 \,, \\ W_1 &= \frac{s \cdot E_{2,n}(A)}{n - A_1} \,. \end{aligned}$$

For $p = 2$ we find in a similar way:

$$\begin{aligned} W_2 &= E_{2,n}(A)\, \frac{s}{n} + \frac{1}{n}\, A_1 W_1 + \frac{1}{n}\, A_2\, W_2 + W_2 \left\{ \frac{s}{n} \cdot \lambda_1 \right\} \\ &= W_1 + \frac{1}{n}\, A_2\, W_2 + \frac{1}{n} \cdot A_1\, W_2 \,, \\ W_2 &= \frac{n\, s\, E_{2,n}(A)}{\{n - A_1\}\, \{n - (A_1 + A_2)\}} \,. \end{aligned}$$

In general we find (Cobham, 1954 [14]):

$$W_p = \frac{n\, s\, E_{2,n}(A)}{\left\{n - A'_{p-1}\right\} \left\{n - A'_p\right\}} \,. \tag{13.33}$$

The case of preemptive resume is more difficult to deal with because customers with higher priority which arrive during a service time not necessarily interrupt a customer being served, because there are more servers. The mean waiting time can be obtained by first considering class one alone, then consider class one and two together, which implies the waiting time for class two, etc. This will only be correct when the service time is exponentially distributed.

13.4.6 Preemptive-resume queueing discipline

We now assume that an ongoing service is interrupted by the arrival of a customer with a higher priority. Later the service continues from where it was interrupted. This situation is typical for computer systems. For a customer with the priority p there is no customer with lower priority. The mean waiting time W_p for a customer in class p consists of two contributions.

a) Waiting time due to customers with higher or same priority, who are already in the queueing system. This is the waiting time experienced by a customer in a system without priority where only the first p classes exists:

$$\frac{V_p}{1-A'_p}\,, \quad \textit{where} \quad V_p = \sum_{i=1}^{p} \frac{\lambda_i}{2} \cdot m_{2,i}\,, \tag{13.34}$$

is the expected remaining service time due to customers with a higher or the same priority and A'_p is given by (13.31).

b) The waiting time due to the customers with higher priority who arrive during the waiting time or service time and overtake the customer considered:

$$(W_p + s_p) \sum_{i=1}^{p-1} s_i \cdot \lambda_i = (W_p + s_p) \cdot A'_{p-1}\,.$$

We thus get:

$$W_p = \frac{V_p}{1-A'_p} + (W_p + s_p) \cdot A'_{p-1}\,. \tag{13.35}$$

This can be rewritten as follows:

$$W_p(1-A'_{p-1}) = \frac{V_p}{\left\{1-A'_p\right\}} + s_p \cdot A'_{p-1}\,, \tag{13.36}$$

resulting in:

$$W_p = \frac{V_p}{\left(1-A'_{p-1}\right)\left(1-A'_p\right)} + \frac{A'_{p-1}}{1-A'_{p-1}} \cdot s_p\,. \tag{13.37}$$

In the same way as in Sec. 13.4.4 we may write the formula for the average waiting time for the *SJF*–queueing discipline with preemptive resume. The total response time becomes:

$$T_p = W_p + s_p\,. \tag{13.38}$$

Example 13.4.3: SPC–system (cf. example 13.4.1)
We now assume the computer system in Example 13.4.1 is working with the discipline preemptive-resume and find:

Type 1 highest priority:

$$
\begin{aligned}
W_1 &= \frac{\frac{1}{2}(0.1)^2}{1-0.1} + 0 = 0.0056\ s\,, \\
W_2 &= \frac{1.2850}{(1-0.1)(1-0.9)} + \frac{0.1}{1-0.1} \cdot 1.6 = 14.46\ s\,.
\end{aligned}
$$

Type 2 highest priority:

$$
\begin{aligned}
W_2 &= \frac{\frac{1}{2} \cdot 0.5 \cdot 2 \cdot (1.6)^2}{1-0.8} + 0 = 6.40\ s\,, \\
W_1 &= \frac{1.2850}{(1-0.8)(1-0.9)} + \frac{0.8}{1-0.8} \cdot 0.1 = 64.65\ s\,.
\end{aligned}
$$

This shows that by upgrading type 1 to the highest priority, we can give these customers a very short waiting time, without disturbing type 2 customers, but the inverse is not the case.

The conservation law is only valid for preemptive queueing systems if the preempted service times are exponentially distributed. In the general case a job may be preempted several times and therefore the remaining service time will not be given by V. □

13.5 Queueing systems with constant holding times

In this section we focus upon the queueing system *M/D/n, FCFS*. Systems with constant service times have the particular property that the customers leave the servers in the same order in which they are accepted for service.

13.5.1 Historical remarks on $M/D/n$

Queueing systems with Poisson arrival process and constant service times were the first systems to be analysed. Intuitively, one would think that it is easier to deal with constant service times than with exponentially distributed service times, but this is definitely not the case. The exponential distribution is easy to deal with due to its lack of memory: the remaining life-time has the same distribution as the total life-time (Sec. 4.1), and therefore we can forget about the epoch (point of time) when the service time starts. Constant holding times require that we remember the exact starting time.

Erlang was the first to analyse *M/D/n, FCFS* (Brockmeyer & al., 1948 [11]):

Erlang:	1909	$n = 1$	errors for $n > 1$,
Erlang:	1917	$n = 1, 2, 3$	without proof,
Erlang:	1920	n arbitrary	explicit solutions for $n = 1, 2, 3$.

Erlang derived the waiting time distribution, but did not consider the state probabilities. Fry (1928 [30]) also dealt with *M/D/1* and derived the state probabilities (*Fry's equations of state*) by using Erlang's principle of statistical equilibrium, whereas Erlang himself applied more theoretical methods.

Crommelin (1932 [20], 1934 [21]), a British telephone engineer, presented a general solution to *M/D/n*. He generalised Fry's equations of state to an arbitrary n and derived the waiting time distribution, now named *Crommelin's distribution*.

Pollaczek (1930-34) presented a very general time-dependent solution for arbitrary service time distributions. Under the assumption of statistical equilibrium he was able to obtain explicit solutions for exponentially distributed and constant service times. Also Khintchine (1932 [62]) dealt with *M/D/n* and derived the waiting time distribution.

13.5.2 State probabilities of *M/D/1*

Under the assumption of statistical equilibrium we now derive the state probabilities for *M/D/1* in a simple way. The arrival intensity is denoted by λ and the constant holding time by h. As we consider a pure waiting time system with a single server we have:

$$\textit{Offered traffic} = \textit{Carried traffic} = \lambda \cdot h < 1\,, \tag{13.39}$$

i.e.

$$A = Y = \lambda \cdot h = 1 - p(0)\,,$$

as in every state except zero the carried traffic is equal to one erlang.

We consider two epochs (points of time) t and $t+h$ at a distance of h. Every customer being served at epoch t (at most one) has left the server at epoch $t+h$. Customers arriving during the interval $(t,\ t+h)$ are still in the queueing system at epoch $t+h$ (waiting or being served).

The arrival process is a Poisson process. Hence we have a Poisson distributed number of arrivals in the time interval $(t,\ t+h)$:

$$p(j,h) = p\{j \text{ calls in } h\} = \frac{(\lambda h)^j}{j!} \cdot \mathrm{e}^{-\lambda h}\,, \quad j = 0, 1, 2 \ldots\,. \tag{13.40}$$

The probability of being in a given state at epoch $t+h$ is obtained from the state at epoch t by taking account of all arrivals and departures during $(t,\ t+h)$. By looking at these epochs we obtain a Markov Chain embedded in the original traffic process (Fig. 13.4).

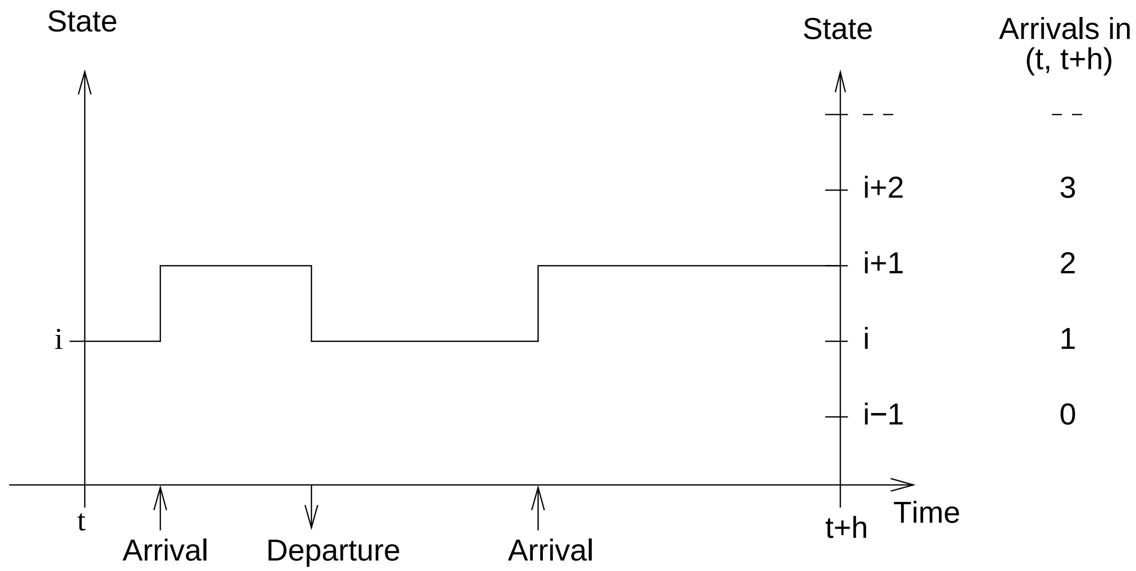


Figure 13.4: *Illustration of Fry's equations of state for the queueing system M/D/1.*

We obtain *Fry's equations of state* for $n = 1$ (Fry, 1928 [30]):

$$p_{t+h}(i) = \{p_t(0) + p_t(1)\}\, p(i,h) + \sum_{j=2}^{i+1} p_t(j) \cdot p(i-j+1, h) \tag{13.41}$$

Above we found:

$$p(0) = 1 - A$$

and under the assumption of statistical equilibrium $p_t(i) = p_{t+h}(i)$, we successively find:

$$\begin{aligned} p(1) &= (1-A) \cdot \left\{ \mathrm{e}^{A} - 1 \right\} , \\ p(2) &= (1-A) \cdot \left\{ -\mathrm{e}^{A} \cdot (1+A) + \mathrm{e}^{2A} \right\} , \end{aligned}$$

and in general:

$$p(i) = (1-A) \cdot \sum_{j=1}^{i} (-1)^{i-j} \cdot \mathrm{e}^{jA} \cdot \left\{ \frac{(jA)^{i-j}}{(i-j)!} + \frac{(jA)^{i-j-1}}{(i-j-1)!} \right\}, \qquad i = 2, 3, \ldots \tag{13.42}$$

The last term corresponding to $j = i$ always equals e^{iA}, as $(-1)! \equiv \infty$. In principle $p(0)$ can also be obtained by requiring that all state probabilities must add to one.

13.5.3 Mean waiting times and busy period of *M/D/1*

For a Poison arrival process the probability of delay D is equal to the probability of not being in state zero (*PASTA property*):

$$D = A = 1 - p(0)\,. \tag{13.43}$$

W denotes the mean waiting time for all customers and w denotes the mean waiting time for customers experiencing a positive waiting time. We have for any queueing system (3.20):

$$w = \frac{W}{D}\,. \tag{13.44}$$

W and w are easily obtained by using Pollaczek-Khintchine's formula (13.2):

$$W = \frac{A \cdot h}{2(1-A)}\,, \tag{13.45}$$

$$w = \frac{h}{2(1-A)}\,. \tag{13.46}$$

The mean value of a busy period was obtained for *M/G/1* in (13.7) and illustrated for constant service times in Fig. 13.1:

$$m_{T_1} = \frac{h}{1-A}\,. \tag{13.47}$$

The mean waiting time for delayed customers are thus half the busy period. It looks like customers arrive at random during the busy period, but we know that are no customers arrive during the last service time of a busy period.

The distribution of the number of customer arriving during a busy period can be shown to be given by a *Borél distribution*:

$$B(i) = \frac{(i\,A)^{i-1}}{i!}\,\mathrm{e}^{-i\,A}, \qquad i = 1, 2, \ldots \tag{13.48}$$

13.5.4 Waiting time distribution: *M/D/1, FCFS*

This can be shown to be:

$$p\{W \le t\} = 1 - (1-\lambda) \cdot \sum_{j=1}^{\infty} \frac{\{\lambda(j-\tau)\}^{T+j}}{(T+j)!} \cdot \mathrm{e}^{-\lambda(j-\tau)}\,, \tag{13.49}$$

where $h = 1$ is chosen as time unit, $t = T + \tau$, T is an integer, and $0 \le \tau < 1$.

The graph of the waiting time distribution has an irregularity every time the waiting time exceeds an integral multiple of the constant holding time. An example is shown in Fig. 13.5.

Formula (13.49) is not suitable for numerical evaluation. It can be shown (Iversen, 1982 [39]) that the waiting time can be written in a closed form, as given by Erlang in 1909:

$$p\{W \le t\} = (1-\lambda) \cdot \sum_{j=0}^{T} \frac{\{\lambda(j-t)\}^{j}}{j!} \cdot \mathrm{e}^{-\lambda(j-t)}\,, \tag{13.50}$$

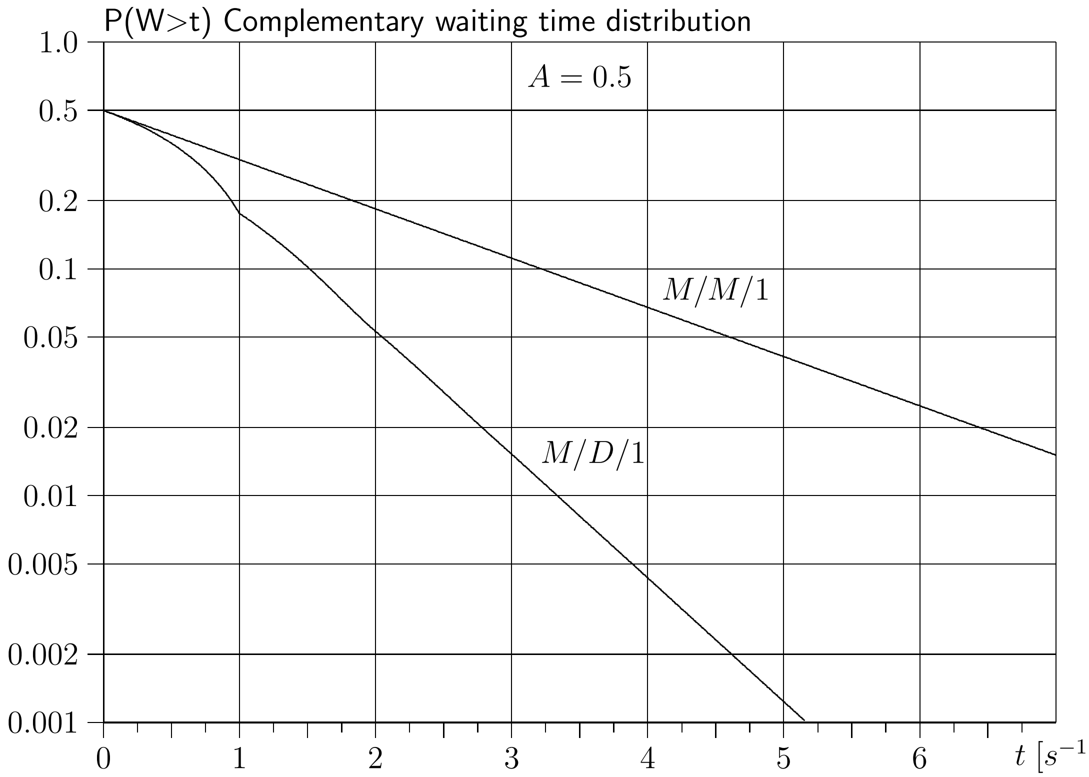


Figure 13.5: *The complementary waiting time distribution for all customers in the queueing system M/M/1 and M/D/1 for ordered queue (FCFS). Time unit = mean service time. We notice that the mean waiting time for M/D/1 is only half of that for M/M/1.*

which is fit for numerical evaluation for small waiting times.

For larger waiting times we are usually only interested in integral values of t. It can be shown (Iversen, 1982 [39]) that for an integral value of t we have:

$$p\{W \leq t\} = p(0) + p(1) + \cdots + p(t)\,. \tag{13.51}$$

The state probabilities $p(i)$ are calculated most accurately by using a recursive formula based on Fry's equations of state (13.42):

$$p(i+1) = \frac{1}{p(0,h)} \left\{ p(i) - \{p(0)+p(1)\} \cdot p(i,h) - \sum_{j=2}^{i} p(j) \cdot p(i-j+1,h) \right\}. \tag{13.52}$$

For non-integral waiting-times we are able to express the waiting time distribution in terms of integral waiting times.

If we let $h = 1$, then (13.50) may be a binomial expansion be written in powers of τ, where

$$t = T + \tau, \quad T \text{ integer}, \quad 0 \leq \tau < 1\,.$$

We find:

$$p\{W \leq T+\tau\} = \mathrm{e}^{\lambda\tau} \sum_{j=0}^{T} \frac{(-\lambda\tau)^j}{j!} \cdot p\{W \leq T-j\}\,, \tag{13.53}$$

where $p\{W \leq T-j\}$ is given by (13.51).

The numerical evaluation is very accurate when using (13.51), (13.52) and (13.53).

13.5.5 State probabilities: $M/D/n$

When setting up *Fry's equations of state* (13.41) we obtain more combinations:

$$p_{t+h}(i) = \left\{ \sum_{j=0}^{n} p_t(j) \right\} p(i,h) + \sum_{j=n+1}^{n+i} p_t(j) \cdot p(n+i-j,h)\,. \tag{13.54}$$

On the assumption of statistical equilibrium ($A < n$) we can leave the absolute points of time out of account:

$$p(i) = \left\{ \sum_{j=0}^{n} p(j) \right\} p(i,h) + \sum_{j=n+1}^{n+i} p(j) \cdot p(in+i-j,h), \qquad i = 0,1,\ldots \tag{13.55}$$

The system of equations (13.55) can only be solved directly by substitution, if we know the first n state probabilities $\{p(0), p(1), \ldots, p(n-1)\}$. In practice we may obtain numerical values by guessing an approximate set of values for $\{p(0),\ p(1),\ \ldots\ , p(n-1)\}$, then substitute these values in the recursion formula (13.55) and obtain new values. After a few approximations we obtain the exact values.

The explicit mathematical solution is obtained by means of generating functions (The *Erlang book*, [11] pp. 75–83).

13.5.6 Waiting time distribution: $M/D/n$, FCFS

The waiting time distribution is given by Crommelin's distribution:

$$p\{W \leq t\} = 1 - \sum_{i=0}^{n-1} \sum_{k=0}^{i} p(k) \cdot \sum_{j=1}^{\infty} \frac{\{A(j-\tau)\}^{(T+j+1)n-1-i}}{\{(T+j+1)n-1-i\}!}\,, \tag{13.56}$$

where A is the offered traffic and

$$t = T \cdot h + \tau, \qquad 0 \leq \tau < h\,. \tag{13.57}$$

Formula (13.56) can be written in a closed form in analogy with (13.50):

$$p\{W \leq t\} = \sum_{i=0}^{n-1} \sum_{k=0}^{i} p(k) \sum_{j=0}^{T} \frac{\{A(j-t)\}^{j \cdot n+n-1-i}}{\{j \cdot n+n-1-i\}!} \cdot \mathrm{e}^{-A(j-t)} \,. \tag{13.58}$$

For integral values of the waiting time t we have:

$$p\{W \leq t\} = \sum_{j=0}^{n(t+1)-1} p(j)\,. \tag{13.59}$$

For non-integral waiting times $t = T + \tau$, T integer, $0 \leq \tau < 1$, we are able to express the waiting time distribution in terms of integral waiting times as for *M/D/1*:

$$p\{W \leq t\} = p\{W \leq T+\tau\} = \mathrm{e}^{\lambda\tau} \sum_{j=0}^{k} \left\{ \frac{(-\lambda\tau)^j}{j!} \cdot \sum_{i=0}^{k-j} p(i) \right\} , \tag{13.60}$$

where $k = n(T+1)-1$ and $p(i)$ is the state probability (13.55).

The exact mean waiting time of all customers W is difficult to derive. An approximation was given by Molina:

$$W \approx \frac{n}{n+1} \cdot E_{2,n}(A) \cdot \frac{h}{n-A} \cdot \frac{1-\left(\frac{A}{n}\right)^{n+1}}{1-\left(\frac{A}{n}\right)^{n}} \,. \tag{13.61}$$

For any queueing system we have (3.20)

$$w = \frac{W}{D} \,,$$

where for all values of n:

$$D = 1 - \sum_{j=0}^{n-1} p(j)\,.$$

13.5.7 Erlang-k arrival process: $E_k/D/r$

Let us consider a queueing system with $n = r \cdot k$ servers (r, k integers), general arrival process *GI*, constant service time and ordered (*FCFS*) queueing discipline. Customers arriving during idle periods choose servers in cyclic order

$$1, 2, \ldots, n-1, n, 1, 2, \ldots$$

Then a certain server will serve just every $n'th$ customers as the customers due to the constant service time depart from the servers in the same order as they arrive at the servers. No customer can overtake another customer.

A group of servers made up from the servers

$$x, x+k, x+2\cdot k, \ldots, x+(r-1)\cdot k\,, \qquad 0 < x \leq k\,. \tag{13.62}$$

will serve just every $k'th$ customer. If we consider the servers (13.62), then considered as a single group they are equivalent to the queueing system $GI^{k*}/D/r$, where the arrival process GI^{k*} is a convolution of the arrival time distribution by itself k times.

The same goes for the k–1 other systems. The traffic in these k systems is mutually correlated, but if we only consider one system at a time, then this is a $GI^{k*}/D/n$, *FCFS* queueing system.

The assumption about cyclic hunting of the servers is not necessary within the individual systems (13.62). State probabilities, mean waiting times etc. are independent of the queueing discipline, which is of importance for the waiting time distribution only.

If we let the arrival process *GI* be a Poisson process, then GI^{k*} becomes an Erlang-k arrival process. We thus find that the following systems are equivalent with respect to the waiting time distribution:

$$M/D/r\cdot k,\ FCFS \quad \equiv \quad E_k/D/r,\ FCFS\ .$$

$E_k/D/r$ may therefore be dealt with by tables for $M/D/n$.

Example 13.5.1: Regular arrival processes
In general we know that for a given traffic per server the mean waiting time decreases when the number of servers increases (economy of scale, convexity). For the same reason the mean waiting time decreases when the arrival process becomes more regular. This is seen directly from the above decomposition, where the arrival process for $E_k/D/r$ becomes more regular for increasing k (r constant). For $A = 0.9$ erlang per server (L = mean queue length) we find:

$$\begin{array}{ll} \mathrm{E}_4/\mathrm{E}_1/2: & L = 4.5174\ , \\ \mathrm{E}_4/\mathrm{E}_2/2: & L = 2.6607\ , \\ \mathrm{E}_4/\mathrm{E}_3/2: & L = 2.0493\ , \\ \mathrm{E}_4/\mathrm{D}/2: & L = 0.8100\ . \end{array}$$

□

13.5.8 Finite queue system: *M/D/1/k*

In real systems we always have a finite queue. In computer systems the size of the storage is finite and in *ATM* systems we have finite buffers. The same goes for waiting positions in *FMS* (Flexible Manufacturing Systems).

The state probabilities of the finite buffer system is obtained from the state probabilities of the infinite buffer system by using (13.10) & (13.11).

Integral waiting times are obtained from the state probabilities, and non-integral waiting times from integral waiting times as shown above.

We may also find the finite buffer state probabilities in the following way. In a system with one server and $(k-1)$ queueing positions we have $(k+1)$ states $(0, 1, \cdots, k)$. The balance equations for states $\{0,\ 1,\ \ldots\ , k-2\}$ can be set up in the same way as Fry's equations of state. But it is not possible to write down simple time-independent equations for state $k-1$ and k. However, the first $(k-2)$ equations (13.42) together with the normalisation requirement

$$\sum_{j=0}^{k} p(j) = 1$$

and the fact that the offered traffic equals the carried traffic plus the rejected traffic (*PASTA property*):

$$A = 1 - p(0) + A \cdot p(k)$$

results in $(k+1)$ independent linear equations, which are easy to solve numerically. The two approaches yields the same result. The first method is only valid for $A < 1$, whereas the second is valid for any offered traffic.

Example 13.5.2: Leaky Bucket
Leaky Bucket is a mechanism for control of cell (packet) arrival processes from a user (source) in an *ATM*–system. The mechanism corresponds to a queueing system with constant service time (cell size) and a finite buffer. If the arrival process is a Poisson process, then we have an *M/D/1/k* system. The size of the leak corresponds to the long-term average acceptable arrival intensity, whereas the size of the bucket describes the excess (burst) allowed. The mechanism operates as a virtual queueing system, where the cells either are accepted immediately or are rejected according to the value of a counter which is the integral value of the load function (Fig. 13.2). In a contract between the user and the network an agreement is made on the size of the leak and the size of the bucket. On this basis the network is able to guarantee a certain grade-of-service. □

13.6 Single server queueing system: *GI/G/1*

In Sec. 13.3 we showed that the mean waiting time for all customers in queueing system *M/G/1* is given by Pollaczek-Khintchine's formula:

$$W = \frac{A \cdot s}{2(1-A)} \cdot \varepsilon \tag{13.63}$$

where ε is the form factor of the holding time distribution.

We have earlier analysed the following cases:

M/M/1 (Sec. 12.2.4): $\varepsilon = 2$:

$$W = \frac{A \cdot s}{(1-A)}\,, \qquad \text{Erlang 1917.} \tag{13.64}$$

M/D/1 (Sec. 13.5.3): $\varepsilon = 1$:

$$W = \frac{A \cdot s}{2(1-A)}\,, \qquad \text{Erlang 1909.} \tag{13.65}$$

It shows that the more regular the holding time distribution, the less becomes the waiting time traffic. (For loss systems with limited accessibility it is the opposite way: the bigger form factor, the less congestion).

In systems with non-Poisson arrivals, moments of higher order will also influence the mean waiting time.

13.6.1 General results

We have till now assumed that the arrival process is a Poisson process. For other arrival processes it is seldom possible to find an exact expression for the mean waiting time except in the case where the holding times are exponentially distributed. In general we may require, that either the arrival process or the service process should be Markovian. Till now there is no general accurate formulae for e.g. $M/G/n$.

For *GI/G/1* it is possible to give theoretical upper limits for the mean waiting time. Denoting the variance of the inter-arrival times by v_a and the variance of the holding time distribution by v_d, *Kingman's inequality* (1961) gives an upper limit for the mean waiting time:

$$\textit{GI/G/1:} \qquad W \leq \frac{A \cdot s}{2(1-A)} \cdot \left\{ \frac{v_a + v_d}{s^2} \right\}. \tag{13.66}$$

This formula shows that it is the stochastic variations, that results in waiting times.

Formula (13.66) gives the upper theoretical boundary. A realistic estimate of the actual mean waiting time is obtained by *Marchal's approximation* (Marchal, 1976 [77]):

$$W \approx \frac{A \cdot s}{2(1-A)} \cdot \left\{ \frac{v_a + v_d}{s^2} \right\} \cdot \left\{ \frac{s^2 + v_d}{a^2 + v_d} \right\}. \tag{13.67}$$

where a is the mean inter-arrival time ($A = s/a$). The approximation is a scaling of Kingman's inequality so it agrees with the Pollaczek-Khintchine's formula for the case $M/G/1$.

13.6.2 State probabilities: *GI/M/1*

As an example of a non-Poisson arrival process we shall analyse the queueing system *GI/M/1*, where the distribution of the inter-arrival times is a general distribution given by the density function $f(t)$. Service times are exponentially distributed with rate μ.

If the system is considered at an arbitrary point of time, then the state probabilities will not be described by a Markov process, because the probability of an arrival will depend on the time interval since the last arrival. The *PASTA* property is not valid.

However, if the system is considered immediately before (or after) an arrival epoch, then there will be independence in the traffic process since the inter-arrival times are stochastic independent the holding times are exponentially distributed. The arrival epochs are *equilibrium points* (regeneration points, Sec. 5.2.2), and we consider the so-called *embedded Markov chain*.

The probability that we *immediately before an arrival epoch* observe the system in state j is denoted by $\pi(j)$. In statistical equilibrium it can be shown that we will have the following result (D.G. Kendall, 1953 [61]):

$$\pi(i) = (1-\alpha)\alpha^i\,, \qquad i = 0, 1, 2, \ldots \tag{13.68}$$

where α is the positive real root satisfying the equation:

$$\alpha = \int_0^\infty \mathrm{e}^{-\mu(1-\alpha)t} f(t)\,\mathrm{d}t\,. \tag{13.69}$$

The steady state probabilities can be obtained by considering two successive arrival epochs t_1 and t_2 (similar to Fry's state equations, Sec. 13.5.5).

As the departure process is a Poisson process with the constant intensity μ when there are customers in the system, then the probability $p(j)$ that j customers complete service between two arrival epochs can be expressed by the number of events in a Poisson process during a stochastic interval (the inter-arrival time). We can set up the following state equations:

$$\begin{aligned}
\pi_{t_2}(0) &= \sum_{j=0}^{\infty} \pi_{t_1}(j) \cdot \left\{ 1 - \sum_{i=0}^{j} p(i) \right\}, \\
\pi_{t_2}(1) &= \sum_{j=0}^{\infty} \pi_{t_1}(j) \cdot p(j)\,, \\
&\vdots \\
\pi_{t_2}(i) &= \sum_{j=0}^{\infty} \pi_{t_1}(j) \cdot p(j-i+1)\,.
\end{aligned} \tag{13.70}$$

The normalisation condition is as usual:

$$\sum_{i=0}^{\infty} \pi_{t_1}(i) = \sum_{j=0}^{\infty} \pi_{t_2}(j) = 1\,. \tag{13.71}$$

It can be shown that the above-mentioned geometric distribution is the only solution to this system of equations (Kendall, 1953 [61]).

In principle, the queueing system *GI/M/n* can be solved in the same way. The state probability $p(j)$ becomes more complicated since the departure rate depends on the number of busy channels.

Notice that $\pi(i)$ is *not* the probability of finding the system in state i at an arbitrary point of time (time average), but the probability of finding the system in state i immediately before an arrival (call average).

13.6.3 Characteristics of *GI/M/1*

The probability of immediate service becomes:

$$p\{\text{immediate}\} = \pi(0) = 1 - \alpha\,. \tag{13.72}$$

The corresponding probability of being delayed the becomes:

$$D = p\{\text{delay}\} = \alpha\,. \tag{13.73}$$

The average number of busy servers at a random point of time (time average) is equal to the carried traffic (= the offered traffic $A < 1$).

The average number of *waiting* customers, immediately before the arrival of a customer, is obtained via the state probabilities:

$$\begin{aligned} L_1 &= \sum_{i=1}^{\infty} (1-\alpha)\,\alpha^i\,(i-1)\,, \\ L_1 &= \frac{\alpha^2}{1-\alpha}\,. \end{aligned} \tag{13.74}$$

The average number of customers in the system before an arrival epoch is:

$$\begin{aligned} L_2 &= \sum_{i=0}^{\infty} (1-\alpha)\,\alpha^i \cdot i \\ &= \frac{\alpha}{1-\alpha}\,. \end{aligned} \tag{13.75}$$

The average waiting time for all customers then becomes:

$$W = \frac{1}{\mu} \cdot \frac{\alpha}{1-\alpha}\,. \tag{13.76}$$

The average queue length taken over the whole time axis (the virtual queue length) therefore becomes (Little's theorem):

$$L = A \cdot \frac{\alpha}{1-\alpha}\,. \tag{13.77}$$

The mean waiting time for the customers, who obtain waiting times, becomes

$$\begin{aligned} w &= \frac{W}{D}\,, \\ w &= \frac{1}{\mu} \cdot \frac{1}{1-\alpha}\,. \end{aligned} \tag{13.78}$$

Example 13.6.1: Mean waiting times GI/M/1
For *M/M/1* we find $\alpha = \alpha_m = A$. For *D/M/1* $\alpha = \alpha_d$ is obtained from the equation:

$$\alpha_d = \mathrm{e}^{-\,(1-\alpha_d)\,/A}\,,$$

where α_d must be within (0,1). It can be shown that $0 < \alpha_d < \alpha_m < 1$. Thus the queueing system *D/M/1* will always have less mean waiting time than *M/M/1.*

For $A = 0.5$ erlang we find the following mean waiting times for all customers (13.76):

M/M/1:	$\alpha = 0.5\,,$	$W = 1\,,$	$w = 2\,.$
D/M/1:	$\alpha = 0.2032\,,$	$W = 0.2550\,,$	$w = 1.3423\,.$

where the mean holding time is used as the time unit ($\mu = 1$). The mean waiting time is thus far from proportional with the form factor of the distribution of the inter-arrival time. □

13.6.4 Waiting time distribution: *GI/M/1, FCFS*

When a customer arrives at the queueing system, the number of customers in the system is geometric distributed, and the customer therefore, under the assumption that he gets a positive waiting time, has to wait a geometrically distributed number of exponential phases. This will result in an exponentially distributed waiting time with a parameter given in (13.78), when the queueing discipline is *FCFS* (Sec. 12.4 and Fig. 4.9).

13.7 Round Robin and Processor-Sharing

The Round Robin (*RR*) queueing model (Fig. 13.6) is a model for a time-sharing computer system, where we wish a fast response time for the shortest jobs. This queueing discipline is also called *fair queueing* because the available resources are equally distributed among the jobs (customers) in the system.

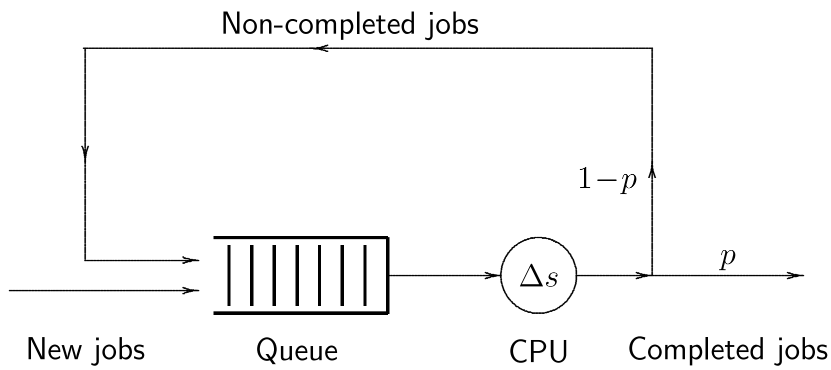


Figure 13.6: *Round robin queueing system. A task is allocated a time slice Δs (at most) every time it is served. If the task is not finished during this time slice, it is returned to a FCFS queue, where it waits on equal terms with new tasks. If we let Δs decrease to zero we obtain the queueing discipline PS (Processor Sharing).*

New jobs are placed in a *FCFS*–queue, where they wait until they obtain service within a time slice (slot) Δs which is the same for all jobs. If a job does is not completed within a time slice, the service is interrupted, and the job is placed at the end of the *FCFS*–queue. This continues until the required total service time is fulfilled.

We assume that the queue is unlimited, and that new jobs arrive according to a Poisson process (λ). The service time distribution can be general with the mean value s.

The time slice can vary. If it becomes infinite, all jobs will be completed the first time, and we have simply an *M/G/1* queueing system with *FCFS* discipline. If we let the time slice decrease to zero, then we get the *PS* = *Processor-Sharing* model, which has a number of nice analytical properties. The *PS*–was introduced by Kleinrock (1967) and is dealt with in detail in (Kleinrock, 1976 [66]).

The Processor-Sharing model can be interpreted as a queueing system where all jobs are served continuously by the server (time sharing). If there are i jobs in the system, each of them obtain the fraction $1/i$ of the capacity of the computer. So there is no queue, and the queueing discipline is meaningless.

When the offered traffic $A = \lambda \cdot s$ is less than one, it can be shown that the steady state probabilities are given by:

$$p(i) = (1 - A) \cdot A^i \,, \qquad i = 0, 1, \ldots \,, \tag{13.79}$$

i.e. a geometric distribution with the mean value $A/(1-A)$. The mean holding time (average response time) for the jobs with duration t becomes:

$$R_t = \frac{t}{1-A}\,. \tag{13.80}$$

If this job was alone in the system, then its holding time would be t. Since there is no queue, we can then talk about an average delay for jobs with duration t:

$$\begin{aligned} W_t &= R_t - t \\ &= \frac{A}{1-A}\cdot t\,. \end{aligned} \tag{13.81}$$

The corresponding mean values for a random job naturally becomes:

$$R = \frac{s}{1-A}\,, \tag{13.82}$$

$$W = \frac{A}{1-A}\cdot s\,. \tag{13.83}$$

This shows that we obtain exactly the same mean values as for *M/M/1* (Sec. 12.2.4). But the actual mean waiting time becomes proportional to the duration of the job, which is often a desirable property. We don't assume any knowledge in advance about the duration of the job. The mean waiting time becomes proportional to the mean service time. The proportionality should *not* be understood in the way that two jobs of the same duration have the same waiting time; it is only valid on the average. In comparison with the results we have earlier obtained for *M/G/1* (Pollaczek-Khintchine's formula (13.2)) the results may surprise the intuition.

A very useful property of the Processor-Sharing model is that the departure process is a Poisson process as the arrival process (Sec. 14.2). It is intuitively explained by the fact that the departure process is obtained from the arrival process by a stochastic shifting of the individual arrival epochs. The time shift is equal to the response time with a mean value given by (13.80) (Sec. 6.3.1, Palm's theorem).

The Processor-Sharing model is very useful for analysing time-sharing systems and for modelling queueing networks (Chap. 14).

Chapter 14

Networks of queues

Many systems can be modelled in such a way that a customer achieves services from several successive nodes, i.e. once he has obtained service at one node, then he goes on to another node. The total service demand is composed of service demands at several nodes. Hence, the system is a network of queues, a *queueing network* where each individual queue is called a *node*. Examples of queueing networks are telecommunication systems, computer systems, packet switching networks, and *FMS* (*F*lexible *M*anufacturing *S*ystems). In queueing networks we define the queue-length in a node as the total number of customers in the node, including customers being served.

The aim of this chapter is to introduce the basic theory of queueing networks, illustrated by applications. Usually, the theory is considered as being rather complicated, which is mainly due to the complex notation. However, in this chapter we will give a simple introduction to general analytical queueing network models based on product forms, the convolution algorithm, the *MVA*–algorithm, and examples.

The theory of queueing networks is analogous to the theory of multi–dimensional loss systems (Chap. 10 & 11). In Chap. 10 we considered multi-dimensional loss systems whereas in this chapter we are looking at networks of queueing systems.

14.1 Introduction to queueing networks

Queueing networks are classified as closed and open queueing networks. In *closed queueing networks* the number of customers is fixed whereas in *open queueing networks* the number of customers is varying. In principle, an open network can be transformed into a closed network by adding an extra node.

Erlang's classical waiting system, $M/M/n$, is an example of an open queueing system, whereas Palm's machine/repair model with S terminals is a closed network. If there is more than one type of customers, the network can be a mixed closed and open network. Since the departure process from one node is the arrival process at another node, we shall pay special attention to the departure process, in particular when it can modelled as a Poisson process. This is investigated in the section on symmetric queueing systems (Sec. 14.2).

The state of a queueing network is defined as the simultaneous distribution of number of customers in each node. If K denotes the total number of nodes, then the state is described by a vector $p(i_1, i_2, \ldots, i_K)$ where i_k is the number of customers in node k $(k = 1, 2, \ldots, K)$. Frequently, the state space is very large and it is difficult to calculate the state probabilities by solving node balance equations. If every node is a symmetric queueing system, for example a Jackson network (Sec. 14.3), then we will have product form. The state probabilities of networks with product form can be aggregated and obtained by using the convolution algorithm (Sec. 14.4.1) or the *MVA*–algorithm (Sec. 14.4.2).

Jackson networks can be generalised to *BCMP*–networks (Sec. 14.5), where there are N types of customers. Customers of one specific type all belongs to a so-called *chain*. Fig. 14.1 illustrates an example of a queueing network with 4 chains. When the number of chains increases the state space increases correspondingly, and only systems with a small number of chains can be calculated exactly. In case of a multi-chain network, the state of each node becomes multi-dimensional (Sec. 14.6). The product form between nodes is maintained, and the *convolution* and the *MVA*–algorithm are applicable (Sec. 14.7). A number of approximate algorithms for large networks can found in the literature.

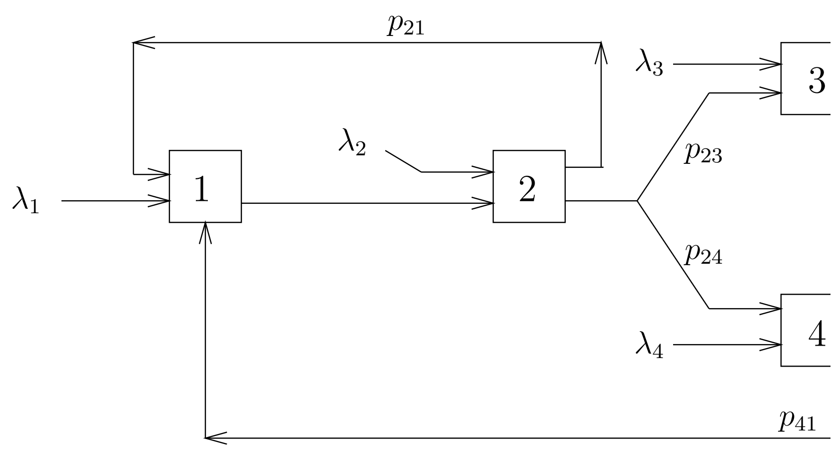


Figure 14.1: *An example of a queueing network with four open chains.*

14.2 Symmetric queueing systems

In order to analyse queueing systems, it is important to know when the departure process of a queueing system is a Poisson process. Four queueing models are known to have this property:

1. *M/M/n*. This is *Burke's theorem* (Burke, 1956 [12]), which states, that the *departure process* of an *M/M/n*–system is a Poisson process. The state space probabilities are given by (12.2):

$$p(i) = \begin{cases} \dfrac{A^i}{i!} \cdot p(0)\,, & 0 \le i \le n\,, \\ \left(\dfrac{A}{n}\right)^{i-n} \cdot p(n)\,, & i \ge n\,. \end{cases} \tag{14.1}$$

   where $A = \lambda/\mu$.

2. *M/G/∞*. This corresponds to the Poisson case (Sec. 7.2). From Sec. 6.3 we know that a random translation of the events of a Poisson process results in a new Poisson process. This model is sometimes denoted as a system with the queueing discipline *IS*, *I*nfinite number of *S*ervers. The state probabilities are given by the Poisson distribution (7.6):

$$p(i) = \frac{A^i}{i!} \cdot \mathrm{e}^{-A}\,, \qquad i = 0, 1, 2, \ldots\ . \tag{14.3}$$

3. *M/G/1–PS*. This is a single server queueing system with a general service time distribution and processor sharing. The state probabilities are similar to the *M/M/1* case (13.79):

$$p(i) = (1 - A) \cdot A^i\,, \qquad i = 0, 1, 2, \ldots\ . \tag{14.4}$$

4. *M/G/1–LCFS-PR* (PR = Preemptive Resume). This system also has the same state space probabilities as *M/M/1* (14.4).

In the theory of queueing networks usually only these four queueing disciplines are considered. But for example also for Erlang's loss system, the departure process will be a Poisson process, if we include blocked customers.

The above-mentioned four queueing systems are called *symmetric queueing systems* as they are symmetric in time. Both the arrival process and the departure process are Poisson processes and the systems are reversible (Kelly, 1979 [59]). The process is called reversible because it looks the same way when we reverse the time (cf. when a film is reversible it looks the same whether we play it forward or backward). Apart from *M/M/n* these symmetric queueing systems have the common feature that a customer is served immediately upon arrival. In the following we mainly consider *M/M/n* nodes, but the *M/M/1* model also includes *M/G/1–PS* and *M/G/1–LCFS–PR*.

14.3 Jackson's theorem

In 1957, J.R. Jackson who was working with production planning and manufacturing systems, published a paper with a theorem, now called *Jackson's theorem* (1957 [45]). He showed that a queueing network of $M/M/n$ – nodes has product form. Knowing the fundamental theorem of Burke (1956 [12]) Jackson's result is obvious. Historically, the first paper on queueing systems in series was by R.R.P. Jackson (1954 [44]).

Theorem 14.1 Jackson's theorem: *Consider an open queueing network with K nodes satisfying the following conditions:*

a) Each node is an M/M/n–queueing system. Node k has n_k servers, and the average service time is $1/\mu_k$.

b) Customers arrive from outside the system to node k according to a Poisson process with intensity λ_k. Customers may also arrive from other nodes to node k.

c) A customer, who has just finished his service at node j, immediately transfers to node k with probability p_{jk} or leaves the network with probability:

$$1 - \sum_{k=1}^{K} p_{jk}\,.$$

A customer may visit the same node several times if $p_{kk} > 0$.

The average arrival intensity Λ_k at node k is obtained by looking at the flow balance equations:

$$\Lambda_k = \lambda_k + \sum_{j=1}^{K} \Lambda_j \cdot p_{jk}\,. \tag{14.5}$$

Let $p(i_1, i_2, \ldots, i_K)$ denote the state space probabilities under the assumption of statistical equilibrium, i.e. the probability that there is i_k customers at node k. Furthermore, we assume that

$$\frac{\Lambda_k}{\mu_k} = A_k < n_k\,. \tag{14.6}$$

Then the state space probabilities are given on *product form:*

$$p\left(i_1, i_2, \ldots, i_K\right) = \prod_{k=1}^{K} p_k\left(i_k\right)\,. \tag{14.7}$$

Here for node k, $p_k(i_k)$ is the state probabilities of an $M/M/n$ queueing system with arrival intensity Λ_k and service rate μ_k (14.1). The offered traffic Λ_k/μ_k to node k must be less than the capacity n_k of the node to enter statistical equilibrium (14.6). The key point of Jackson's theorem is that each node can be considered independently of all other nodes and that the state probabilities are given by Erlang's C–formula. This simplifies the calculation of the state space probabilities significantly. The proof of the theorem was derived by Jackson in 1957 by showing that the solution satisfy the balance equations for statistical equilibrium. Jackson's first model thus only deals with open queueing networks.

In Jackson's second model (Jackson, 1963 [46]) the arrival intensity from outside:

$$\lambda = \sum_{j=1}^{K} \lambda_j \tag{14.8}$$

may depend on the current number of customers in the network. Furthermore, μ_k can depend on the number of customers at node k. In this way, we can model queueing networks which are either closed, open, or mixed. In all three cases, the state probabilities have product form.

The model by Gordon & Newell (1967 [31]), which is often cited in the literature, can be treated as a special case of Jackson's second model.

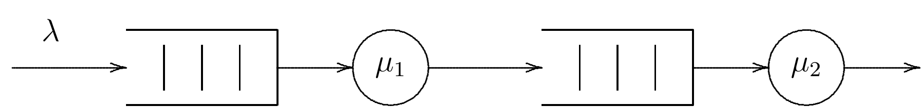


Figure 14.2: *State transition diagram of an open queueing network consisting of two M/M/1–systems in series.*

Example 14.3.1: Two M/M/1 nodes in series
Fig. 14.2 shows an open queueing network of two *M/M/1* nodes in series. The corresponding state transition diagram is given in Fig. 14.3. Clearly, the state transition diagram is not reversible: (between two neighbour states there is only flow in one direction, (cf. Sec. 10.2) and apparently there is no product form. If we solve the balance equations to obtain the state probabilities we find that the solution can be written on a product form:

$$\begin{aligned} p(i,j) &= p(i) \cdot p(j)\,, \\ p(i,j) &= \left\{(1-A_1)\cdot A_1^i\right\} \cdot \left\{(1-A_2)\cdot A_2^j\right\}\,, \end{aligned}$$

where $A_1 = \lambda/\mu_1$ and $A_2 = \lambda/\mu_2$. The state probabilities can be expressed in a product form $p(i,j) = p(i) \cdot p(j)$, where $p(i)$ is the state probabilities for a $M/M/1$ system with offered traffic A_1 and $p(j)$ is the state probabilities for a $M/M/1$ system with offered traffic A_2. The state probabilities of Fig. 14.3 are identical to those of Fig. 14.4, which has local balance and product form. Thus it is possible to find a system which is reversible and has the same state probabilities as the non-reversible system. There is *regional* but not *local* balance in Fig. 14.3. If we consider a square of four states, then to the outside world there will be balance, but internally there will be circulation via the diagonal state shift. □

In queueing networks customers will often be looping, so that a customer may visit the same node several times. If we have a queueing network with looping customers, where the nodes are $M/M/n$–systems, then the arrival processes to the individual nodes are no more Poisson processes. Anyway, we may calculate the state probabilities as if the individual nodes are independent $M/M/n$ systems. This is explained in the following example.

Example 14.3.2: Networks with feed back
Feedback is introduced in *Example 14.3.1* by letting a customer, which has just ended its service at node 2, return to node 1 with probability p_{21}. With probability $1 - p_{21}$ the customer leaves the system. The flow balance equations (14.5) gives the total arrival intensity to each node and p_{21} must be chosen such that both Λ_1/μ_1 and Λ_2/μ_2 are less than one. Letting $\lambda_1 \to 0$ and $p_{21} \to 1$ we realise that the arrival processes are not Poisson processes: only rarely a new customer will arrive, but once he has entered the system he will circulate for a relatively long time. The number of circulations will be geometrically distributed and the inter-arrival time is the sum of the two service times. I.e. when there is one (or more) customers in the system, then the arrival rate to each node will be relatively high, whereas the rate will be very low if there is no customers in the system. The arrival process will be *bursty*.
The situation is similar to the decomposition of an exponential distribution into a weighted sum of *Erlang-k* distributions, with geometrical weight factors (Sec. 4.4). Instead of considering a single exponential inter-arrival distribution we can decompose this into k phases (Fig. 4.9) and consider each phase as an arrival. Hence, the arrival process has been transformed from a Poisson process to a process with bursty arrivals. □

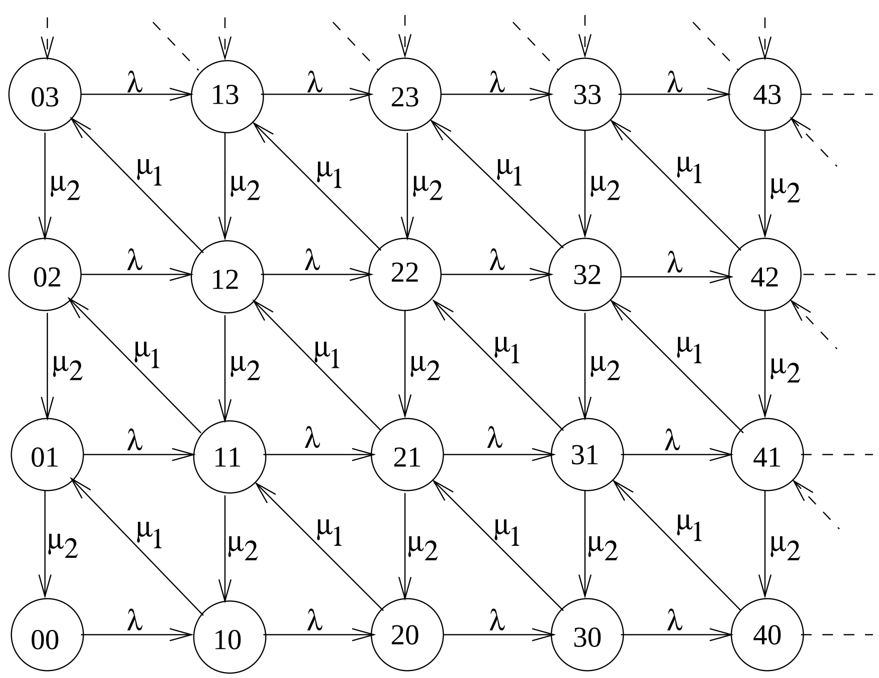


Figure 14.3: *State transition diagram for the open queueing network shown in Fig. 14.2. The diagram is non–reversible.*

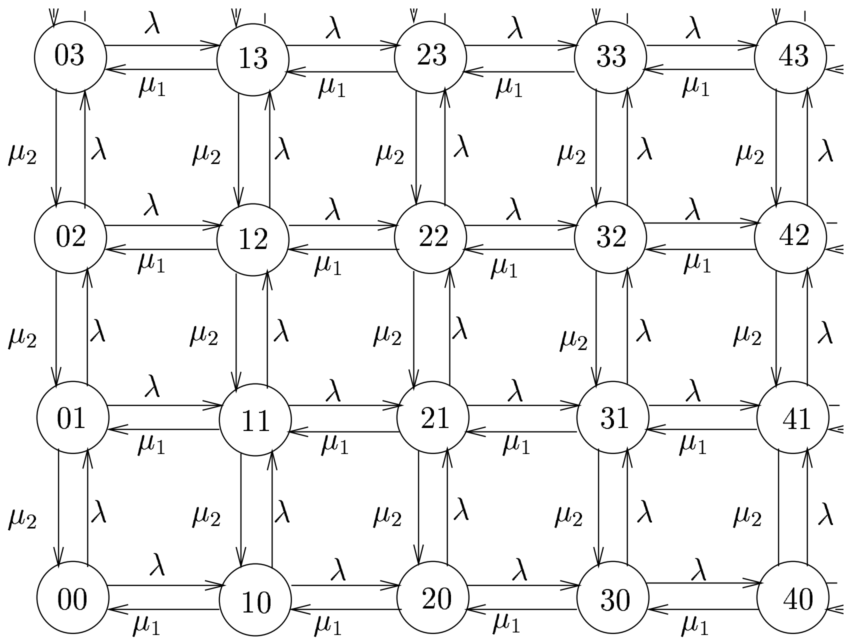


Figure 14.4: *State transition diagram for two independent $M/M/1$–queueing systems with identical arrival intensity, but individual mean service times. The diagram is reversible.*

14.3.1 Kleinrock's independence assumption

If we consider a real-life data network, then the packets will have the same constant length, and therefore the same service time on all links and nodes of equal speed. The theory of queueing networks assumes that a packet (a customer) samples a new service time in every node. This is a necessary assumption for the product form. This assumption was first investigated by Kleinrock (1964 [64]), and it turns out to be a good approximation in praxis.

14.4 Single chain queueing networks

We are interested in the state probabilities defined by $p(i_1, i_2, \ldots, i_k, \ldots, i_K)$, where i_k is the number of customers in node k $(1 \leq k \leq K)$.
Dealing with *open systems* is easy. First we solve the flow balance equation (14.5) and obtain the aggregated arrival intensity to each node (Λ_k). Combining the arrival intensities with the service time distribution (μ_k) we get the offered traffic A_k at each node and then by considering Erlang's delay system we get the state probabilities for each node.

14.4.1 Convolution algorithm for a closed queueing network

Dealing with *closed queueing networks* is much more complicated. We only know the relative load at each node, not the absolute load, i.e. $c \cdot \Lambda_j$ is known, but c is unknown. We can obtain the unnormalised relative state probabilities. Finally, by normalising we get the normalised state probabilities. Unfortunately, the normalisation implies that we must sum up all state probabilities, i.e. we must calculate each (unnormalised) state probability. The number of states increases rapidly when the number of nodes and/or customers increases. In general, it is only possible to deal with small systems. The complexity is similar to that of multi dimensional loss systems (Chapter 10).

We will now show how the convolution algorithm can be applied to queueing networks. The algorithm corresponds to the convolution algorithm for loss systems (Chapter 10). We consider a queueing network with K nodes and a single chain with S customers. We assume that the queueing systems in each node are symmetric (Sec. 14.2). The algorithm has three steps:

- *Step 1.* Let the arrival intensity to an arbitrary chosen reference node i be equal to some value Λ_i. By solving the flow balance equation (14.5) for the closed network we obtain the relative arrival rates Λ_k $(1 \leq k \leq K)$ to all nodes. Finally, we have the relative offered traffic $\alpha_k = \Lambda_k/\mu_k$. Often we choose the above arrival intensity of the reference node so that the offered traffic to this node becomes one.

- *Step 2.* Consider each node as if it is isolated and has the offered traffic α_k $(1 \leq k \leq K)$. Depending on the actual symmetric queueing system at node k, we derive the relative state probabilities $q_k(i)$ at node k. The state space will be limited by the total number of customers S, i.e. $0 \leq i \leq S$.

- *Step 3.* Convolve the state probabilities for each node recursively. For example, for the first two nodes we have:
$$q_{12} = q_1 * q_2 \,, \tag{14.9}$$
where
$$q_{12}(i) = \sum_{x=0}^{i} q_1(x) \cdot q_2(i-x), \qquad i = 0, 1, \ldots, S \,.$$
When all nodes has been convolved we get:
$$q_{1,2,\ldots,K} = q_{1,2,\ldots,K-1} * q_K \,. \tag{14.10}$$
Since the total number of customers is fixed (S) only state $q_{1,2,\ldots,K}(S)$ exists in the aggregated system and therefore this macro-state must have the probability one. We can then normalise all micro-state probabilities.
When we perform the last convolution we can derive the performance measures for the last node. By changing the order of convolution of the nodes we can obtain the performance measures of all nodes.

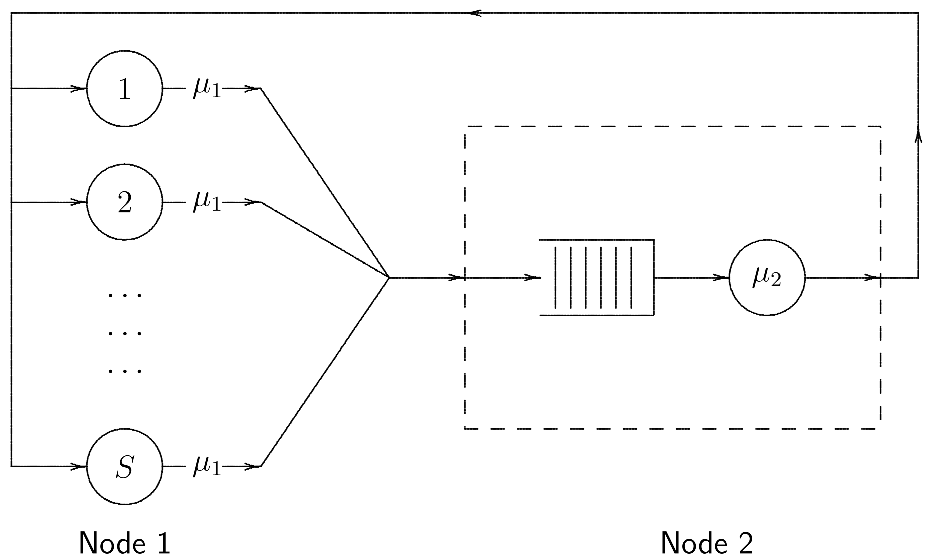


Figure 14.5: *The machine/repair model as a closed queueing networks with two nodes. The terminals correspond to one IS–node, because the tasks always find an idle terminal, whereas the CPU corresponds to an M/M/1–node.*

Example 14.4.1: Palm's machine/repair model
We now consider the machine/repair model of Palm introduced in Sec. 12.5 as a closed queueing network (Fig. 14.5). There are S customers and terminals. The mean *thinking time* is μ_1^{-1} and the mean service time at the *CPU* is μ_2^{-1}. In queueing network terminology there are two nodes: node one is the terminals, i.e. an $M/G/\infty$ (actually it is an $M/G/S$ system, but since the number of customers is limited to S it corresponds to an $M/G/\infty$ system), and node two is the CPU, i.e. an *M/M/1* system with service intensity μ_2.
The flows to the nodes are equal ($\lambda_1 = \lambda_2 = \lambda$) and the relative load at node 1 and node 2 are

$$\alpha_1 = \lambda/\mu_1 \text{ and } \alpha_2 = \lambda/\mu_2 \,,$$

respectively. If we consider each node in isolation we obtain the state probabilities of each node, $q_1(i)$ and $q_2(j)$, and by convolving $q_1(i)$ and $q_2(j)$ we get $q_{12}(x)$, $(0 \le x \le S)$, as shown in Table 14.1. The last term with S customers (an unnormalised probability) $q_{12}(S)$ is compounded of:

$$q_{12}(S) = \alpha_2^S \cdot 1 + \alpha_2^{S-1} \cdot \alpha_1 + \alpha_2^{S-2} \cdot \frac{\alpha_1^2}{2!} + \cdots + 1 \cdot \frac{\alpha_1^S}{S!} \,.$$

A simple rearranging yields:

$$q_{12}(S) = \alpha_2^S \cdot \left\{ 1 + \frac{\varrho}{1} + \frac{\varrho^2}{2!} + \cdots + \frac{\varrho^S}{S!} \right\} ,$$

where

$$\varrho = \frac{\alpha_1}{\alpha_2} = \frac{\mu_2}{\mu_1} \,.$$

State i	Node 1 $q_1(i)$	Node 2 $q_2(i)$	Queueing network $q_{12} = q_1 * q_2$
0	1	1	1
1	α_1	α_2	$\alpha_1 + \alpha_2$
2	$\frac{\alpha_1^2}{2!}$	α_2^2	$\alpha_2^2 + \alpha_1 \cdot \alpha_2 + \frac{\alpha_1^2}{2!}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
i	$\frac{\alpha_1^i}{i!}$	α_2^i	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
S	$\frac{\alpha_1^S}{S!}$	α_2^S	$q_{12}(S)$

Table 14.1: *The convolution algorithm applied to Palm's machine/repair model. Node 1 is an IS-system, and node two is an M/M/1-system (Example 14.4.1).*

The probability that all terminals are *"thinking"* is identified as the last term (normalised by the sum) (S terminals in node 1, zero terminals in node 2):

$$\frac{\frac{\varrho^S}{S!}}{1 + \varrho + \frac{\varrho^2}{2!} + \frac{\varrho^3}{3!} + \cdots + \frac{\varrho^S}{S!}} = E_{1,S}(\varrho)\,,$$

which is Erlang's *B-formula*. Thus the result is in agreement with the result obtained in Sec. 12.5. We notice that λ appears with the same power in all terms of $q_{1,2}(S)$ and thus corresponds to a constant which disappears when we normalise. □

Example 14.4.2: Central server system
In 1971 J. P. Buzen introduced the *central server* model illustrated in Fig. 14.6 to model a multi-programmed computer system with one CPU and a number of input/output channels (peripheral units). The degree of multi-programming S describes the number of jobs processed simultaneously. The number of peripheral units is denoted by $K-1$ as shown in Fig. 14.6, which also shows the transition probabilities.

Typically a job requires service hundreds of times, either by the central unit or by one of the peripherals. We assume that once a job is finished it is immediately replaced by another job, hence S is constant. The service times are all exponentially distributed with intensity μ_i ($i = 1, \ldots, K$).

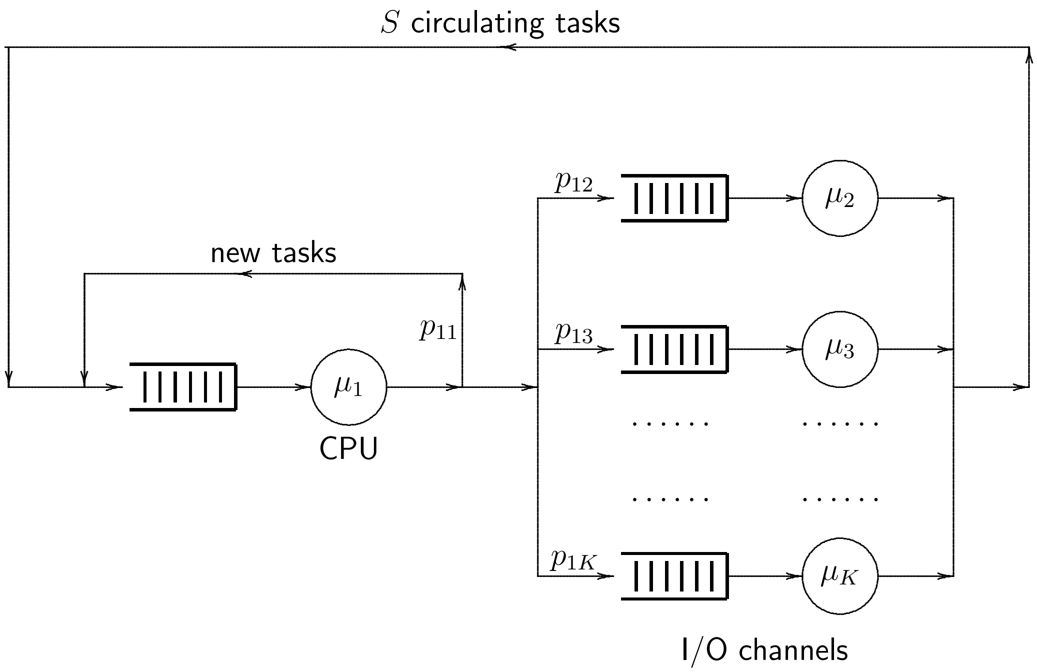


Figure 14.6: *Central server queueing system consisting of one central server (CPU) and (K–1) I/O–channels. A fixed number of tasks S are circulating in the system.*

Buzen drew up a scheme to evaluate this system. The scheme is a special case of the convolution algorithm. Let us illustrate it by a case with $S = 4$ customers and $K = 3$ nodes and:

$$\mu_1 = \frac{1}{28}\,, \quad \mu_2 = \frac{1}{40}\,, \quad \mu_3 = \frac{1}{280}\,,$$

$$p_{11} = 0.1\,, \quad p_{12} = 0.7\,, \quad p_{13} = 0.2\,.$$

The relative loads become:

$$\alpha_1 = 1\,, \qquad \alpha_2 = 1\,, \qquad \alpha_3 = 2\,.$$

If we apply the convolution algorithm we obtain the results shown in Table 14.2. The term $q_{123}(4)$ is made up by:

$$q_{123}(4) = 1 \cdot 16 + 2 \cdot 8 + 3 \cdot 4 + 4 \cdot 2 + 5 \cdot 1 = 57\,.$$

Node 3 serves customers in all states except for state $q_3(0) \cdot q_{12}(4) = 5$. The utilisation of node 3 is therefore $a_3 = 52/57$. Based on the relative loads we now obtain the exact loads:

$$a_1 = \frac{26}{57}\,, \qquad a_2 = \frac{26}{57}\,, \qquad a_3 = \frac{52}{57}\,.$$

State	Node 1	Node 2	Node 1*2	Node 3	Queueing network
i	$q_1(i)$	$q_2(i)$	$q_{12} = q_1 * q_2$	q_3	$q_{123} = (q_1 * q_2) * q_3$
0	1	1	1	1	1
1	1	1	2	2	4
2	1	1	3	4	11
3	1	1	4	8	26
4	1	1	5	16	57

Table 14.2: *The convolution algorithm applied to the central server system.*

The average number of customers at node 3 is:

$$L_3 = \{1 \cdot (4 \cdot 2) + 2 \cdot (3 \cdot 4) + 3 \cdot (2 \cdot 8) + 4 \cdot (1 \cdot 16)\} \,/\, 57\,,$$

$$L_3 = \frac{144}{57}\,.$$

By changing the order of convolution we get the average queue lengths L_1 and L_2 and ends up with:

$$L_1 = \frac{42}{57}\,, \quad L_2 = \frac{42}{57}\,, \quad L_3 = \frac{144}{57}\,.$$

The sum of all average queue lengths is of course equal to the number of customers S. Notice, that in queueing networks we define the queue length as the total number of customers in the node, including customers being served. From the utilisation and mean service time we find the average number of customers finishing service per time unit at each node:

$$\lambda_1 = \frac{26}{57} \cdot \frac{1}{28}\,, \quad \lambda_2 = \frac{26}{57} \cdot \frac{1}{40}\,, \quad \lambda_3 = \frac{52}{57} \cdot \frac{1}{280}\,.$$

Applying Little's result we finally obtain the mean sojourn time $W_k = L_k/\lambda_k$:

$$W_1 = 45.23\,, \quad W_2 = 64.62\,, \quad W_3 = 775.38\,.$$

□

14.4.2 The *MVA*–algorithm

The Mean Value Algorithm (*MVA*) is an algorithm to calculate performance measures of queueing networks. It combines in an elegant way two main results in queueing theory: the arrival theorem (8.28) and Little's law (5.20). The algorithm was first published by Lavenberg & Reiser (1980 [72]).

We consider a queueing network with K nodes and S customers (all belonging to a single chain). The relative loads of the nodes are denoted by α_k $(k = 1, 2, \ldots, K)$. The algorithm is recursively in the number of customers, i.e. a network with $x + 1$ customers is evaluated from a network with x customers.

Assume that the average number of customers at node k is $L_k(x)$ where x is the total number of customers in the network. Obviously

$$\sum_{k=1}^{K} L_k(x) = x \,. \tag{14.11}$$

The algorithm goes recursively in two steps:

Step 1:
Increase the number of customers from x to $(x + 1)$. According to the arrival theorem, the $(x + 1)th$ customer will see the system as a system with x customers in statistically equilibrium. Hence, the average sojourn time (waiting time + service time) at node k is:

- For *M/M/1, M/G/1–PS, and M/G/1–LCFS–PR:*

$$W_k(x + 1) = \{L_k(x) + 1\} \cdot s_k \,.$$

- For *M/G/∞:*

$$W_k(x + 1) = s_k \,.$$

where s_k is the average service time in node k which has n_k servers. As we only calculate mean waiting times, we may assume *FCFS* queueing discipline.

Step 2:
We apply Little's law ($L = \lambda \cdot W$), which is valid for all systems in statistical equilibrium. For node k we have:

$$L_k(x{+}1) = c \cdot \lambda_k \cdot W_k(x{+}1) \,,$$

where λ_k is the relative arrival rate to node k. The normalising constant c is obtained from the total number of customers::

$$\sum_{k=1}^{K} L_k(x{+}1) = x + 1 \,. \tag{14.12}$$

By these two steps we have performed the recursion from x to $(x + 1)$ customers. For $x = 1$ there will be no waiting time in the system and $W_k(1)$ equals the average service time s_k.

The *MVA*–algorithm is below shown for a single server nodes, but it is fairly easy to generalise to nodes with either multiple servers or even infinite server discipline.

Example 14.4.3: Central server model
We apply the *MVA*–algorithm to the central server model (Example 14.4.2). The relative arrival rates are:

$$\lambda_1 = 1\,, \qquad \lambda_2 = 0.7\,, \qquad \lambda_3 = 0.2\,.$$

	Node 1	Node 2	Node 3
$S=1$	$W_1(1) = 28$ $L_1(1) = c \cdot 1 \cdot 28$ $L_1(1) = 0.25$	$W_2(1) = 40$ $L_2(1) = c \cdot 0.7 \cdot 40$ $L_2(1) = 0.25$	$W_3(1) = 280$ $L_3(1) = c \cdot 0.2 \cdot 280$ $L_3(1) = 0.50$
$S=2$	$W_1(2) = 1.25 \cdot 28$ $L_1(2) = c \cdot 1 \cdot 1.25 \cdot 28$ $L_1(2) = 0.4545$	$W_2(2) = 1.25 \cdot 40$ $L_2(2) = c \cdot 0.7 \cdot 1.25 \cdot 40$ $L_2(2) = 0.4545$	$W_3(2) = 1.50 \cdot 280$ $L_3(2) = c \cdot 0.2 \cdot 1.50 \cdot 280$ $L_3(2) = 1.0909$
$S=3$	$W_1(3) = 1.4545 \cdot 28$ $L_1(3) = c \cdot 1 \cdot 1.4545 \cdot 28$ $L_1(3) = 0.6154$	$W_2(3) = 1.4545 \cdot 40$ $L_2(3) = c \cdot 0.7 \cdot 1.4545 \cdot 40$ $L_2(3) = 0.6154$	$W_3(3) = 2.0909 \cdot 280$ $L_3(3) = c \cdot 0.2 \cdot 2.0909 \cdot 280$ $L_3(3) = 1.7692$
$S=4$	$W_1(4) = 1.6154 \cdot 28$ $L_1(4) = c \cdot 1 \cdot 1.6154 \cdot 28$ $L_1(4) = 0.7368$	$W_2(4) = 1.6154 \cdot 40$ $L_2(4) = c \cdot 0.7 \cdot 1.6154 \cdot 40$ $L_2(4) = 0.7368$	$W_3(4) = 2.7692 \cdot 280$ $L_3(4) = c \cdot 0.2 \cdot 2.7692 \cdot 280$ $L_3(4) = 2.5263$

Naturally, the result is identical to the one obtained with the convolution algorithm. The sojourn time at each node (using the original time unit):

$$\begin{aligned} W_1(4) &= 1.6154 \cdot 28 &= 45.23\,, \\ W_2(4) &= 1.6154 \cdot 40 &= 64.62\,, \\ W_3(4) &= 2.7693 \cdot 280 &= 775.38\,. \end{aligned}$$

□

Example 14.4.4: MVA-algorithm applied to the machine/repair model
We consider the machine/repair model with S sources, terminal thinking time A and *CPU*–service time equal to one time unit. As mentioned in Sec. 12.5.2 this is equivalent to Erlang's loss system with S servers and offered traffic A. It is also a closed queueing network with two nodes and S customers in one chain. If we apply the *MVA*–algorithm to this system, then we get the recursion formula for the Erlang–B formula (7.27). The relative visiting rates are identical, as a customer alternatively visits node one and two: $\lambda_1 = \lambda_2 = 1$.

	Node 1	Node 2
$S=1$	$W_1(1) = A$ $L_1(1) = c \cdot 1 \cdot A$ $L_1(1) = \frac{A}{1+A}$	$W_2(1) = 1$ $L_2(1) = c \cdot 1 \cdot 1$ $L_2(1) = \frac{1}{1+A}$
$S=2$	$W_1(2) = A$ $L_1(2) = c \cdot 1 \cdot A$ $L_1(2) = A \cdot \frac{1+A}{1+A+\frac{A^2}{2!}}$	$W_2(2) = 1 + \frac{1}{1+A}$ $L_2(2) = c \cdot 1 \cdot (1 + \frac{1}{1+A})$ $L_2(2) = 2 - A \cdot \frac{1+A}{1+A+\frac{A^2}{2!}}$

We know that the queue-length at the terminals (node 1) is equal to the carried traffic in the equivalent Erlang–B system and that all other customers stay in the *CPU* (node 2). We thus have in general:

	Node 1	Node 2
$S = x$	$W_1(x) = A$	$W_2(x) = 1 + L_2(x-1)$
	$L_1(x) = c \cdot A$	$L_2(x) = c \cdot \{1 + L_2(x-1)\}$
	$L_1(x) = A \cdot \{1 - E_x(A)\}$	$L_2(x) = x - A \cdot \{1 - E_x(A)\}$

From this we have the normalisation constant $c = 1 - E_x(A)$ and find for the (x+1)'th customer:

$$
\begin{aligned}
L_1(x+1) + L_2(x+1) &= c \cdot A + c \cdot \{1 + L_2(x)\} \\
&= c \cdot A + c \cdot \{1 + x - A \cdot (1 - E_x)\} \\
&= x + 1\,, \\
E_{x+1} &= \frac{A \cdot E_x}{x + 1 + A \cdot E_x}\,,
\end{aligned}
$$

because we know $c = 1 - E_{x+1}$. This is just the recursion formula for the Erlang–B formula. □

14.5 BCMP queueing networks

In 1975 the second model of Jackson was further generalised by *B*askett, *C*handy, *M*untz and *P*alacios (1975 [4]). They showed that queueing networks with more than one type of customers also have product form, provided that:

a) Each node is a symmetric queueing system (cf. Sec. 14.2: Poisson arrival process ⇒ Poisson departure process).

b) The customers are classified into N *chains*. Each chain is characterised by its own mean service time s_i and transition probabilities p_{ij}. Furthermore, a customer may change from one chain to another chain with a certain probability after finishing service at a node. A restriction applies if the queueing discipline at a node is *M/M/n* (including *M/M/1*): *the average service time must be identical for all chains in a node*.

BCMP–networks can be evaluated with the multi-dimensional convolution algorithm and the multidimensional *MVA* algorithm.

Mixed queueing networks (open & closed) are calculated by first calculating the traffic load in each node from the open chains. This traffic must be carried to enter statistical equilibrium. The capacity of the nodes are reduced by this traffic, and the closed queueing network is calculated by the reduced capacity. So the main problem is to calculate closed networks. For this we have more algorithms among which the most important ones are *convolution algorithm* and the *MVA* (*M*ean *V*alue *A*lgorithm) algorithm.

14.6 Multidimensional queueing networks

In this section we consider queueing networks with more than one type of customers. Customers of same type belong to a specific *class* or *chain*. In Chap. 10 we considered loss systems with several types of customers (services) and noticed that the product form was maintained and that the convolution algorithm could be applied.

14.6.1 *M/M/1* single server queueing system

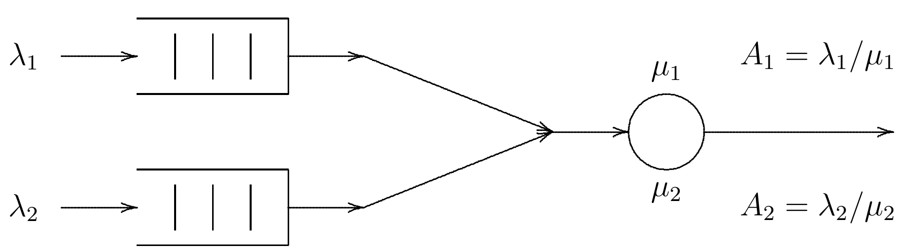


Figure 14.7: *An $M/M/1$–queueing system with two types (chains) of customers.*

Fig. 14.7 illustrates a single server queueing system with $N = 2$ types of customers (*chains*). Customers arrive to the system according to a Poisson arrival process with intensity λ_j $(j = 1, 2)$. State (i, j) is defined as a state with i type 1 customers and j type 2 customers. The service intensity $\mu_{i,j}$ in state (i, j) can be chosen such that it is state dependent, for example:

$$\mu_{ij} = \frac{i}{i+j} \cdot \mu_1 + \frac{j}{i+j} \cdot \mu_2 \,.$$

The service rates can be interpreted in several ways corresponding to the symmetric single server queueing system. One interpretation corresponds to processor sharing, i.e. all $(i + j)$ customers share the server and the capacity of the server is constant. The state dependency is due to the difference in service rates between the two types of customers; i.e. the number of customers terminated per time unit depends on the types of customers currently being served.

Another interpretation corresponds to an *M/M/1* system. If we assume $\mu_1 = \mu_2$, then it

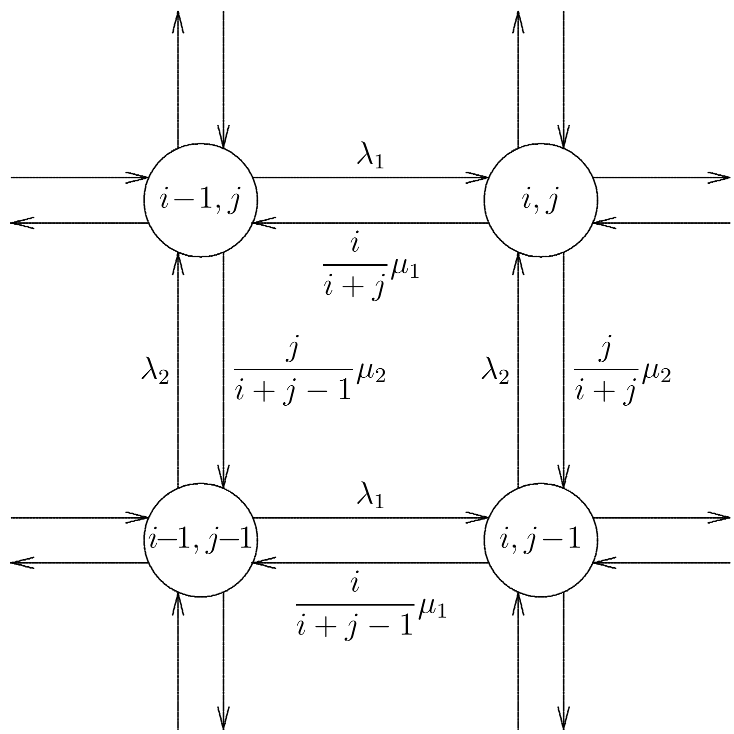


Figure 14.8: *State transition diagram for a multi–dimensional $M/M/1$–system with processor sharing.*

can be shown that the customer being served is with probability $i/(i+j)$ type 1, and with probability $j/(i+j)$ type 2. This is independent of the service discipline.

Part of the state transition diagram is given by Fig. 14.8. The diagram is reversible, since the flow clockwise equals the flow counter-clockwise. Hence, there is *local balance* and all state probabilities can be expressed by $p(0,0)$:

$$p(i,j) = \frac{A_1^i}{i!} \cdot \frac{A_2^j}{j!} \cdot (i+j)! \cdot p(0,0)\,. \tag{14.13}$$

By normalisation we get $p(0,0)$:

$$\sum_{i=0}^{\infty} \sum_{j=0}^{\infty} p(i,j) = 1\,.$$

In comparison with the multidimensional Erlang–B formula we now have the additional factor $(i+j)!$. The product form between chains (inside a node) is lost, but the product form between nodes will still be maintained.

If there are N different types of customers (chains) the state probabilities for a *single* node becomes:

$$p(\underline{i}) = p(i_1, i_2, \ldots, i_N) = \left\{ \prod_{j=1}^{N} A_j^{i_j} \right\} \cdot \frac{\left\{ \sum_{j=1}^{N} i_j \right\}!}{\left\{ \prod_{j=1}^{N} i_j! \right\}} \cdot p(\underline{0}) \,. \tag{14.14}$$

This can be expressed by the polynomial distribution (4.37):

$$p(\underline{i}) = \left\{ \prod_{j=1}^{N} A_j^{i_j} \right\} \cdot \binom{i_1 + i_2 + \cdots + i_N}{i_1, i_2, \ldots, i_N} \cdot p(0) \,. \tag{14.15}$$

For an unlimited number of queueing positions the state probabilities of the total number of customers are:

$$p(j) = p\{i_1 + i_2 + \cdots + i_N = j\} \,.$$

If $\mu_i = \mu$, then the system is identical to an *M/M/1* System with arrival rate $\lambda = \sum_i \lambda_i$:

$$\begin{aligned} p(j) &= (A_1 + A_2 + \cdots + A_N)^j \cdot p(0) \\ &= A^j \cdot (1 - A) \,. \end{aligned}$$

The Binomial expansion is used to obtain this result. The state transition diagram in Fig. 14.8 can also be interpreted as the state transition diagram of an *M/G/1-LCFS-PR* (preemptive resume) system. It is obvious that *M/G/1-LCFS-PR* is reversible because the process follows exactly the same path in the state transition diagram away from state zero as back to state zero.

The state transition diagram can be shown to be insensitive to the service time distribution so that it is valid for *M/G/1 queueing system*. Fig. 14.8 corresponds to a state transition diagram for a single server queueing system with hyper–exponentially distributed service times (cf. (10.7)), e.g. *M/*H_2*/1–LCFS–PR* or *PS*.

Notice, that for *M/M/1* (*FCFS, LCFS, SIRO*) it is necessary to assume that all customers have the same mean service time, which must be exponentially distributed. Other ways, the customer being served will not be a random customer among the $(i + j)$ customers in the system.

In conclusion, single server queueing systems with more types of customers will only have product form when the node is a symmetric queueing system: *M/G/1–PS*, *M/G/1–LCFS–PR*, or *M/M/1* with same service time for all customers.

14.6.2 $M/M/n$ queueing system

We may also carry through the above for a system with n servers. For $(i+j) \leq n$ we get the same relative state probabilities as for the multi–dimensional Erlang-B formula. For $(i+j) > n$ we only get a simple interpretation when $\mu_i = \mu$, i.e. when all types (chains) of customers have the same mean holding time. We then find the state probabilities given in (10.9), and the system has product form. $M/M/\infty$ may be considered as a special case of $M/M/n$ and has already been dealt with in connection with loss systems (Chap. 12).

14.7 Closed queueing networks with multiple chains

Dealing with queueing networks having multiple chains is analogous to the case with a single chain. The only difference is that the classical formulæ and algorithms are replaced by the corresponding multi-dimensional formulæ.

14.7.1 Convolution algorithm

The algorithm is essentially the same as in the single chain case:

- Step 1. Consider each chain as if it is alone in the network. Find the relative load at each node by solving the flow balance equation (14.5). At an arbitrary *reference node* we assume that the load is equal to one. For each chain we may choose a different node as reference node. For chain j in node k the relative arrival intensity λ_k^j is obtained from (we use the upper index to denote the chain):

  $$\lambda_k^j = \sum_{i=1}^{K} p_{ik}^j \cdot \lambda_i^j, \qquad j = 1, \ldots, N\,, \tag{14.16}$$

  where:

  K = number of nodes,

  N = number of chains,

  p_{ik}^j = the probability that a customer of chain j jumps from node i to node k.

  We choose an arbitrary node as reference node, e.g. node 1, i.e. $\lambda_1^j = 1$. The relative load at node k due to customers of chain j is then:

  $$\alpha_k^j = \lambda_k^j \cdot s_k^j$$

  where s_k^j = is the mean service time at node k for customers of chain j. Note, j is an index *not* a power.

- *Step 2.* Based on the relative loads found in step 1, we obtain the multi-dimensional state probabilities for each node. Each node is considered in isolation and we truncate the state space according to the number of customers in each chain. For example for node k $(1 \leq k \leq K)$:

$$\underline{p}_k = p_k(i_1, i_2, \ldots, i_N)\,, \quad 0 \leq i_j \leq S_j\,, \quad j = 1, 2, \ldots N\,,$$

  where S_j is the number of customers in chain j.

- *Step 3.* In order to find the state probabilities of the total network, the state probabilities of each node are convolved together similar to the single chain case. The only difference is that the convolution is multi-dimensional. When we perform the last convolution we may obtain the performance measures of the last node. Again, by changing the order of nodes, we can obtain the performance measures of all nodes.

The total number of states increases rapidly. For example, if chain j has S_j customers, then the total number of states in each node becomes:

$$\prod_{j=1}^{N} (S_j + 1)\,.$$

The number of ways N chains with S_j customers in chain j can be distributed in a queueing network with K nodes is:

$$\mathcal{C} = \prod_{j=1}^{N} C(S_j, k_j) \tag{14.17}$$

where k_j $(1 \leq k_j < k)$ is the number of nodes visited by chain j and:

$$C(S_j, k_j) = \left(\begin{array}{c} S_j + k_j - 1 \\ k_j - 1 \end{array} \right) = \left(\begin{array}{c} S_j + k_j - 1 \\ S_j \end{array} \right). \tag{14.18}$$

The algorithm is best illustrated with an example.

Example 14.7.1: Palm's machine-repair model with two types of customers
As seen in Example 14.4.1, this system can be modelled as a queueing network with two nodes. Node 1 corresponds to the terminals (machines) while node 2 is the CPU (repair man). Node 2 is a single server system whereas node 1 is modelled as a Infinite Server system. The number of customers in the chains are $(S_1 = 2,\ S_2 = 3)$ and the mean service time in node k is s_k^j. The relative load of chain 1 is denoted by α_1 in node 1 and by α_2 in node 2. Similarly, the load af chain 2 is denoted by β_1, respectively β_2. Applying the convolution algorithm yields:

- *Step 1.*

Chain 1:	$S_1 = 2$ customers	
Relative load:	$\alpha_1 = \lambda_1 \cdot s_1^1\,,$	$\alpha_2 = \lambda_1 \cdot s_2^1\ .$
Chain 2:	$S_2 = 3$ customers	
Relative load:	$\beta_1 = \lambda_2 \cdot s_1^2\ ,$	$\beta_2 = \lambda_2 \cdot s_2^2\ .$

- *Step 2.*
  For node 1 (*IS*) the relative state probabilities are (cf. 10.9):

$$
\begin{aligned}
q_1(0,0) &= 1 & q_1(0,2) &= \frac{\beta_1^2}{2}\\
q_1(1,0) &= \alpha_1 & q_1(1,2) &= \frac{\alpha_1 \cdot \beta_1^2}{2}\\
q_1(2,0) &= \frac{\alpha_1^2}{2} & q_1(2,2) &= \frac{\alpha_1^2 \cdot \beta_1^2}{4}\\
q_1(0,1) &= \beta_1 & q_1(0,3) &= \frac{\beta_1^3}{6}\\
q_1(1,1) &= \alpha_1 \cdot \beta_1 & q_1(1,3) &= \frac{\alpha_1 \cdot \beta_1^3}{6}\\
q_1(2,1) &= \frac{\alpha_1^2 \cdot \beta_1}{2} & q_1(2,3) &= \frac{\alpha_1^2 \cdot \beta_1^3}{12}
\end{aligned}
$$

  For node 2 (single server) (cf. 14.14) we get:

$$
\begin{aligned}
q_2(0,0) &= 1 & q_2(0,2) &= \beta_2^2\\
q_2(1,0) &= \alpha_2 & q_2(1,2) &= 3 \cdot \alpha_2 \cdot \beta_2^2\\
q_2(2,0) &= \alpha_2^2 & q_2(2,2) &= 6 \cdot \alpha_2^2 \cdot \beta_2^2\\
q_2(0,1) &= \beta_2 & q_2(0,3) &= \beta_2^3\\
q_2(1,1) &= 2 \cdot \alpha_2 \cdot \beta_2 & q_2(1,3) &= 4 \cdot \alpha_2 \cdot \beta_2^3\\
q_2(2,1) &= 3 \cdot \alpha_2^2 \cdot \beta_2 & q_2(2,3) &= 10 \cdot \alpha_2^2 \cdot \beta_2^3
\end{aligned}
$$

- *Step 3.*
  Next we convolve the two nodes. We know that the total number of customers are $(2,3)$, i.e. we are only interested in state $(2,3)$:

$$
\begin{aligned}
q_{12}(2,3) \quad = \quad & q_1(0,0) \cdot q_2(2,3) + q_1(1,0) \cdot q_2(1,3)\\
& + q_1(2,0) \cdot q_2(0,3) + q_1(0,1) \cdot q_2(2,2)\\
& + q_1(1,1) \cdot q_2(1,2) + q_1(2,1) \cdot q_2(0,2)\\
& + q_1(0,2) \cdot q_2(2,1) + q_1(1,2) \cdot q_2(1,1)\\
& + q_1(2,2) \cdot q_2(0,1) + q_1(0,3) \cdot q_2(2,0)\\
& + q_1(1,3) \cdot q_2(1,0) + q_1(2,3) \cdot q_2(0,0)
\end{aligned}
$$

Using the actual values yields:

$$
\begin{aligned}
q_{12}(2,3) = &+ \quad 1 \cdot 10 \cdot \alpha_2^2 \cdot \beta_2^3 && + \quad \alpha_1 \cdot 4 \cdot \alpha_2 \cdot \beta_2^3 \\
&+ \quad \frac{\alpha_1^2}{2} \cdot \beta_2^3 && + \quad \beta_1 \cdot 6 \cdot \alpha_2^2 \cdot \beta_2^2 \\
&+ \quad \alpha_1 \cdot \beta_1 \cdot 3 \cdot \alpha_2 \cdot \beta_2^2 && + \quad \frac{\alpha_1^2 \cdot \beta_1}{2} \cdot \beta_2^2 \\
&+ \quad \frac{\beta_1^2}{2} \cdot 3 \cdot \alpha_2^2 \cdot \beta_2 && + \quad \frac{\alpha_1 \cdot \beta_1^2}{2} \cdot 2 \cdot \alpha_2 \cdot \beta_2 \\
&+ \quad \frac{\alpha_1^2 \cdot \beta_1^2}{4} \cdot \beta_2 && + \quad \frac{\beta_1^3}{6} \cdot \alpha_2^2 \\
&+ \quad \frac{\alpha_1 \cdot \beta_1^3}{6} \cdot \alpha_2 && + \quad \frac{\alpha_1^2 \cdot \beta_1^3}{12} \cdot 1
\end{aligned}
$$

Note that α_1 and α_2 together (chain 1) always appears in the second power whereas β_1 and β_2 (chain 2) appears in the third power corresponding to the number of customers in each chain. Because of this, only the relative loads are relevant, and the absolute probabilities are obtain by normalisation by dividing all the terms by $q_{12}(2,3)$. The detailed state probabilities are now easy to obtain. Only in the state with the term $(\alpha_1^2 \cdot \beta_1^3)/12$ is the CPU (repair man) idle. If the two types of customers are identical the model simplifies to Palm's machine/repair model with 5 terminals. In this case we have:

$$E_{1,5}(x) = \frac{\frac{1}{12} \cdot \alpha_1^2 \cdot \beta_1^3}{q_{12}(2,3)} \,.$$

Choosing $\alpha_1 = \beta_1 = \alpha$ and $\alpha_2 = \beta_2 = 1$, yields:

$$
\begin{aligned}
\frac{\frac{1}{12} \cdot \alpha_1^2 \cdot \beta_1^3}{q_{12}(2,3)} &= \frac{\alpha^5/12}{10 + 4\alpha + \frac{1}{2}\alpha^2 + 6\alpha + 3\alpha^2 + \frac{1}{2}\alpha^3 + \frac{3}{2}\alpha^2 + \alpha^3 + \frac{1}{4}\alpha^4 + \frac{1}{6}\alpha^3 + \frac{1}{6}\alpha^4 + \frac{1}{12}\alpha^5} \\
&= \frac{\dfrac{\alpha^5}{5!}}{1 + \alpha + \dfrac{\alpha^2}{2} + \dfrac{\alpha^3}{3!} + \dfrac{\alpha^4}{4!} + \dfrac{\alpha^5}{5!}} \,,
\end{aligned}
$$

i.e. the Erlang–B formula as expected. □

14.8 Other algorithms for queueing networks

Also the *MVA*—–algorithm is applicable to queueing networks with more chains, but it will not be described here. During the last decade several algorithms have been published. An overview can be found in (Conway & Georganas, 1989 [15]). In general, exact algorithms are not applicable for bigger networks. Therefore, many approximative algorithms have been developed to deal with queueing networks of realistic size.

14.9 Complexity

Queueing networks has the same complexity as circuit switched networks with direct routing (Sec. 11.5 and Tab. 11.2). The state space of the network shown in Tab. 14.3 has the following number of states for every node:

$$\prod_{i=0}^{N}(S_i+1)\,. \tag{14.19}$$

The worst case is when every chain consists of one customer. Then the number of states becomes 2^S where S is the number of customers.

Chain	Node				Population
	1	2	$\cdots$	K	Size
1	α_{11}	α_{21}	$\cdots$	α_{K1}	S_1
2	α_{12}	α_{22}	$\cdots$	α_{K2}	S_2
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
N	α_{1N}	α_{2N}	$\cdots$	α_{KN}	S_N

Table 14.3: *The parameters of a queueing network with N chains, K nodes and $\sum_i S_i$ customers. The parameter α_{jk} denotes the load from customers of chain j in node k (cf. Tab. 11.2).*

14.10 Optimal capacity allocation

We now consider a data transmission system with K nodes, which are independent single server queueing systems *M/M/1* (Erlang's delay system with one server). The arrival process to node k is a Poisson process with intensity λ_k messages (customers) per time unit, and the message size is exponentially distributed with mean value $1/\mu_k$ [bits]. The capacity of node k is φ_k [bits per time unit].

We introduce the following linear restriction on the total capacity:

$$F=\sum_{k=1}^{K}\varphi_k\,. \tag{14.20}$$

For every allocation of capacity which satisfies (14.20), we have the following mean sojourn time (call average) ((12.34) and combination in parallel):

$$T = \sum_{k=1}^{K} \frac{\lambda_k}{\lambda} \cdot \frac{1}{\mu_k \cdot \varphi_k - \lambda_k}\,, \tag{14.21}$$

where:

$$\lambda = \sum_{k=1}^{K} \lambda_k\,. \tag{14.22}$$

By defining:

$$A = \frac{\lambda}{\mu \cdot F}\,, \tag{14.23}$$

$$\frac{1}{\mu} = \sum_{k=1}^{K} \frac{\lambda_k}{\lambda} \cdot \frac{1}{\mu_k}\,, \tag{14.24}$$

we get Kleinrock's law for optimal capacity allocation (Kleinrock, 1964 [64]):

Theorem 14.2 Kleinrock's square root law: *The optimal allocation of capacity which minimises T (and thus the total number of messages in all nodes) is:*

$$\varphi_k = \frac{\lambda_k}{\mu_k} + F \cdot (1-A) \frac{\sqrt{\lambda_k/\mu_k}}{\sum_{i=1}^{K} \sqrt{\lambda_i/\mu_i}}\,, \tag{14.25}$$

under the condition that:

$$F > \sum_{k=1}^{K} \frac{\lambda_k}{\mu_k}\,. \tag{14.26}$$

This can be shown by introducing *Lagrange multiplier* ϑ and consider:

$$G = T - \vartheta \left\{ \sum_{k=1}^{K} \varphi_k - F \right\}\,. \tag{14.27}$$

Minimum of G is obtained by choosing φ_k as given in (14.25).

With this optimal allocation we find:

$$T = \frac{\left\{ \sum_{k=1}^{K} \sqrt{\lambda_k/\mu_k} \right\}^2}{\lambda \cdot F \cdot (1-A)}\,. \tag{14.28}$$

This optimal allocation corresponds to that all nodes first are allocated the necessary minimum capacity λ_i/μ_i. The remaining capacity (14.24):

$$F - \sum_{k=1}^{K} \frac{\lambda_i}{\mu_i} = F \cdot (1-A) \tag{14.29}$$

is allocated among the nodes proportional the square root of the average flow λ_k/μ_k.

If all messages have the same mean value ($\mu_k = \mu$), then we may consider different costs in the nodes under the restriction that a fixed amount is available (Kleinrock, 1964 [64]).

Chapter 15

Traffic measurements

Traffic measurements are carried out in order to obtain quantitative information about the load on a system to be able to dimension the system. By traffic measurements we understand any kind of collection of data on the traffic loading a system. The system considered may be a physical system, for instance a computer, a telephone system, or the central laboratory of a hospital. It may also be a fictitious system. The collection of data in a computer simulation model corresponds to a traffic measurements. Billing of telephone calls also corresponds to a traffic measurement where the measuring unit used is an amount of money.

The extension and type of measurements and the parameters (traffic characteristics) measured must in each case be chosen in agreement with the demands, and in such a way that a minimum of technical and administrative efforts result in a maximum of information and benefit. According to the nature of traffic a measurement during a limited time interval corresponds to a registration of a certain realisation of the traffic process. A measurement is thus a sample of one or more random variables. By repeating the measurement we usually obtain a different value, and in general we are only able to state that the unknown parameter (the population parameter, for example the mean value of the carried traffic) with a certain probability is within a certain interval, the confidence interval. The full information is equal to the distribution function of the parameter. For practical purposes it is in general sufficient to know the mean value and the variance, i.e. the distribution itself is of minor importance.

In this chapter we shall focus upon the statistical foundation for estimating the reliability of a measurement, and only to a limited extent consider the technical background. As mentioned above the theory is also applicable to stochastic computer simulation models.

15.1 Measuring principles and methods

The technical possibilities for measuring are decisive for what is measured and how the measurements are carried out. The first program controlled measuring equipment was developed at the Technical University of Denmark, and described in (Andersen & Hansen & Iversen, 1971 [2]). Any traffic measurement upon a traffic process, which is discrete in state and continuous in time can in principle be implemented by combining two fundamental operations:

1. *Number of events:* this may for example be the number of errors, number of call attempts, number of errors in a program, number of jobs to a computing centre, etc. (cf. number representation, Sec. 5.1.1).

2. *Time intervals:* examples are conversation times, execution times of jobs in a computer, waiting times, etc. (cf. interval representation, Sec. 5.1.2).

By combining these two operations we may obtain any characteristic of a traffic process. The most important characteristic is the (carried) traffic volume, i.e. the summation of all (number) holding times (interval) within a given measuring period.

From a functional point of view all traffic measuring methods can be divided into the following two classes:

1. Continuous measuring methods.

2. Discrete measuring methods.

15.1.1 Continuous measurements

In this case the measuring point is active and it activates the measuring equipment at the instant of the event. Even if the measuring method is continuous the result may be discrete.

Example 15.1.1: Measuring equipment: continuous time
Examples of equipment operating according to the continuous principle are:

(a) Electro-mechanical counters which are increased by one at the instant of an event.

(b) Recording x–y plotters connected to a point which is active during a connection.

(c) Ampère-hour meters, which integrate the power consumption during a measuring period. When applied for traffic volume measurements in old electro-mechanical exchanges every trunk is connected through a resistor of 9,6 kΩ, which during occupation is connected between –48 volts and ground and thus consumes 5 mA.

(d) Water meters which measure the water consumption of a household.

□

15.1.2 Discrete measurements

In this case the measuring point is passive, and the measuring equipment must itself test (poll) whether there have been changes at the measuring points (normally binary, on-off). This method is called *the scanning method* and the scanning is usually done at regular instants (constant = deterministic time intervals). All events which have taken place between two consecutive scanning instants are from a time point of view referred to the latter scanning instant, and are considered as taking place at this instant.

Example 15.1.2: Measuring equipment: discrete time
Examples of equipment operating according to the discrete time principle are:

(a) Call charging according to the *Karlsson principle*, where charging pulses are issued at regular time instants (distance depends upon the cost per time unit) to the meter of the subscriber, who has initiated the call. Each unit (step) corresponds to a certain amount of money. If we measure the duration of a call by its cost, then we observe a discrete distribution ($0, 1, 2, \ldots$ units). The method is named after S.A. Karlsson from Finland (Karlsson, 1937 [56]). In comparison with most other methods it requires a minimum of administration.

(b) The carried traffic on a trunk group of an electro-mechanical exchange is in practice measured according to the scanning principle. During one hour we observe the number of busy trunks 100 times (every 36 seconds) and add these numbers on a mechanical counter, which thus indicate the average carried traffic with two decimals. By also counting the number of calls we can estimate the average holding time.

(c) The scanning principle is particularly appropriate for implementation in digital systems. For example, the processor controlled equipment developed at *DTU*, Technical University of Denmark, in 1969 was able to test 1024 measuring points (e.g. relays in an electro-mechanical exchange, trunks or channels) within 5 milliseconds. The states of each measuring point (idle/busy or off/on) at the two latest scannings are stored in the computer memory, and by comparing the readings we are able to detect changes of state. A change of state $0 \rightarrow 1$ corresponds to start of an occupation and $1 \rightarrow 0$ corresponds to termination of an occupation *(last–look principle)*. The scannings are controlled by a clock. Therefore we may monitor every channel during time and measure time intervals and thus observe time distributions. Whereas the classical equipment (erlang-meters) mentioned above observes the traffic process in the state space (*vertical*, number representation), then the program controlled equipment observes the traffic process in time space (*horizontal*, interval representation), in discrete time. The amount of information is almost independent of the scanning interval as only state changes are stored (the time of a scanning is measured in an integral number of scanning intervals).

□

Measuring methods have had decisive influence upon the way of thinking and the way of formulating and analysing the statistical problems. The classical equipment operating in state space has implied that the statistical analyses have been based upon state probabilities, i.e. basically birth and death processes. From a mathematically point of view these models have been rather complex *(vertical measurements)*.

The following derivations are in comparison very elementary and even more general, and they are inspired by the operation in time space of the program controlled equipment. (Iversen, 1976 [36]) *(horizontal measurements).*

15.2 Theory of sampling

Let us assume we have a sample of n *IID* (Independent and Identically Distributed) observations $\{X_1, X_2, \ldots, X_n\}$ of a random variable with unknown finite mean value m_1 and finite variance σ^2 (population parameters).

The mean value and variance of the *sample* are defined as follows:

$$\bar{X} = \frac{1}{n} \cdot \sum_{i=1}^{n} X_i \tag{15.1}$$

$$s^2 = \frac{1}{n-1} \left\{ \sum_{i=1}^{n} X_i^2 - n \cdot \bar{X}^2 \right\} \tag{15.2}$$

Both $\bar{X}$ and s^2 are functions of a random variable and therefore also random variables, defined by a distribution we call the *sample distribution*. $\bar{X}$ is a central estimator of the unknown population mean value m_1, i.e.:

$$E\{\bar{X}\} = m_1 \tag{15.3}$$

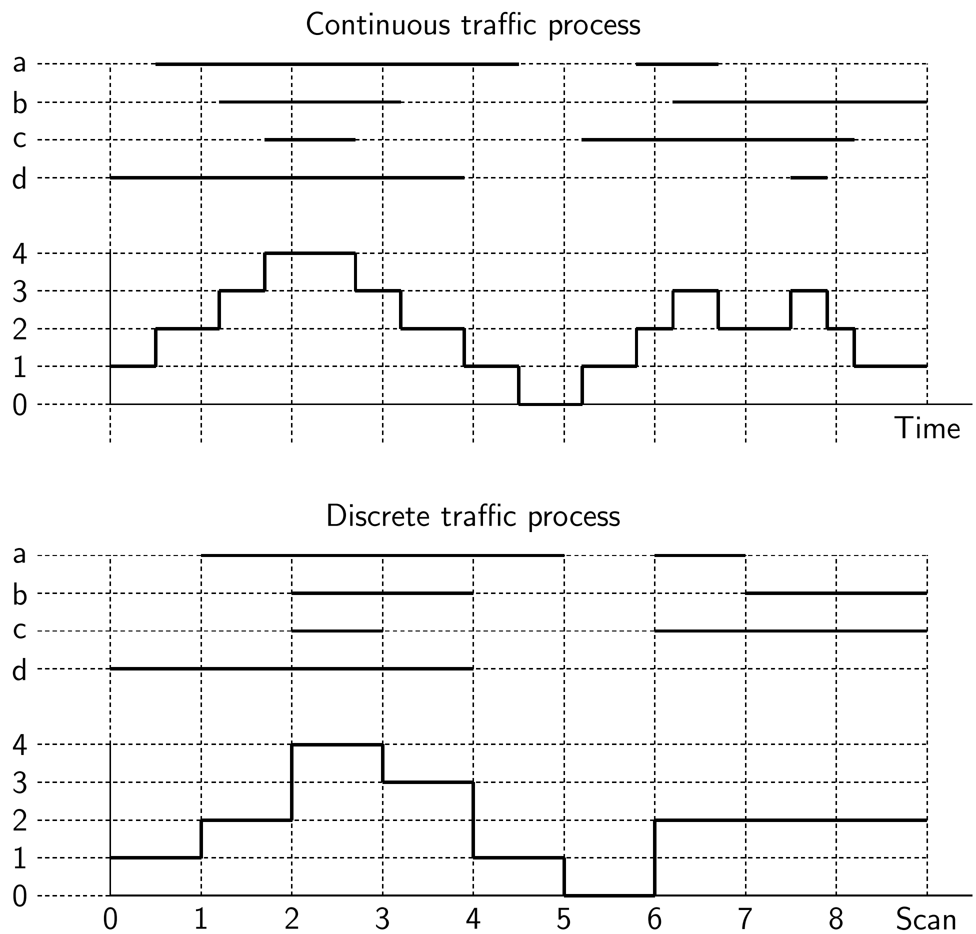
Furthermore, s^2/n is a central estimator of the unknown variance of the sample mean $\bar{X}$, i.e.:

$$\bar{\sigma^2}\{\bar{X}\} = s^2/n. \tag{15.4}$$

We describe the accuracy of an estimate of a sample parameter by means of a confidence interval, which with a given probability specifies how the estimate is placed relatively to the unknown theoretical value. In our case the confidence interval of the mean value becomes:

$$\bar{X} \pm t_{n-1,1-\alpha/2} \cdot \sqrt{\frac{s^2}{n}} \tag{15.5}$$

where $t_{n-1,1-\alpha/2}$ is the upper $(1-\alpha/2)$ percentile of the t-distribution with $n-1$ degrees of freedom. The probability that the confidence interval includes the unknown theoretical mean value is equal to $(1-\alpha)$ and is called the level of confidence. Some values of the t-distribution are given in Table 15.1. When n becomes large, then the t-distribution converges to the Normal distribution, and we may use the percentile of this distribution. The assumption of independence are fulfilled for measurements taken on different days, but for example not for successive measurements by the scanning method within a limited time interval, because the number of busy channels at a given instant will be correlated with the number of busy circuits in the previous and the next scanning. In the following sections we calculate the mean value and the variance of traffic measurements during for example one hour. This aggregated value



Scanning method

a	0	→	1		1		1		1	→	0	→	1	→	0		0		0
b	0		0	→	1		1	→	0		0		0	→	1		1		1
c	0		0	→	1	→	0		0		0	→	1		1		1	→	0
d	1		1		1		1	→	0		0		0		0		0		0
Σ	1		2		4		3		1		0		2		2		2		1

Figure 15.1: *Observation of a traffic process by a continuous measuring method and by the scanning method with regular scanning intervals. By the scanning method it is sufficient to observe the changes of state.*

n	$\alpha = 10\%$	$\alpha = 5\%$	$\alpha = 1\%$
1	6.314	12.706	63.657
2	2.920	4.303	9.925
5	2.015	2.571	4.032
10	1.812	2.228	3.169
20	1.725	2.086	2.845
40	1.684	2.021	2.704
∞	1.645	1.960	2.576

Table 15.1: *Percentiles of the t-distribution with n degrees of freedom. A specific value of α corresponds to a probability mass $\alpha/2$ in both tails of the t-distribution. When n is large, then we may use the percentiles of the Normal distribution.*

for a given day may then be used as a single observation in the formulæ above, where the number of observations typically will be the number of days, we measure.

Example 15.2.1: Confidence interval for call congestion
On a trunk group of 30 trunks (channels) we observe the outcome of 500 call attempts. This measurement is repeated 11 times, and we find the following call congestion values (in percentage):

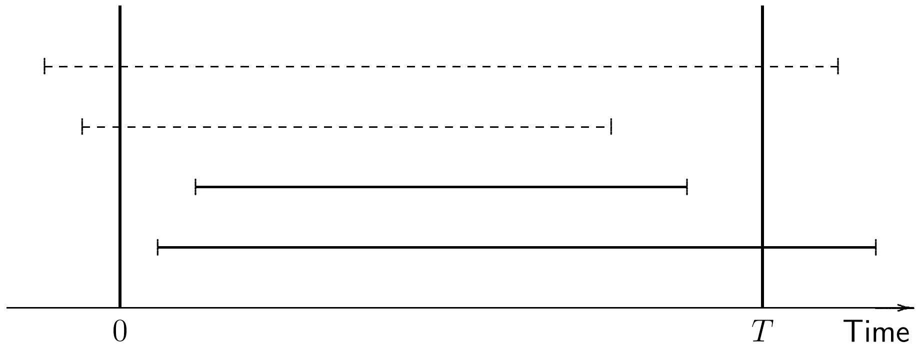
$$\{9.2,\ 3.6,\ 3.6,\ 2.0,\ 7.4,\ 2.2,\ 5.2,\ 5.4,\ 3.4,\ 2.0,\ 1.4\}$$

The total sum of the observations is 45.4 and the total of the squares of the observations is 247.88. We find (15.1) $\bar{X} = 4.1273$ % and (15.2) $s^2 = 6.0502$ $(\%)^2$. At 95%–level the confidence interval becomes by using the t-values in Table 15.1: (2.47–5.78). It is noticed that the observations are obtained by simulating a *PCT–I* traffic of 25 erlang, which is offered to 30 channels. According to the Erlang B–formula the theoretical blocking probability is 5.2603 %. This value is within the confidence interval. If we want to reduce the confidence interval with a factor 10, then we have to do 100 times as many observations (cf. formula 15.5), i.e. 50,000 per measurements (sub-run). We carry out this simulation and observe a call congestion equal to 5.245 % and a confidence interval (5.093 – 5.398). □

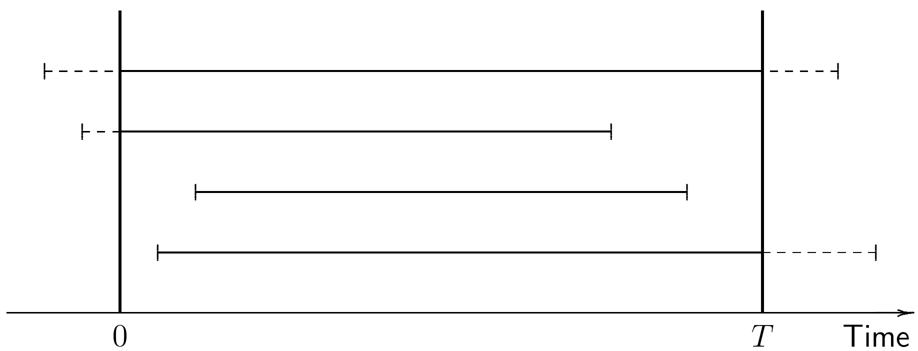
15.3 Continuous measurements in an unlimited period

Measuring of time intervals by continuous measuring methods with no truncation of the measuring period are easy to deal with by the theory of sampling described in Sec. 15.2 above.

For a *traffic volume* or a *traffic intensity* we can apply the formulæ (3.46) and (3.48) for a stochastic sum. They are quite general, the only restriction being stochastic independence



a: Unlimited measuring period



b: Limited measuring period

Figure 15.2: *When analysing traffic measurements we distinguish between two cases: (a) Measurements in an unlimited time period. All calls initiated during the measuring period contributes with their total duration. (b) Measurements in a limited measuring period. All calls contribute with the portion of their holding times which are located inside the measuring period. In the figure the sections of the holding times contributing to the measurements are shown with full lines.*

between X and N. In practice this means that the systems must be without congestion. In general we will have a few percentages of congestion and may still as worst case assume independence. By far the most important case is a Poisson arrival process with intensity λ. We then get a stochastic sum (Sec. 3.3). For the Poisson arrival process we have when we consider a time interval T:

$$m_{1,n} = \sigma_n^2 = \lambda \cdot T$$

and therefore we find:

$$\begin{aligned} m_{1,s} &= \lambda T \cdot m_{1,t} \\ \sigma_s^2 &= \lambda T \left\{ m_{1,t}^2 + \sigma_t^2 \right\} \\ &= \lambda T \cdot m_{2,t} \;=\; \lambda T \cdot m_{1,t}^2 \cdot \varepsilon_t \,, \end{aligned} \tag{15.6}$$

where $m_{2,t}$ is the second (non-central) moment of the holding time distribution, and ε_t is Palm's form factor of the same distribution:

$$\varepsilon = \frac{m_{2,t}}{m_{1,t}^2} = 1 + \frac{\sigma_t^2}{m_{1,t}^2} \tag{15.7}$$

The distribution of S_T will in this case be a *compound Poisson distribution* (Feller, 1950 [27]).

The formulæ correspond to a traffic volume (e.g. erlang-hours). For most applications as dimensioning we are interested in the average number of occupied channels, i.e. the traffic intensity (rate) = traffic per time unit ($m_{1,t} = 1$, $\lambda = A$), when we choose the mean holding time as time unit:

$$m_{1,i} = A \tag{15.8}$$

$$\sigma_i^2 = \frac{A}{T} \cdot \varepsilon_t \tag{15.9}$$

These formulæ are thus valid for arbitrary holding time distributions. The formulæ (15.8) and (15.9) are originally derived by C. Palm (1941 [78]). In (Rabe, 1949 [86]) the formulæ for the special cases $\varepsilon_t = 1$ (constant holding time) and $\varepsilon_t = 2$ (exponentially distributed holding times) were published.

The above formulæ are valid for all calls arriving *inside* the interval T when measuring the total duration of all holding times regardless for how long time the stay (Fig. 15.2 a).

Example 15.3.1: Accuracy of a measurement
We notice that we always obtain the correct mean value of the traffic intensity (15.8). The variance, however, is proportional to the form factor ε_t. For some common cases of holding time distributions we get the following variance of the traffic intensity measured:

Constant:	$\sigma_i^2 = \frac{A}{T}\,,$
Exponential distribution:	$\sigma_i^2 = \frac{A}{T} \cdot 2\,,$
Observed (Fig. 4.3):	$\sigma_i^2 = \frac{A}{T} \cdot 3.83\,.$

Observing telephone traffic, we often find that ε_t is significant larger than the value 2 (exponential distribution), which is presumed to be valid in many classical teletraffic models (Fig. 4.3). Therefore, the accuracy of a measurement is lower than given in many tables. This, however, is compensated by the assumption that the systems are non–blocking. In a system with blocking the variance becomes smaller due to negative correlation between holding times and number of calls. □

Example 15.3.2: Relative accuracy of a measurement
The relative accuracy of a measurement is given by the ratio:

$$S = \frac{\sigma_i}{m_{1,i}} = \left\{\frac{\varepsilon_t}{AT}\right\}^{1/2} = \textit{ variation coefficient.}$$

From this we notice that if $\varepsilon_t = 4$, then we have to measure twice as long a period to obtain the same reliability of a measurement as for the case of exponentially distributed holding times. □

For a given time period we notice that the accuracy of the traffic intensity when measuring a small trunk group is much larger than when measuring a large trunk group, because the accuracy only depends on the traffic intensity A. When dimensioning a small trunk group, an error in the estimation of the traffic of 10 % has much less influence than the same percentage error on a large trunk group (Sec. 7.6.1). Therefore we measure the same time period on all trunk groups. In Fig. 15.5 the relative accuracy for a continuous measurement is given by the straight line $h = 0$.

15.4 Scanning method in an unlimited time period

In this section we only consider regular (constant) scanning intervals. The scanning principle is for example applied to traffic measurements, call charging, numerical simulations, and processor control. By the scanning method we observe a discrete time distribution for the holding time which in real time usually is continuous.

In practice we usually choose a constant distance h between scanning instants, and we find the following relation between the observed time interval and the real time interval (fig. 15.3):

Observed time	Real time
$0\ h$	$0\ h - 1\ h$
$1\ h$	$0\ h - 2\ h$
$2\ h$	$1\ h - 3\ h$
$3\ h$	$2\ h - 4\ h$
…	…

We notice that there is overlap between the continuous time intervals, so that the discrete distribution cannot be obtained by a simple integration of the continuous time interval over a fixed interval of length h. If the real holding times have a distribution function $F(t)$, then

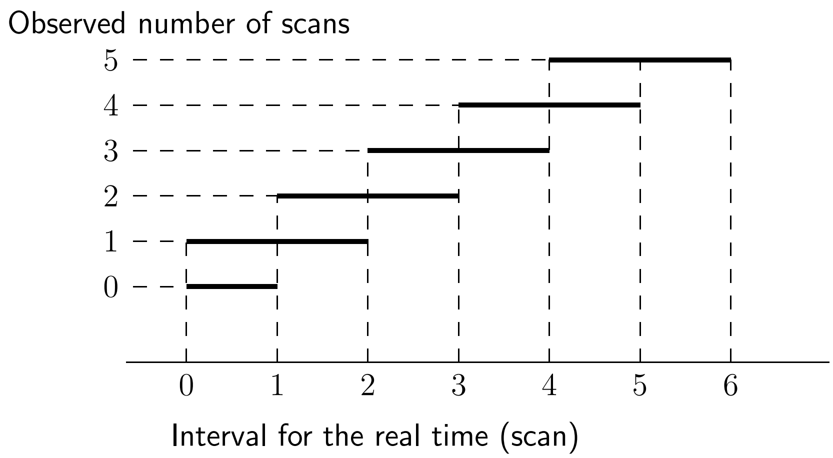


Figure 15.3: *By the scanning method a continuous time interval is transformed into a discrete time interval. The transformation is not unique (cf. Sec. 15.4).*

it can be shown that we will observe the following discrete distribution (Iversen, 1976 [36]):

$$p(0) = \frac{1}{h} \int_0^h F(t)\, dt \tag{15.10}$$

$$p(k) = \frac{1}{h} \int_0^h \{F(t+kh) - F(t+(k-1)h)\}\, dt\,, \quad k = 1, 2, \ldots\,. \tag{15.11}$$

Interpretation: The arrival time of the call is assumed to be independent of the scanning process. Therefore, the density function of the time interval from the call arrival instant to the first scanning time is uniformly distributed and equal to $(1/h)$ (Sec. 6.3.3). The probability of observing zero scanning instants during the call holding time is denoted by $p(0)$ and is equal to the probability that the call terminates before the next scanning time. For at fixed value of the holding time t this probability is equal to $F(t)/h$, and to obtain the total probability we integrate over all possible values t $(0 \leq t < h)$ and get (15.10). In a similar way we derive $p(k)$ (15.11).

By partial integration it can be shown that for any distribution function $F(t)$ we will *always observe the correct mean value:*

$$h \cdot \sum_{k=0}^{\infty} k \cdot p(k) = \int_0^{\infty} t \cdot dF(t)\,. \tag{15.12}$$

When using Karlsson charging we will therefore always in the long run charge the *correct amount*.

For exponential distributed holding time intervals, $F(t) = 1 - e^{-\mu t}$, we will observe a discrete distribution, *Westerberg's distribution* (Iversen, 1976 [36]):

$$p(0) = 1 - \frac{1}{\mu\, h}\left(1-\mathrm{e}^{-\mu\, h}\right), \tag{15.13}$$

$$p(k) = \frac{1}{\mu\, h}\left(1-\mathrm{e}^{-\mu\, h}\right)^2 \cdot \mathrm{e}^{-(k-1)\mu\, h}, \quad k=1,2,\ldots \tag{15.14}$$

This distribution can be shown to have the following mean value and form factor:

$$m_1 = \frac{1}{\mu\, h}, \tag{15.15}$$

$$\varepsilon = \mu\, h \cdot \frac{\mathrm{e}^{\mu\, h}+1}{\mathrm{e}^{\mu\, h}-1} \geq 2\,. \tag{15.16}$$

The form factor ε is equal to one plus the square of the relative accuracy of the measurement. For a continuous measurement the form factor is 2. The contribution $\varepsilon - 2$ is thus due to the influence from the measuring principle.

The form factor is a measure of accuracy of the measurements. Fig. 15.4 shows how the form factor of the observed holding time for exponentially distributed holding times depends on the length of the scanning interval (15.16). By continuous measurements we get an ordinary sample. By the scanning method we get a sample of a sample so that there is uncertainty both because of the measuring method and because of the limited sample size.

Fig. 5.2 shows an example of the Westerberg distribution. It is in particular the zero class which deviates from what we would expect from a continuous exponential distribution. If we insert the form factor in the expression for σ_s^2 (15.9), then we get by choosing the mean holding time as time unit $m_{1,t} = 1/\mu = 1$ the following estimates of the traffic intensity when using the scanning method:

$$m_{1,i} = A\,,$$

$$\sigma_i^2 = \frac{A}{T}\left\{h \cdot \frac{\mathrm{e}^h+1}{\mathrm{e}^h-1}\right\}. \tag{15.17}$$

By the continuous measuring method the variance is $2A/T$. This we also get now by letting $h \rightarrow 0$.

Fig. 15.5 shows the relative accuracy of the measured traffic volume, both for a continuous measurement (15.8) & (15.9) and for the scanning method (15.17). Formula (15.17) was derived by (Palm, 1941 [78]), but became only known when it was "re-discovered" by W.S. Hayward Jr. (1952 [33]).

Example 15.4.1: Billing principles

Various principles are applied for charging (billing) of calls. In addition, the charging rate if usually varied during the 24 hours to influence the habits of the subscriber. Among the principles we may mention:

(a) Fixed amount per call. This principle is often applied in manual systems for local calls (*flat rate*).

(b) Karlsson charging. This corresponds to the measuring principle dealt with in this section because the holding time is placed at random relative to the regular charging pulses. This principle has been applied in Denmark in the crossbar exchanges.

(c) Modified Karlsson charging. We may for instance add an extra pulse at the start of the call. In digital systems in Denmark there is a fixed fee per call in addition to a fee proportional with the duration of the call.

(d) The start of the holding time is synchronised with the scanning process. This is for example applied for operator handled calls and in coin box telephones.

□

15.5 Numerical example

For a specific measurement we calculate $m_{1,i}$ and σ_i^2. The deviation of the observed traffic intensity from the theoretical correct value is approximately Normal distributed. Therefore, the unknown theoretical mean value will be within 95% of the calculated confidence intervals (cf. Sec. 15.2):

$$m_{1,i} \pm 1,96 \cdot \sigma_i \tag{15.18}$$

The variance σ_i^2 is thus decisive for the accuracy of a measurement. To study which factors are of major importance, we make numerical calculations of some examples. All formulæ may easily be calculated on a pocket calculator.

Both examples presume *PCT–I* traffic, (i.e. Poisson arrival process and exponentially distributed holding times), traffic intensity = 10 erlang, and mean holding time = 180 seconds, which is chosen as time unit.

Example a: This corresponds to a classical traffic measurement:

Measuring period = 3600 sec = 20 time units = T.
Scanning interval = 36 sec = 0.2 time units = $h = 1/\lambda_s$.
(100 observations)

Example b: In this case we only scan once per mean holding time:

Measuring period = 720 sec = 4 time units = T.
Scanning interval = 180 sec = 1 time unit = $h = 1/\lambda_s$.
(4 observations)

From Table 15.5 we can draw some general conclusions:

- By the scanning method we loose very little information as compared to a continuous measurement as long as the scanning interval is less than the mean holding time (cf. Fig. 15.4). A continuous measurement can be considered as an optimal reference for any discrete method.

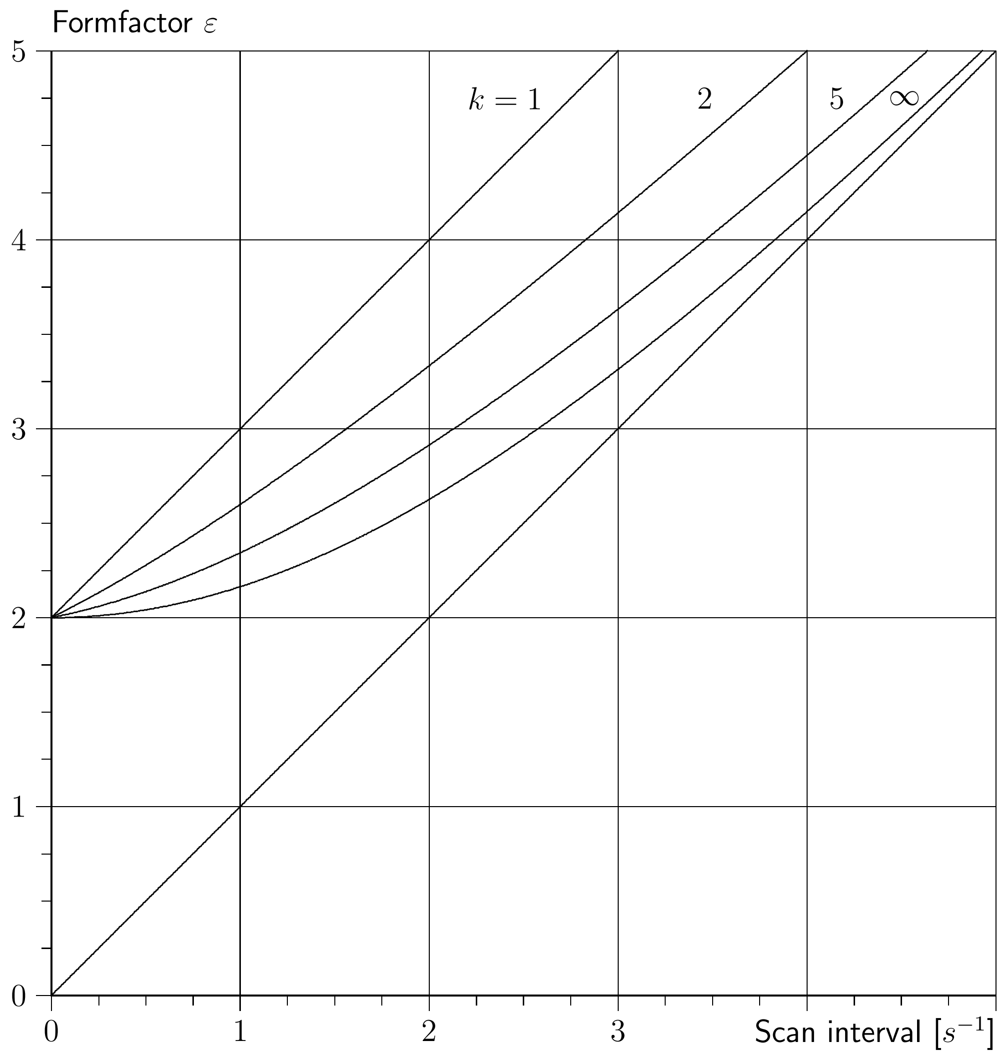


Figure 15.4: *Form factor for exponentially distributed holding times which are observed by Erlang-k distributed scanning intervals in an unlimited measuring period. The case $k = \infty$ corresponds to regular (constant) scan intervals which transform the exponential distribution into Westerberg's distribution. The case $k = 1$ corresponds to exponentially distributed scan intervals (cf. the roulette simulation method). The case $h = 0$ corresponds to a continuous measurement. We notice that by regular scan intervals we loose almost no information if the scan interval is smaller than the mean holding time (chosen as time unit).*

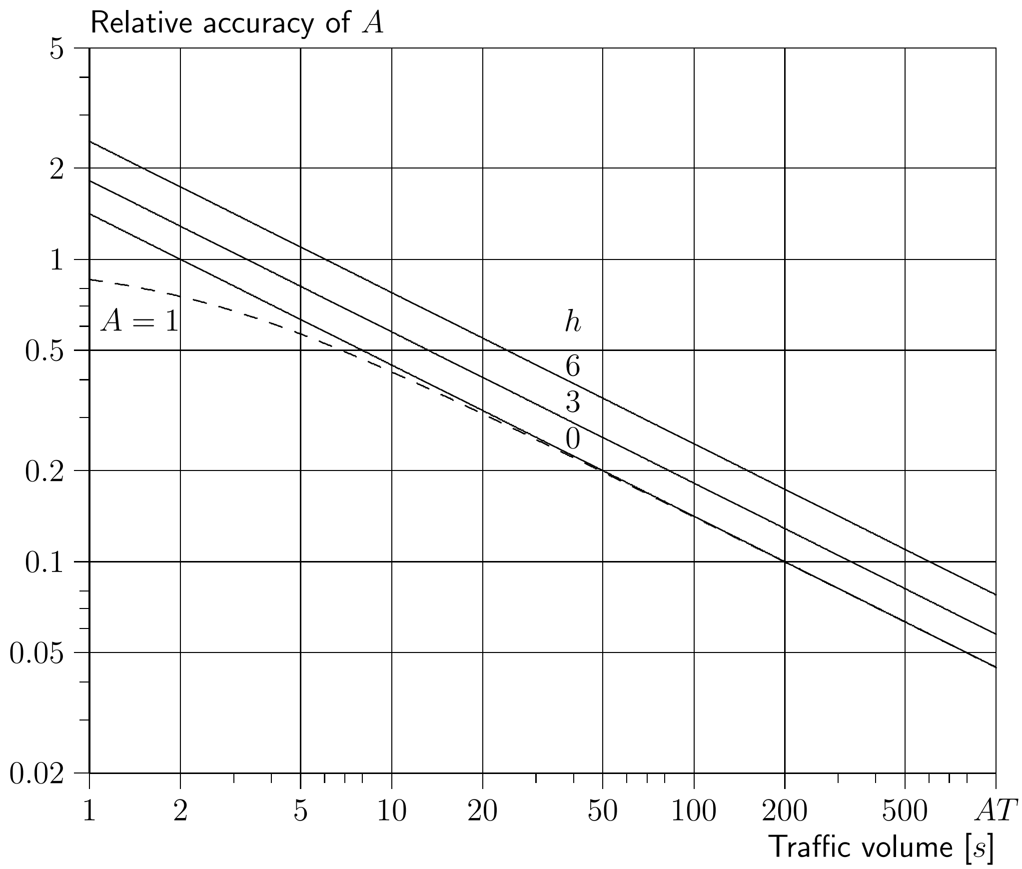


Figure 15.5: *Using double-logarithmic scale we obtain a linear relationship between the relative accuracy of the traffic intensity A and the measured traffic volume $A \cdot T$ when measuring in an unlimited time period. A scan interval $h = 0$ corresponds to a continuous measurement and $h > 0$ corresponds to the scanning method. The influence of a limited measuring method is shown by the dotted line for the case $A = 1$ erlang and a continuous measurement taking account of the limited measuring interval. T is measured in mean holding times.*

- Exploitation of knowledge about a limited measuring period results in more information for a short measurement ($T < 5$), whereas we obtain little additional information for $T > 10$. (There is correlation in the traffic process, and the first part of a measuring period yields more information than later parts).

- By using the roulette method we loose of course more information than by the scanning method (Iversen 1976, [36], 1977 [37]).

All the above mentioned factors have far less influence than the fact that the real holding times often deviate from the exponential distribution. In practice we often observe a form factor about 4–6.

The conclusion to be made from the above examples is that for practical applications it is more relevant to apply the elementary formula (15.8) with a correct form factor than to take account of the measuring method and the measuring period.

	Example a		Example b	
	σ_i^2	σ_i	σ_i^2	σ_i
Continuous Method				
Unlimited (15.8)	1.0000	1.0000	5.0000	2.2361
Limited	0.9500	0.9747	3.7729	1.9424
Scanning Method				
Unlimited (15.17)	1.0033	1.0016	5.4099	2.3259
Limited	0.9535	0.9765	4.2801	2.0688
Roulette Method				
Unlimited	1.1000	1.0488	7.5000	2.7386
Limited	1.0500	1.0247	6.2729	2.5046

Table 15.2: *Numerical comparison of various measuring principles in different time intervals.*

The above theory is exact when we consider charging of calls and measuring of time intervals. For stochastic computer simulations the traffic process in usually stationary, and the theory can be applied for estimation of the reliability of the results. However, the results are approximate as the theoretical assumptions about congestion free systems seldom are of interest.

In real life measurements on working systems we have traffic variations during the day, technical errors, measuring errors etc. Some of these factors compensate each other and the results we have derived give a good estimate of the reliability, and it is a good basis for comparing different measurements and measuring principles.

Bibliography

[1] Abate, J. & Whitt, W. (1997): Limits and approximations for the *M/G/1 LIFO* waiting–time distribution. Operations Research Letters, Vol. 20 (1997) : 5, 199–206.

[2] Andersen, B. & Hansen, N.H. & og Iversen, V.B. (1971): Use of minicomputer for telephone traffic measurements. Teleteknik (Engl. ed.) Vol. 15 (1971) : 2, 33–46.

[3] Ash, G.R. (1998): *Dynamic routing in telecommunications networks*. McGraw-Hill 1998. 746 pp.

[4] Baskett, F. & Chandy, K.M. & Muntz, R.R. & Palacios, F.G. (1975): Open, closed and mixed networks of queues with different classes of customers. Journal of the ACM, April 1975, pp. 248–260. (BCMP queueing networks).

[5] Bear, D. (1988): *Principles of telecommunication traffic engineering*. Revised 3rd Edition. Peter Peregrinus Ltd, Stevenage 1988. 250 pp.

[6] Bech, N.I. (1954): A method of computing the loss in alternative trunking and grading systems. The Copenhagen Telephone Company, May 1955. 14 pp. Translated from Danish: Metode til beregning af spærring i alternativ trunking- og graderingssystemer. Teleteknik, Vol. 5 (1954) : 4, pp. 435–448.

[7] Bolotin, V.A. (1994): Telephone circuit holding time distributions. ITC 14, 14th International Teletraffic Congress. Antibes Juan-les-Pins, France, June 6-10. 1994. Proccedings pp. 125–134. Elsevier 1994.

[8] Bretschneider, G. (1956): Bie Berechnung von Leitungsgruppen für überfließenden Verkehr. Nachrichtentechnische Zeitschrift, NTZ, Vol. 9 (1956) : 11, 533–540.

[9] Bretschneider, G. (1973): Extension of the equivalent random method to smooth traffics. ITC–7, Seventh International Teletraffic Congress, Stockholm, June 1973. Proceedings, paper 411. 9 pp.

[10] Brockmeyer, E. (1954): The simple overflow problem in the theory of telephone traffic. Teleteknik 1954, pp. 361–374. In Danish. English translation by Copenhagen Telephone Company, April 1955. 15 pp.

[11] Brockmeyer, E. & Halstrøm, H.L. & Jensen, Arne (1948): *The life and works of A.K. Erlang*. Transactions of the Danish Academy of Technical Sciences, 1948, No. 2, 277 pp. Copenhagen 1948.

[12] Burke, P.J. (1956): The output of a queueing system. Operations Research, Vol. 4 (1956), 699–704.

[13] Christensen, P.V. (1914): The number of selectors in automatic telephone systems. The Post Office Electrical Engineers Journal, Vol. 7 (1914), 271–281.

[14] Cobham, A. (1954): Priority assignment in waiting line problems. Operations Research, Vol. 2 (1954), 70–76.

[15] Conway, A.E. & Georganas, N.D. (1989): *Queueing networks – exact computational algorithms: A unified theory based on decomposition and aggregation.* The MIT Press 1989. 234 pp.

[16] Cooper, R.B. (1972): *Introduction to queueing theory*. New York 1972. 277 pp.

[17] Cox, D.R. (1955): A use of complex probabilities in the theory of stochastic processes. Proc. Camb. Phil. Soc., Vol. 51 (1955), pp. 313–319.

[18] Cox, D.R. & Miller, H.D. (1965): *The theory of stochastic processes*. Methuen & Co. London 1965. 398 pp.

[19] Cox, D.R.& Isham, V. (1980): *Point processes*. Chapman and Hall. 1980. 188 pp.

[20] Crommelin, C.D. (1932): Delay probability formulae when the holding times are constant. Post Office Electrical Engineers Journal, Vol. 25 (1932), pp. 41–50.

[21] Crommelin, C.D. (1934): Delay probability formulae. Post Office Electrical Engineers Journal, Vol. 26 (1934), pp. 266–274.

[22] Delbrouck, L.E.N. (1983): On the steady–state distribution in a service facility carrying mixtures of traffic with different peakedness factor and capacity requirements. IEEE Transactions on Communications, Vol. COM–31 (1983) : 11, 1209–1211.

[23] Dickmeiss, A. & Larsen, M. (1993): *Spærringsberegninger i telenet* (Blocking calculations in telecommunication networks, in Danish). Master's thesis. Institut for Telekommunikation, Danmarks Tekniske Højskole, 1993. 141 pp.

[24] Eilon, S. (1969): A simpler proof of $L = \lambda\, W$. Operations Research, Vol. 17 (1969), pp. 915–917.

[25] Elldin, A., and G. Lind (1964): *Elementary telephone traffic theory*. Chapter 4. L.M. Ericsson AB, Stockholm 1964. 46 pp.

[26] Engset, T.O. (1918): Die Wahrscheinlichkeitsrechnung zur Bestimmung der Wählerzahl in automatischen Fernsprechämtern. Elektrotechnische Zeitschrift, 1918, Heft 31. Translated to English in Telektronikk (Norwegian), June 1991, 4pp.

[27] Feller, W. (1950): *An introduction to probability theory and its applications*. Vol. 1, New York 1950. 461 pp.

[28] Fortet, R. & Grandjean, Ch. (1964): Congestion in a loss system when some calls want several devices simultaneously. Electrical Communications, Vol. 39 (1964) : 4, 513–526. Paper presented at ITC–4, Fourth International Teletraffic Congress, London. England, 15–21 July 1964.

[29] Fredericks, A.A. (1980): Congestion in blocking systems – a simple approximation technique. The Bell System Technical Journal, Vol. 59 (1980) : 6, 805–827.

[30] Fry, T.C. (1928): *Probability and its engineering uses*. New York 1928, 470 pp.

[31] Gordon, W.J., and & Newell, G.F. (1967): Closed queueing systems with exponential servers. Operations Research, Vol. 15 (1967), pp. 254–265.

[32] Grillo, D. & Skoog, R.A. & Chia, S. & Leung, K.K. (1998): Teletraffic engineering for mobile personal communications in ITU–T work: the need to match theory to practice. IEEE Personal Communications, Vol. 5 (1998) : 6, 38–58.

[33] Hayward, W.S. Jr. (1952): The reliability of telephone traffic load measurements by switch counts. The Bell System Technical Journal, Vol. 31 (1952) : 2, 357–377.

[34] ITU-T (1993): Traffic intensity unit. ITU–T Recommendation B.18. 1993. 1 p.

[35] Iversen, V.B. (1973): Analysis of real teletraffic processes based on computerized measurements. Ericsson Technics, No. 1, 1973, pp. 1–64. *"Holbæk measurements"*.

[36] Iversen, V.B. (1976): *On the accuracy in measurements of time intervals and traffic intensities with application to teletraffic and simulation.* Ph.D.–thesis. IMSOR, Technical University of Denmark 1976. 202 pp.

[37] Iversen, V.B. (1976): On general point processes in teletraffic theory with applications to measurements and simulation. ITC-8, Eighth International Teletraffic Congress, paper 312/1–8. Melbourne 1976. Published in Teleteknik (Engl. ed.) 1977 : 2, pp. 59–70.

[38] Iversen, V.B. (1980): The A–formula. Teleteknik (English ed.), Vol. 23 (1980) : 2, 64–79.

[39] Iversen, V.B. (1982): Exact calculation of waiting time distributions in queueing systems with constant holding times. NTS-4, Fourth Nordic Teletraffic Seminar, Helsinki 1982. 31 pp.

[40] Iversen, V.B. (1987): The exact evaluation of multi–service loss system with access control. Teleteknik, English ed., Vol 31 (1987) : 2, 56–61. NTS–7, Seventh Nordic Teletraffic Seminar, Lund, Sweden, August 25–27, 1987, 22 pp.

[41] Iversen, V.B. & Nielsen, B.F. (1985): Some properties of Coxian distributions with applications. Proceedings of the International Conference on Modelling Techniques and Tools for Performance Analysis, pp. 61–66. 5–7 June, 1985, Valbonne, France. North–Holland Publ. Co. 1985. 365 pp. (Editor N. Abu El Ata).

[42] Iversen, V.B. & Stepanov, S.N. (1997): The usage of convolution algorithm with truncation for estimation of individual blocking probabilities in circuit-switched telecommunication networks. Proceedings of the 15th International Teletraffic Congress, ITC 15, Washington, DC, USA, 22–27 June 1997. 1327–1336.

[43] Iversen, V.B. & Sanders, B. (2001): Engset formulæ with continuous parameters – theory and applications. AEÜ, International Journal of Electronics and Communications, Vol. 55 (2001) : 1, 3-9.

[44] Jackson, R.R.P. (1954): Queueing systems with phase type service. Operational Research Quarterly, Vol. 5 (1954), 109–120.

[45] Jackson, J.R. (1957): Networks of waiting lines. Operations Research, Vol. 5 (1957), pp. 518–521.

[46] Jackson, J.R. (1963): Jobshop–like queueing systems. Management Science, Vol. 10 (1963), No. 1, pp. 131–142.

[47] Jensen, Arne (1948): An elucidation of A.K. Erlang's statistical works through the theory of stochastic processes. Published in "*The Erlangbook*": E. Brockmeyer, H.L. Halstrøm and A. Jensen: *The life and works of A.K. Erlang*. København 1948, pp. 23–100.

[48] Jensen, Arne (1948): Truncated multidimensional distributions. Pages 58–70 in "The Life and Works of A.K. Erlang". Ref. Brockmeyer et al., 1948 [47].

[49] Jensen, Arne (1950): *Moe's Principle – An econometric investigation intended as an aid in dimensioning and managing telephone plant. Theory and Tables*. Copenhagen 1950. 165 pp.

[50] Jerkins, J.L. & Neidhardt, A.L. & Wang, J.L. & Erramilli A. (1999): Operations measurement for engineering support of high-speed networks with self-similar traffic. ITC 16, 16th International Teletraffic Congress, Edinburgh, June 7–11, 1999. Proceedings pp. 895–906. Elsevier 1999.

[51] Johannsen, Fr. (1908): "Busy". Copenhagen 1908. 4 pp.

[52] Johansen, K. & Johansen, J. & Rasmussen, C. (1991): The broadband multiplexer, "TransMux 1001". Teleteknik, English ed., Vol. 34 (1991) : 1, 57–65.

[53] Joys, L.A.: Variations of the Erlang, Engset and Jacobæus formulæ. ITC–5, Fifth International Teletraffic Congress, New York, USA, 1967, pp. 107–111. Also published in: Teleteknik, (English edition), Vol. 11 (1967) :1 , 42–48.

[54] Joys, L.A. (1968): Engsets formler for sannsynlighetstetthet og dens rekursionsformler. (Engset's formulæ for probability and its recursive formulæ, in Norwegian). Telektronikk 1968 No 1–2, pp. 54–63.

[55] Joys, L.A. (1971): Comments on the Engset and Erlang formulae for telephone traffic losses. Thesis. Report TF No. 25/71, Research Establishment, The Norwegian Telecommunications Administration. 1971. 127 pp.

[56] Karlsson, S.A. (1937): Tekniska anordninger för samtalsdebitering enligt tid (Technical arrangement for charging calls according to time, In Swedish). Helsingfors Telefonförening, Tekniska Meddelanden 1937, No. 2, pp. 32–48.

[57] Kaufman, J.S. (1981): Blocking in a shared resource environment. IEEE Transactions on Communications, Vol. COM–29 (1981) : 10, 1474–1481.

[58] Keilson, J. (1966): The ergodic queue length distribution for queueing systems with finite capacity. Journal of Royal Statistical Society, Series B, Vol. 28 (1966), 190–201.

[59] Kelly, F.P. (1979): *Reversibility and stochastic networks*. John Wiley & Sons, 1979. 230 pp.

[60] Kendall, D.G. (1951): Some problems in the theory of queues. Journal of Royal Statistical Society, Series B, Vol. 13 (1951) : 2, 151–173.

[61] Kendall, D.G. (1953): Stochastic processes occuring in the theory of queues and their analysis by the method of the imbedded Markov chain. Ann. Math. Stat., Vol. 24 (1953), 338–354.

[62] Khintchine, A.Y. (1955): *Mathematical methods in the theory of queueing*. London 1960. 124 pp. (Original in Russian, 1955).

[63] Kingman, J.F.C. (1969): Markov population processes. J. Appl. Prob., Vol. 6 (1969), 1–18.

[64] Kleinrock, L. (1964): *Communication nets: Stochastic message flow and delay*. McGraw–Hill 1964. Reprinted by Dover Publications 1972. 209 pp.

[65] Kleinrock, L. (1975): *Queueing systems*. Vol. I: *Theory*. New York 1975. 417 pp.

[66] Kleinrock, L. (1976): *Queueing systems*. Vol. II: *Computer applications*. New York 1976. 549 pp.

[67] Kosten, L. (1937): Über Sperrungswahrscheinlichkeiten bei Staffelschaltungen. Elek. Nachr. Techn., Vol. 14 (1937) 5–12.

[68] Kraimeche, B. & Schwartz, M. (1983): Circuit access control strategies in integrated digital networks. IEEE INFOCOM, April 9–12, 1984, San Francisco, USA, Proceedings pp. 230–235.

[69] Kruithof, J. (1937): Telefoonverkehrsrekening. De Ingenieur, Vol. 52 (1937) : E15–E25.

[70] Kuczura, A. (1973): The interrupted Poisson process as an overflow process. The Bell System Technical Journal, Vol. 52 (1973) : 3, pp. 437–448.

[71] Kuczura, A. (1977): A method of moments for the analysis of a switched communication network's performance. IEEE Transactions on Communications, Vol. Com–25 (1977) : 2, 185–193.

[72] Lavenberg, S.S. & Reiser, M. (1980): Mean–value analysis of closed multichain queueing networks. Journal of the Association for Computing Machinery, Vol. 27 (1980) : 2, 313–322.

[73] Lind, G. (1976): Studies on the probability of a called subscriber being busy. ITC–8, Eighth International Teletraffic Congress, Melbourne, November 1976. Paper 631. 8 pp.

[74] Listov–Saabye, H. & Iversen V.B. (1989): *ATMOS*: a PC–based tool for evaluating multi–service telephone systems. IMSOR, Technical University of Denmark 1989, 75 pp. (In Danish).

[75] Little, J.D.C. (1961): A proof for the queueing formula $L = \lambda\, W$. Operations Research, Vol. 9 (1961) : 383–387.

[76] Maral, G. (1995): *VSAT networks*. John Wiley & Sons, 1995. 282 pp.

[77] Marchal, W.G. (1976): An approximate formula for waiting time in single server queues. AIIE Transactions, December 1976, 473–474.

[78] Palm, C. (1941): Mättnoggrannhet vid bestämning af trafikmängd enligt genomsökningsförfarandet (Accuracy of measurements in determining traffic volumes by the scanning method). Tekn. Medd. K. Telegr. Styr., 1941, No. 7–9, pp. 97–115.

[79] Palm, C. (1943): *Intensitätsschwankungen im Fernsprechverkehr*. Ericsson Technics, No. 44, 1943, 189 pp. English translation by Chr. Jacobæus: Intensity Variations in Telephone Traffic. North–Holland Publ. Co. 1987.

[80] Palm, C. (1947): The assignment of workers in servicing automatic machines. Journal of Industrial Engineering, Vol. 9 (1958) : 28–42. First published in Swedish in 1947.

[81] Palm, C. (1947): *Table of the Erlang loss formula*. Telefonaktiebolaget L M Ericsson, Stockholm 1947. 23 pp.

[82] Palm, C. (1957): Some propositions regarding flat and steep distribution functions, pp. 3–17 in TELE (English edition), No. 1, 1957.

[83] Pinsky, E. & Conway, A.E. (1992): *Computational algorithms for blocking probabilities in circuit–switched networks*. Annals of Operations Research, Vol. 35 (1992) 31–41.

[84] Postigo–Boix, M. & García–Haro, J. & Aguilar–Igartua, M. (2001): (Inverse Multiplexing of ATM) *IMA* – technical foundations, application and performance analysis. Computer Networks, Vol. 35 (2001) 165–183.

[85] Press, W.H. & Teukolsky, S.A. & Vetterling, W.T. & Flannery, B.P. (1995): *Numerical recipes in C, the art of scientific computing*. 2nd edition. Cambridge University Press, 1995. 994 pp.

[86] Rabe, F.W. (1949): Variations of telephone traffic. Electrical Communications, Vol. 26 (1949) 243–248.

[87] Rapp, Y. (1965): Planning of junction network in a multi–exchange area. Ericsson Technics 1965, No. 2, pp. 187–240.

[88] Riordan, J. (1956): Derivation of moments of overflow traffic. Appendix 1 (pp. 507–514) in (Wilkinson, 1956 [103]).

[89] Roberts, J.W. (1981): A service system with heterogeneous user requirements – applications to multi–service telecommunication systems. *Performance of data communication systems and their applications*. G. Pujolle (editor), North–Holland Publ. Co. 1981, pp. 423–431.

[90] Roberts, J.W. (2001): Traffic theory and the Internet. IEEE Communications Magazine Vol. 39 (2001) : 1, 94–99.

[91] Ross, K.W. & Tsang, D. (1990): Teletraffic engineering for product–form circuit–switched networks. Adv. Appl. Prob., Vol. 22 (1990) 657–675.

[92] Ross, K.W. & Tsang, D. (1990): Algorithms to determine exact blocking probabilities for multirate tree networks. IEEE Transactions on Communications. Vol. 38 (1990) : 8, 1266–1271.

[93] Rönnblom, N. (1958): Traffic loss of a circuit group consisting of both–way circuits which is accessible for the internal and external traffic of a subscriber group. TELE (English edition), 1959 : 2, 79–92.

[94] Sanders, B. & Haemers, W.H. & Wilcke, R. (1983): Simple approximate techniques for congestion functions for smooth and peaked traffic. ITC–10, Tenth International Teletraffic Congress, Montreal, June 1983. Paper 4.4b–1. 7 pp.

[95] Stepanov, S.S. (1989): Optimization of numerical estimation of characteristics of multiflow models with repeated calls. Problems of Information Transmission, Vol. 25 (1989) : 2, 67–78.

[96] Sutton, D.J. (1980): The application of reversible Markov population processes to teletraffic. A.T.R. Vol. 13 (1980) : 2, 3–8.

[97] Techguide (2001): Inverse Multiplexing – scalable bandwidth solutions for the WAN. Techguide (The Technologu Guide Series), 2001, 46 pp. <www.techguide.com>

[98] Vaulot, É. & Chaveau, J. (1949): Extension de la formule d'Erlang au cas ou le trafic est fonction du nombre d'abonnés occupés. Annales de Télécommunications, Vol. 4 (1949) 319–324.

[99] Veirø, B. (2002): Proposed Grade of Service chapter for handbook. ITU–T Study Group 2, WP 3/2. September 2001. 5 pp.

[100] Villén, M. (2002): Overview of ITU Recommendations on traffic engineering. ITU–T Study Group 2, COM 2-KS 48/2-E. May 2002. 21 pp.

[101] Wallström, B. (1964): A distribution model for telephone traffic with varying call intensity, including overflow traffic. Ericsson Technics, 1964, No. 2, pp. 183–202.

[102] Wallström, B. (1966): Congestion studies in telephone systems with overflow facilities. Ericsson Technics, No. 3, 1966, pp. 187–351.

[103] Wilkinson, R.I. (1956): Theories for toll traffic engineering in the U.S.A. The Bell System Technical Journal, Vol. 35 (1956) 421–514.

Author index

Index