Teletraffic engineering and network planning

Iversen, Villy Bæk

Publication date:
2015

Document Version
Publisher's PDF, also known as Version of record

TELETRAFFIC ENGINEERING
and
NETWORK PLANNING

Revised 2015

Villy B. Iversen

Department of Photonics Engineering
Technical University of Denmark
H C Ørsteds Plads 343
DK–2800 Kgs. Lyngby

http://www.fotonik.dtu.dk

villy.b.iversen@gmail.com

PREFACE

This book covers the basic theory of teletraffic engineering. The mathematical background required is elementary probability theory. The purpose of the book 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,
- Mathematical background: Chapter 2 – 3,
- Telecommunication loss models: Chapter 4 – 8,
- Data communication delay models: Chapter 9 – 12,
- Measurement and simulation: Chapter 13.

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 book 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.

Villy Bæk Iversen
2015

Contents

Notations

a	Offered traffic per source
A	Offered traffic $= A_o$
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 telecommunication network or number of nodes in a queueing network
L	Mean queue length
$L_{k\emptyset}$	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 central 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 normalized) state probabilities
$Q(i)$	Cumulated values of $q(i)$: $Q(i) = \sum_{x=-\infty}^{i} q(x)$
Q	Normalization 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	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	Utilization = mean carried traffic per channel, y_i = traffic carried by channel i
Y	Total carried traffic
Z	Peakedness

α	Carried traffic per source
β	Offered traffic per idle source
γ	Arrival rate for an idle source
ε	Palm's form factor
ϑ	Lagrange-multiplier
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

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 computer 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, 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 relationships 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 effective 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 for example the determination of the number of channels in a

base station of a cellular network, the number of operators in a call center, the number of open lanes in the supermarket, and the allocation of priorities to jobs in a computer system. *Long term decisions* include decisions concerning the development and extension of data- and telecommunication networks, extension of cables, radio links, establishing a new base station, etc.

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

1.1 Modeling of telecommunication systems

For the analysis of a telecommunication system, a model of the system considered must be set up. Especially for applications of teletraffic theory to the design of new systems, this modeling process is of fundamental importance. It requires knowledge of the technical system, available mathematical tools, and the implementation of the model in 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.

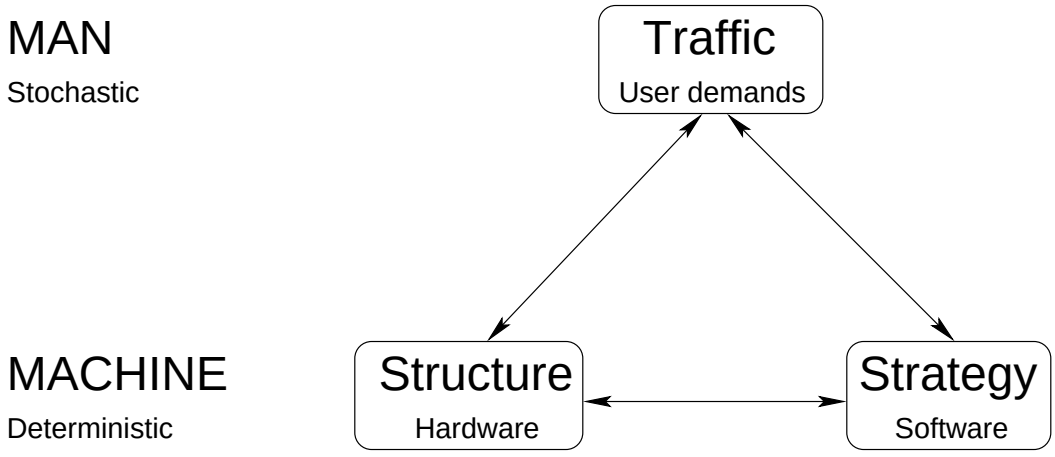


Figure 1.1: *Telecommunication systems are complex man/machine systems. The task of teletraffic theory is to configure optimal systems from knowledge of user requirements and behavior.*

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 random processes as failures occur more or less at random, and they can be dealt with as traffic with highest priority. The system structure is given by hardware and software which is described in manuals. In road traffic systems, roads, traffic signals, roundabouts, etc. make up the structure.

1.1.2 Operational strategy

A given physical system can be used in different ways in order to adapt the system to the traffic demand. In road traffic, it is implemented with traffic rules and strategies which may adapt to traffic variations during the day.

In a computer, this adaption takes place by means of the operating 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 calls already accepted, 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 modeled by statistical properties of the traffic. It is only possible to validate that a mathematical models is in agreement with reality by comparing results obtained from the model with measurements on real systems. This process must necessarily be of 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 agreement, a new iteration of the process must take place.

It appears natural to split the description of the traffic properties into random 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 arrive. Models also exists for describing the behavior 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.

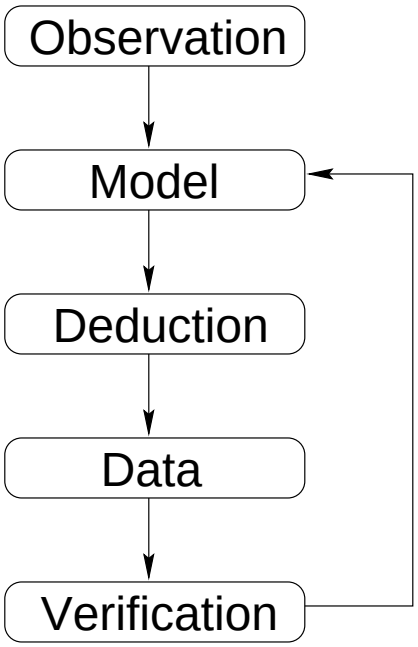


Figure 1.2: *Teletraffic theory is an inductive discipline. From observations of real systems we establish theoretical models, from which we derive parameters, which can be compared with corresponding observations from the real system. If there is agreement, the model has been validated. If not, then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

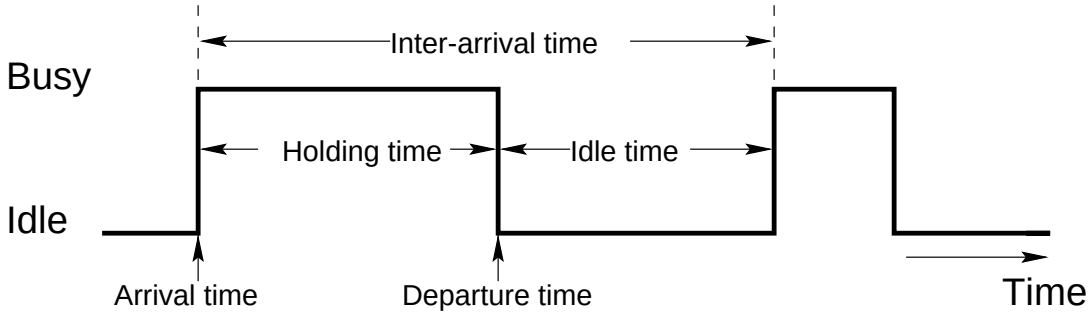


Figure 1.3: *Illustration of the terminology applied for a traffic process. Notice the difference between time intervals and instants of time. We use the terms arrival, call, and connection synonymously. The inter-arrival time, respectively the inter-departure time, are the time intervals between arrivals,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

1.1.4 Models

General requirements to an engineering model are:

1. It must without major difficulties be possible to verify the model and 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 calls in a telephone exchange, which changes 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 *random (= stochastic) process*, and the inter arrival times between call attempts are described by probability distributions which characterize the random process.

We may classify models into three classes:

1. *Mathematical models* which are very general, but often approximate. We may optimize the parameters analytically or numerically.

2. Simulation models where we may use either measured data or artificial data from statistical distributions. It is more resource demanding to work with simulation models as they are not very general. Every individual case must be simulated.

3. *Physical models* (prototypes) are even much 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 of what happens when a call attempt arrives to a traditional telephone exchange. 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. Transit switches are used to interconnect local exchanges or to increase the accessibility and availability.

1.2.1 System structure

Let us consider a historical telephone exchange of the crossbar type. Even though this type has been taken out of service, 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).

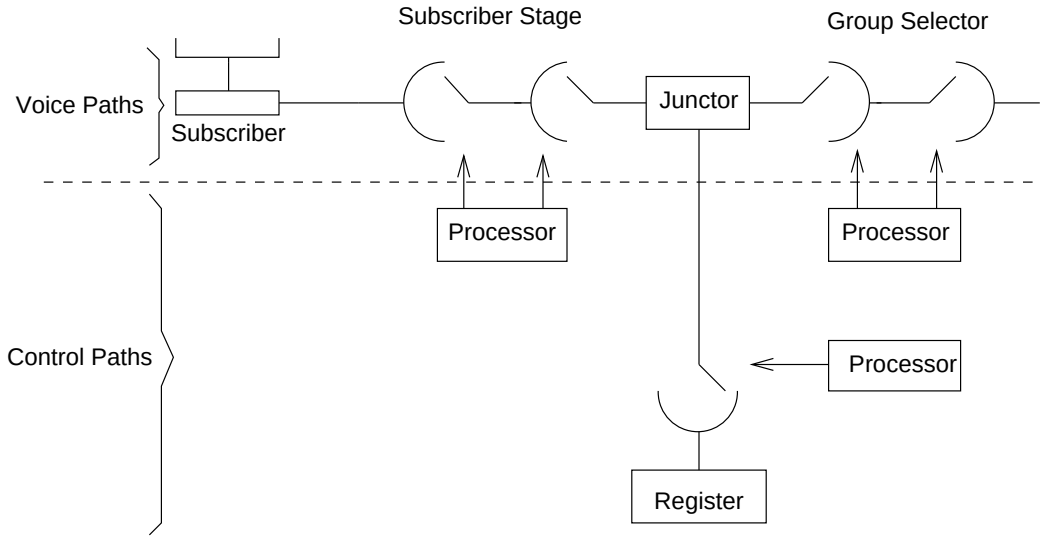


Figure 1.4: *Fundamental structure of a switching system.*

The voice paths (cf. data) are occupied during the whole duration of the call (on the average 2–3 minutes) while the control paths only are occupied during the phase of call establishment (in 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 division 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 (cf. control plane) are responsible for establishing the connection. Usually, this happens in a number of steps where each step is performed by a control device: a *microprocessor*, or a *register* (originally a human operator).

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 when the conversation ends (performed sometimes by the voice path itself).

In addition the charging of the calls must also be taken care of. In conventional exchanges the control path is build up on relays and/or electronic devices and the logical operations are implemented by *wired logic*. Changes in functions require hardware modifications which are complex and expensive.
In digital exchanges the control devices are processors. The logical functions are carried out by software, and changes are much easier 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 still consider a conventional telephone system. When an *A-subscriber* initiates a call, the handset is activated and the wired pair connecting the subscriber to the exchange 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 *junctor (cord)*. The subscriber line and the junctor are 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/junctor has three outlets.

Through another switching stage, the register selector, a *register* is connected to the cord/junctor. Thereby the subscriber line is connected to a register via the cord/junctor. This phase takes less than one second.

The register sends a dial tone to the A-subscriber who dials the digits of the telephone number of the *B-subscriber*; the digits are received and stored by the register. The duration of this phase depends on the subscriber.

A microprocessor analyzes the *B-number* and by means of a group selector establishes a connection to the desired B-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 a code receiver at the desired exchange which then transfers the information to the registers of this exchange.

The register of the originating exchange has now fulfilled its obligations and is released so it is idle for serving new call attempts. The microprocessors work very fast (around 1–10 ms) and independently of the subscribers. The cord/junctor is occupied during the entire 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), charging information, and release 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 Sec. 1.6. Call attempts towards a subscriber take place in a similar way. A code receiver at the exchange of the B-subscriber receives the digits and a connection is set up through the group selector stage and the local switch stage to the B-subscriber by using the registers of the destination exchange.

1.2.3 Operational strategy

The voice paths normally operate as loss systems while the control path operates as delay systems (Sec. 1.6).

If there is not both an idle junctor as well as an idle register then the subscriber will get no dial tone no matter how long time he waits. If no outlet is idle 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 never be established any connection.

If a microprocessor (or all microprocessors of a specific type when there are more than one) is busy, then the call will wait until the microprocessor becomes idle. Due to the very short holding time the 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 usually be served in random order independent of the time of arrival.

The way by which control devices of the same type and the junctors/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 seldom will get the same faulty junctor/cord or control path again if the call attempt is repeated.
If a control path is occupied longer 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. Also it is only possible to activate the ringing tone towards a B-subscriber for a limited time interval. Therefore, a B-subscriber line is only blocked for a limited time for each call attempt. An exchange must be able to operate independently of subscriber behaviour.

The cooperation between the different parts takes place in accordance to strict and well defined rules, called protocols, which in conventional systems is determined by the wired logic and in software controlled 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* = Narrow-band *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 are based on *ATM* = Asynchronous Transfer Mode and *MPLS* (Multi Protocol Label Switching).

1.3 Wireless 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) instead of conventional wired 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 this is the limited resource in wireless communication systems. The optimal utilization of this resource is a main issue in the cellular technology. In the following subsection generic systems are described.

1.3.1 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 attenuated 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 the base stations cover the planned traffic area without too much overlapping between neighbouring base-stations. It may not be possible to use the same radio frequency in two neighbour base stations but two base stations without a common border can use the same frequency thereby allowing the frequency bands to be reused.

In Fig. 1.5 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.
Frequency allocation is a complex problem. In addition to the restrictions given above, a number of other limitations also exist. For example, there has to be a certain distance (number of channels) between two channels at the same base station (neighbour channel restriction) and to avoid interference also other restrictions exist.

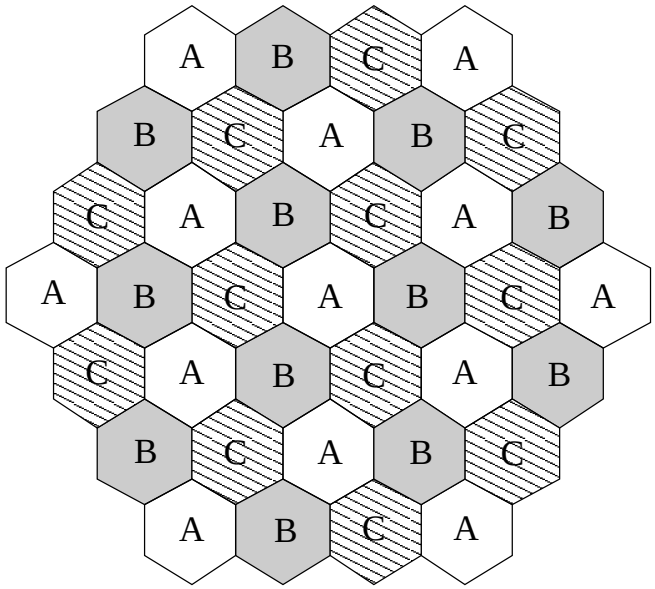


Figure 1.5: *Cellular mobile communication system. By dividing the frequencies into 3 groups (A, B and C) they can be reused as shown.*

Strategy. In mobile telephone systems we need a database with information about all subscribers. 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 in 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) or busy with a call, the A-subscriber is informed that the B-subscriber is non-available or busy. Is the B-subscriber active, then the number is sent out on all control channels in the traffic area. The B-subscriber recognizes his own number and informs via the control channel the system about the identity of the cell (base station) in which he is located. 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 recording the digits and testing the accessibility 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 (S/N) ratio. The *MTX* transfers 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, usually 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. From the identity of the subscriber the *MTX* in the new area is 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 are always routed the home-*MTX* which will then forward the call to the new *MTX*. Outgoing calls will are taken care of in 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 centers 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 low orbit satellites correspond to base stations. 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 kilometers above the Earth. Iridium was unsuccessful, but newer systems such as the *Inmarsat* system are now in operation.

1.3.2 Wireless Broadband Systems

In these systems we have an analogue high capacity channel, for example 10 Mhz, which is turned into a digital channel with a capacity up to 100 Mbps, depending on the coding scheme which again depends on the quality of the channel. The digital channel (media) is shared my many users according to a media access control (MAC) protocol.

If all services have the same constant bandwidth demand we could split the digital channel up into many constant bit rate channels. This was done in classical systems by frequency division multiple access, *FDMA*.

Most data and multimedia services have variable bandwidth demand during the occupation time. Therefore, the digital channel in time is split up into time-slots, and we apply time

division multiple access, *TDMA*. A certain number of time slots make up a frame, which is repeated infinitely during time. Thus a time slot in each frame corresponding to the minimum bandwidth allocated. The information transmitted by a user is thus aggregated and transmitted in one or more slots in every frame. The frame size specifies the maximum delay the information experience due to the slotted time. Slot size and frame size should be specified according to the quality of service and restrictions from coding and switching mechanisms. One slot in a frame in *TDMA* thus corresponds to one channel in *FDMA*. The advantage of *TDMA* is that we can change the allocation of slots from one frame to frame and thus reallocate the bandwidth resources very fast.

Service classes

In most digital service-integrated systems we specify four services classes: two for real-time services and two for non-real-time services.

- Real-time services
  - Constant bit-rate real time services. These services require a constant bandwidth. Examples are voice services as *ISDN* and *VoIP* (voice over IP). For this kind a services we have to reserve a fixed number of slots in each frame.
  - Variable bit-rate real time services. These services have a variable bandwidth demand. Examples are most data services. Also voice and video services with codecs (coder/decoder) having variable bit-rate. During each frame we allocate a certain capacity to a service. We may have restrictions upon the maximum number of slots, the average number of slots, etc.
- Non Real-time services
  - Non real-time polling services. This is services which do not require real time transmission, but there may be restrictions on the minimum bandwidth allocated. The services ask for a certain number of slots, and the system allocate slots in each frame dependent on the number of idle slots.
  - Best effort traffic. This traffic uses the remaining capacity left over from the other services. Also here we may guarantee a certain minimum bandwidth. This could for example be ftp-traffic.

By traffic engineering we develop strategies for connections acceptance control *CAC*, specify strategies for allocation of capacity to the classes, so that we can fulfil the service level agreement (*SLA*) between user and operator. We also specify policing agreements to ensure that the user traffic conform with the agreed parameters. The *SLA* specifies the quality-of-service (QoS) guaranteed by the operator. For each service there may be different levels of QoS, for example named Gold, Silver, and Bronze. A subscriber asking for Gold service will require more resources and also pay more for the service. The task of traffic engineering is simultaneously to maximize the utilization of the resources and fulfil QoS requirements.

1.4 Communication networks

There exists different kinds of communications networks: telephone 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, MPLS*).

1.4.1 Classical telephone network

The telephone network has traditionally been build up as a hierarchical structure (Fig. 1.7). 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. 1.6). 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.

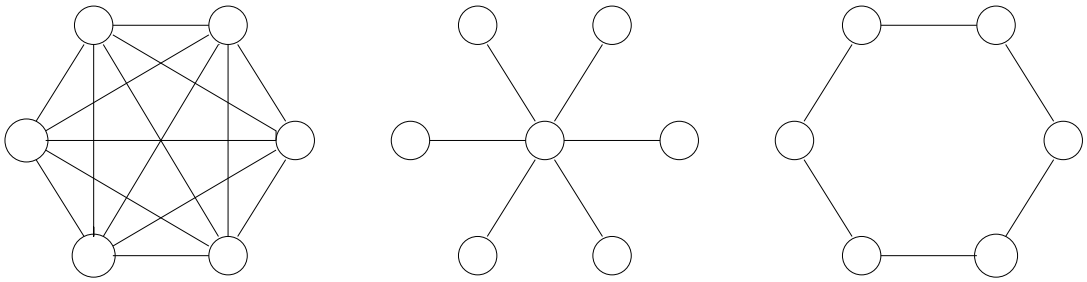


Figure 1.6: *There are three basic structures of networks: mesh, star and ring. Mesh networks are applicable when there are few large exchanges (upper part of the hierarchy, also named polygon network), whereas star networks are proper when there are many small exchanges (lower part of the hierarchy). 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.

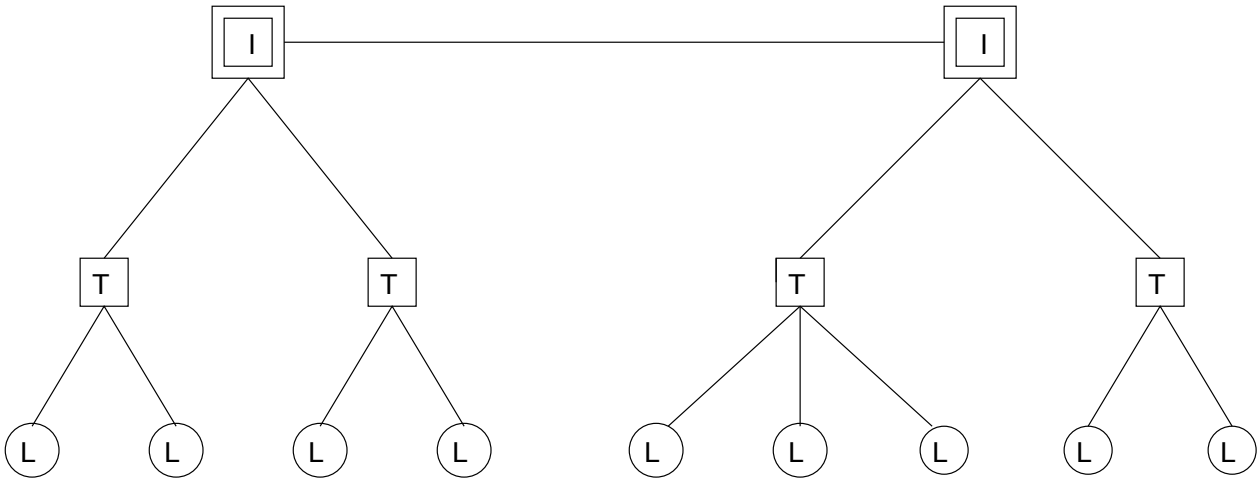


Figure 1.7: *In a telecommunication network all exchanges are typically arranged in a three-level hierarchy. Local-exchanges or subscriber-exchanges (L), to which the subscribers are connected, are connected to main exchanges (T), which again are connected to inter-urban exchanges (I). An inter-urban 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.*

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 characterized by the fact that before any two subscribers can communicate, a two-way (duplex) *connection* must be created, and the connection must exist 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* and *MPLS*) or connection-less. In the discipline of network planning, the objective is to optimize network structures and traffic routing under the consideration of traffic demands, service and reliability requirement etc.

Example 1.4.1: VSAT-networks
VSAT-networks (Maral, 1995 [87]) are for instance used by multi-national organizations for transmission of speech and data between different divisions of news-broadcasting, in case of disasters, 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 utilization 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 ($VSAT \rightarrow hub$ and $hub \rightarrow VSAT$) 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. □

1.4.2 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 by *packet switching network,* which works according to the *store-and-forward* principle (see Fig. 1.8). The data to be transmitted are sent from transmitter to receiver step-by-step 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 any acknowledgement within some given time interval a new copy of the packet (or a whole

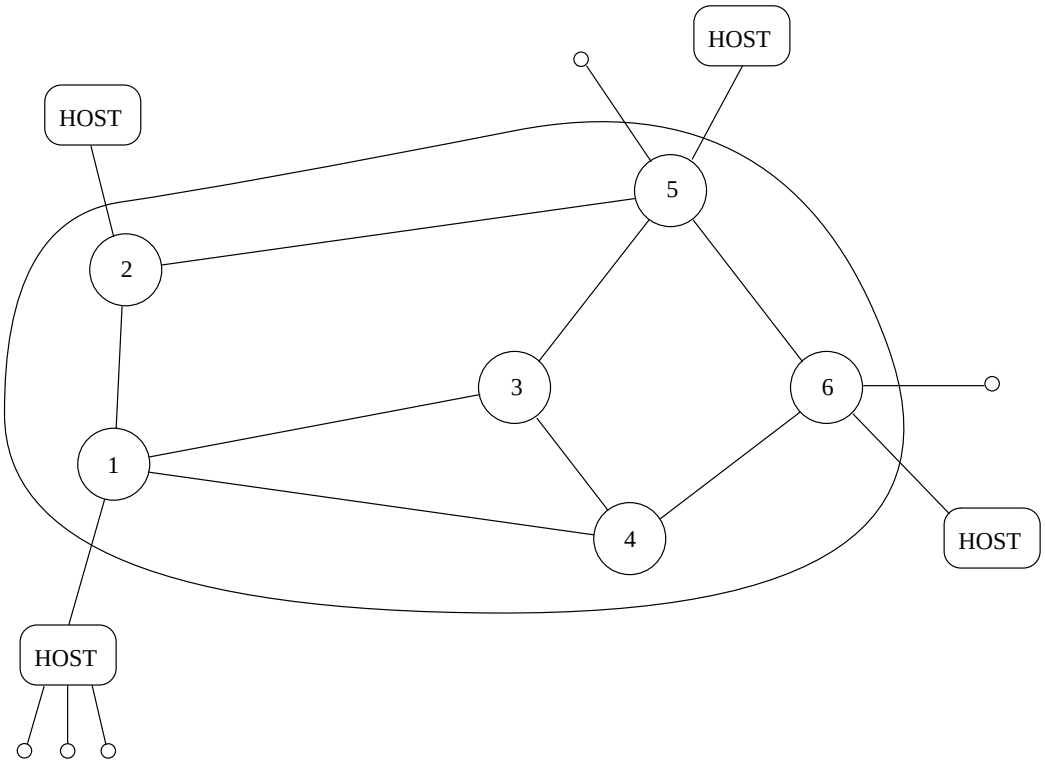


Figure 1.8: *Datagram network: Store- and forward principle for a packet switching data network.*

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.

1.4.3 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 should be received and stored. A terminal wanting to transmit a packet transmits 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 (exponential 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 circulating between the terminals according to predefined rules.

Local area networks based on the *ATM* technique are also in operation. Furthermore, wireless *LAN*s are very 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.

1.5 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 [118]). *The International Telecommunication Union* (*ITU*) is an organization sponsored by the *United Nations* for promoting international telecommunications. It has three sectors:

- Telecommunication Standardization Sector (*ITU–T*),
- Radio communication Sector (*ITU–R*), and
- 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 recommen-

dations 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.

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 characterization;
- 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 characterize 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.

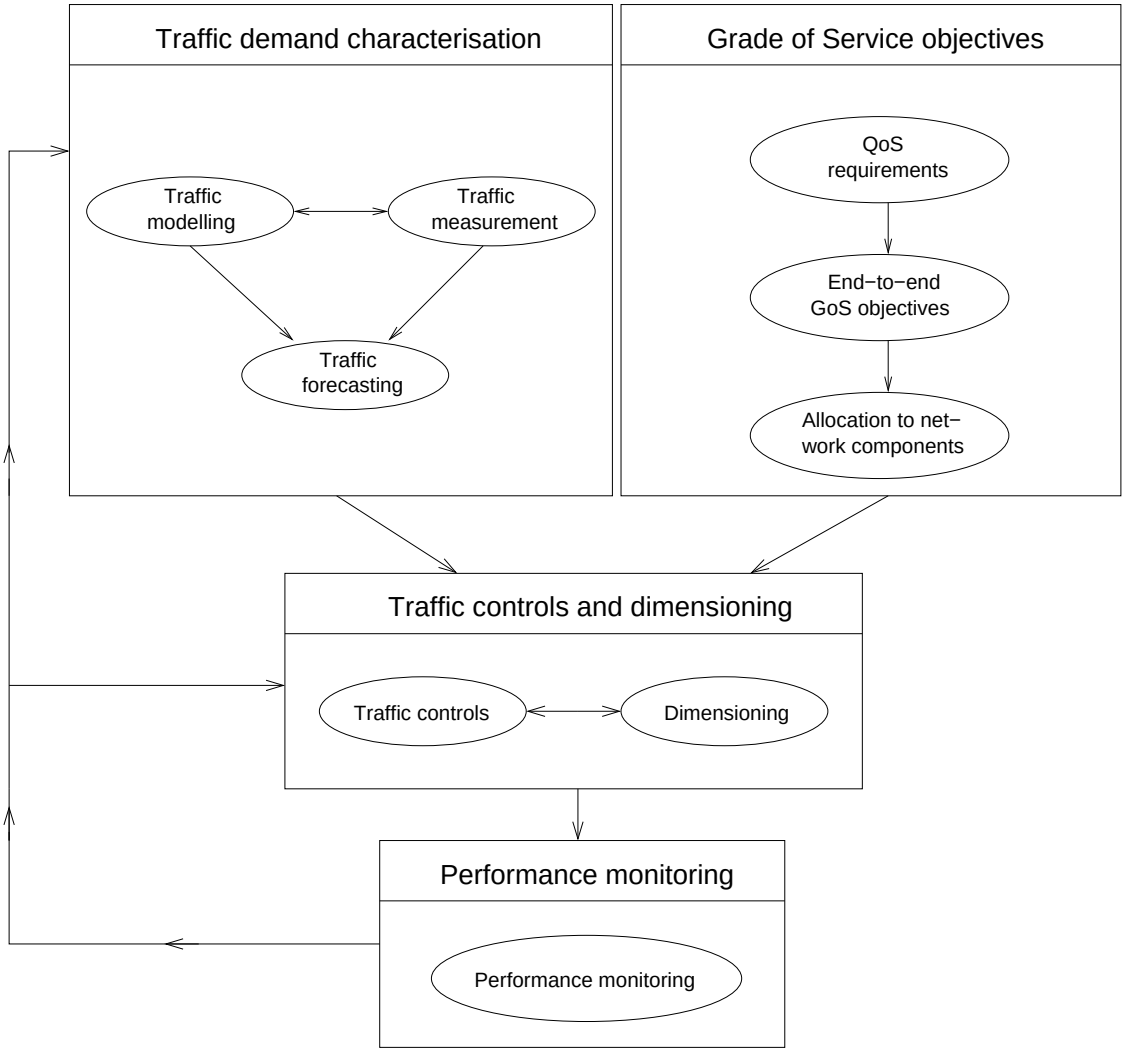


Figure 1.9: *Traffic engineering tasks.*

1.6 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 optimization 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.

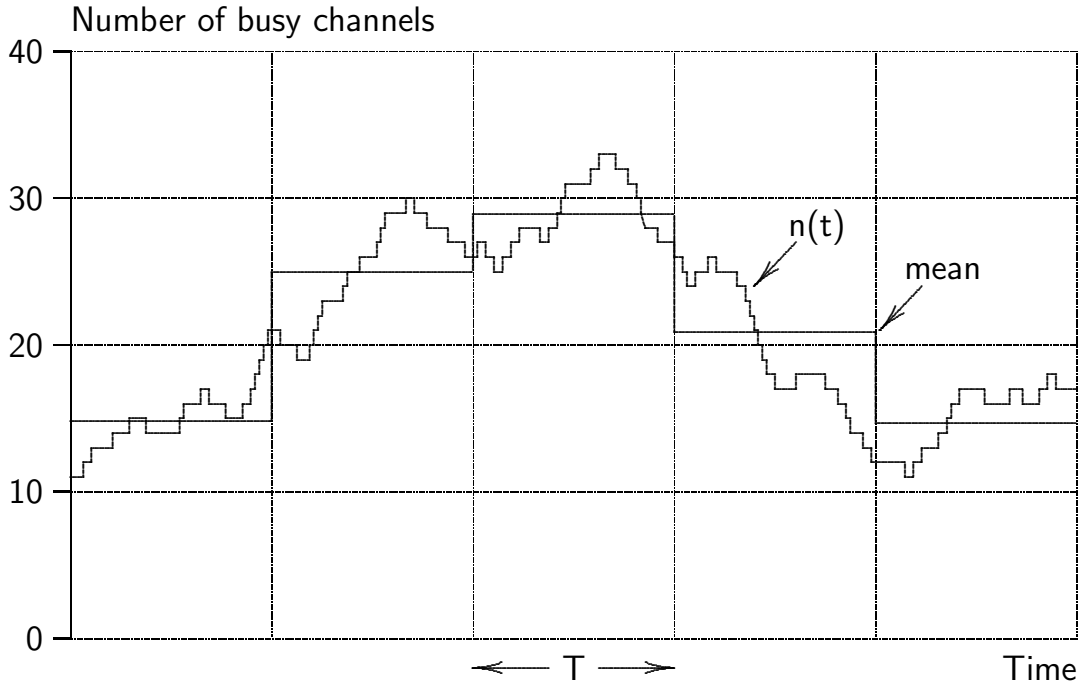


Figure 1.10: *The carried traffic (intensity) (= number of busy devices) as a function $n(t)$ of time. For dimensioning purposes we use the average traffic intensity during a period of time T (mean).*

1.7 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 [40]) 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 unit of traffic intensity is *erlang*, abbreviated E or Erl.

Depending on the technology considered, the pool of resources corresponds to a group of servers, lines, circuits, channels, trunks, computers, etc. The statistical moments (mean value, variance) of the traffic intensity may be calculated for a given period of time T. For the average traffic intensity we get:

$$Y(T) = \frac{1}{T} \cdot \int_0^T n(t)\,\mathrm{d}t. \tag{1.1}$$

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. 1.10). In applications, the term traffic intensity usually has the meaning of average traffic intensity. The carried traffic can never exceed the number of channels (lines). A channel can at most carry one erlang. The revenue is often proportional to the carried traffic.

The *ITU-T* recommendation also specifies that the unit used for traffic intensity is *erlang*. This name was given to the traffic unit in 1946 by CCIF (predecessor to *CCITT* and *ITU-T*), to honor the Danish mathematician A. K. Erlang (1878-1929), who is the founder of traffic theory in telephony. Before this name was introduced it was simply called [traffic unit], abbreviated [T.U.]. 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*), or if more convenient for example erlang–seconds. It is equal to the sum of all holding times inside the time period.

In mathematical models we use the concept *offered traffic* which is defined in the following two ways which are equivalent:

Definition of offered traffic A:

- The offered traffic is the traffic carried when no call attempts are rejected due to lack of capacity, i.e. when the number of servers is unlimited.
- The offered traffic is the average number of call attempts per mean holding time:

$$A = \lambda \cdot s \,, \tag{1.2}$$

  where λ is the call intensity, i.e. mean number of calls offered per time unit, and s is the mean service time.

The parameters should be specified using the same time unit. From this equation it is seen that the unit of traffic has no dimension.

The offered traffic is a theoretical quantity which cannot be measured. It can only be estimated from the carried traffic. A definition of offered traffic should be independent of the actual system.

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

Example 1.7.1: 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 1.7.2: 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 mathematical models and simulation models. However, the only measurable parameter in reality is the carried traffic, which depends upon the actual system.

Data transmission and multi-rate traffic: In data transmissions systems we do not talk about service times but about transmission demands. 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). The service time for such a job, i.e. transmission time, is s/φ time units (e.g. seconds), i.e. dependent upon φ. If on the average λ jobs are served per time unit, then *the utilization* ϱ of the system is:

$$\varrho = \frac{\lambda \cdot s}{\varphi}. \tag{1.3}$$

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

We split the total capacity up into units called *Basic Bandwidth Units* (*BBU*) or channels. We choose this unit so that all services require an integral number of bandwidth units, for example 64 kbps. If calls of type j simultaneously occupy d_j channels, then the offered traffic expressed in number of channels becomes:

$$A = \sum_{j=0}^{N} \lambda_j \cdot s_j \cdot d_j \quad [\text{erlang-channels}]\,, \tag{1.4}$$

where N is number of traffic types, and λ_j and s_j denotes the arrival rate, respectively the mean holding time of traffic type j. In multi-service networks both offered and carried traffic

are measured in *BBU* as the traffic is a mix of different connections with different bandwidth. The offered traffic in number of connections for one service is $A_{j,co} = \lambda_j \cdot s_j$ [erlang-connections]. But in a multi-service network it makes no sense to specify the total load by the total number of connections.

Potential traffic: In planning and demand models we use the term potential traffic, which is equal the offered traffic if there are no limitations on the use of the phone due to cost or availability (always a free telephone available).

1.8 Traffic variations and the concept busy hour

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

An investigation of the traffic variations shows that it is partly of a stochastic nature, partly of a deterministic nature. Fig. 1.11 shows the variation in the number of calls on a Monday morning. By comparing several days we can recognize a deterministic curve with superposed stochastic variations.

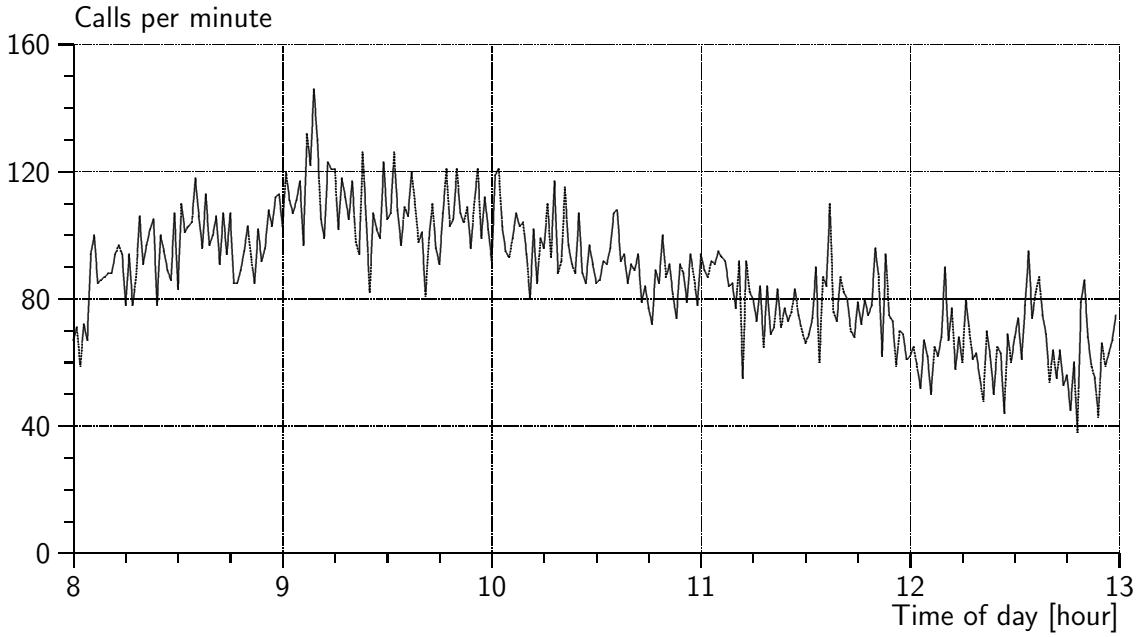


Figure 1.11: *Number of calls per minute to a switching center a Monday morning. The regular 24-hour variations are superposed by stochastic variations. (Iversen, 1973 [41]).*

During a 24 hours period the traffic typically looks as shown in Fig. 1.12. 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.

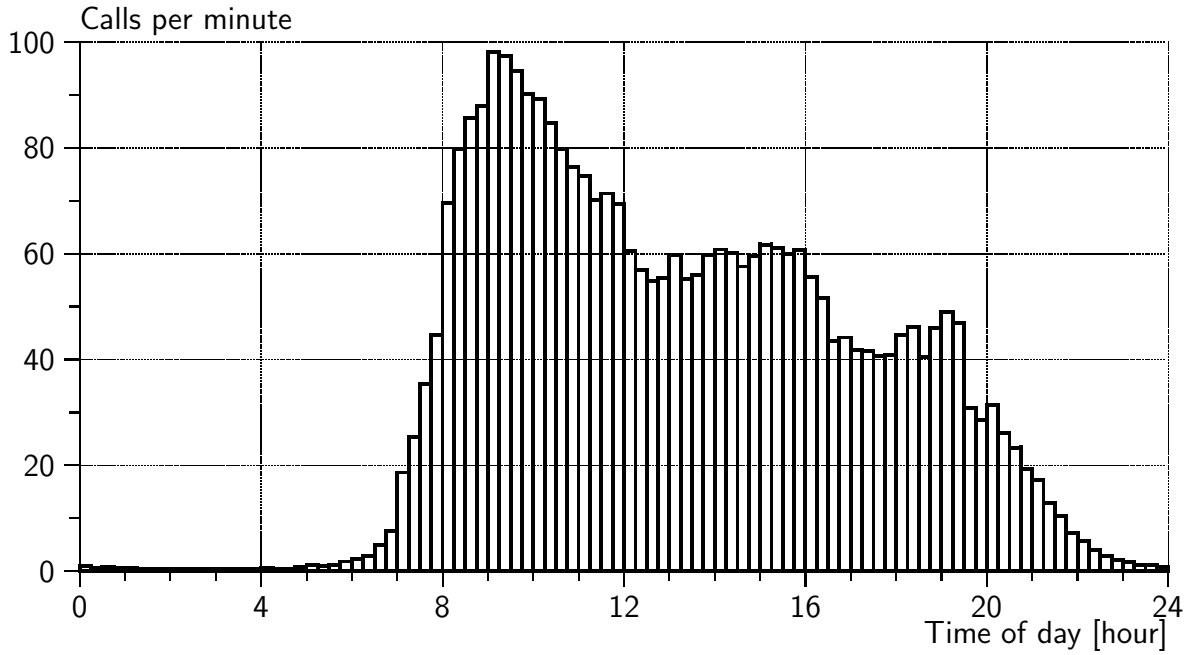


Figure 1.12: *The mean number of calls per minute to a switching center taken as an average for periods of 15 minutes during 10 working days (Monday – Friday). At the time of the measurements there were no reduced rates outside working hours (Iversen, 1973 [41]).*

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 Europe and 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. 1.13 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 (fixed with an accuracy of 15 minutes) which during a long period on the average has the highest traffic.

Some days it may therefore happen that the traffic during the *busiest hour* is larger than the time consistent busy hour, but on the average during many 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. 1.12 and 1.13).
- Weekly variations (Fig. 1.14). Normally the highest traffic is on Monday, then Friday, Tuesday, Wednesday and Thursday. Saturday and especially Sunday have a low traffic level. A useful rule of thumb is that the 24 hour traffic is equal to 8 times the busy hour traffic (Fig. 1.14), i.e. only one third of capacity in the telephone system is utilized. This is the reason for reducing rates outside 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. 1.15 we show the variation in the number of calls per 15 minutes 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. 1.16.

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 utilization of the resources.

1.9 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 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.

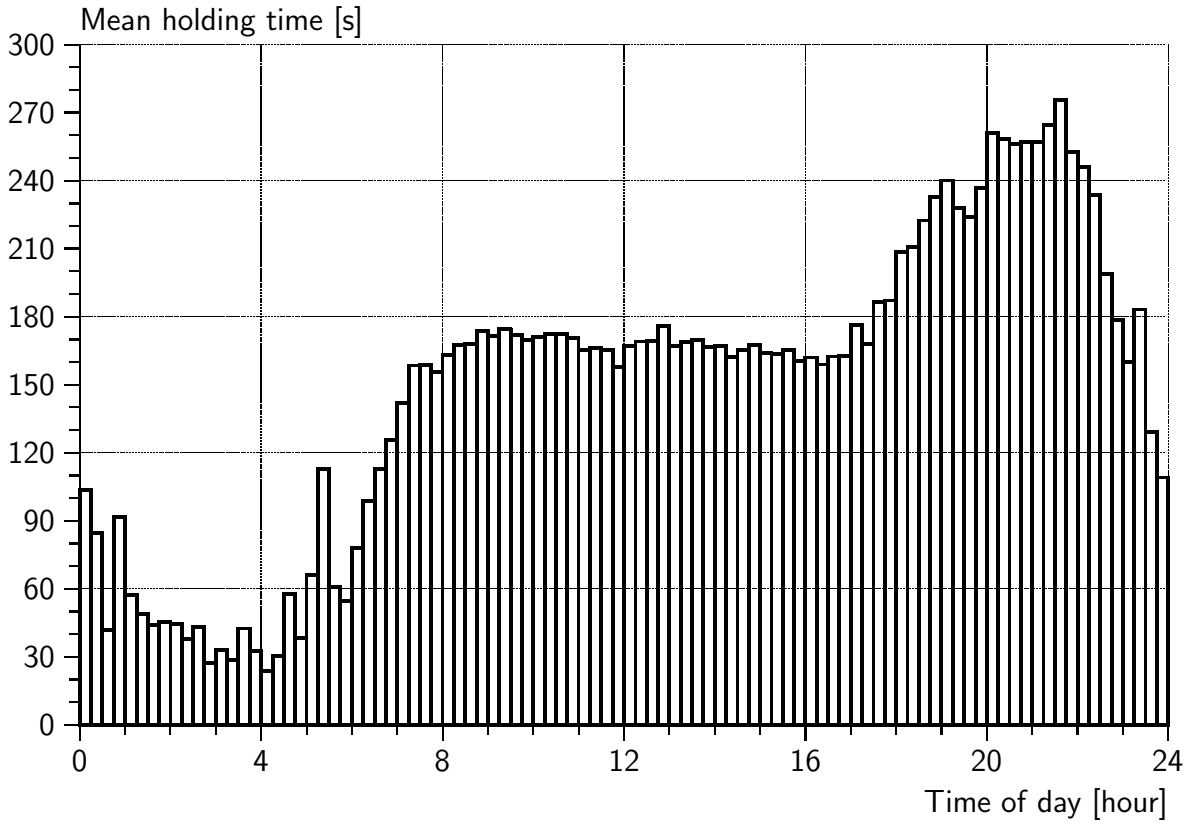


Figure 1.13: *Mean holding time for trunk lines as a function of time of day. (Iversen, 1973 [41]). The measurements exclude local calls.*

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 is *blocked* (the subscriber for example experiences busy tone and has to repeat the 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 combination 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).

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 offered traffic which is not carried, possibly despite several attempts.

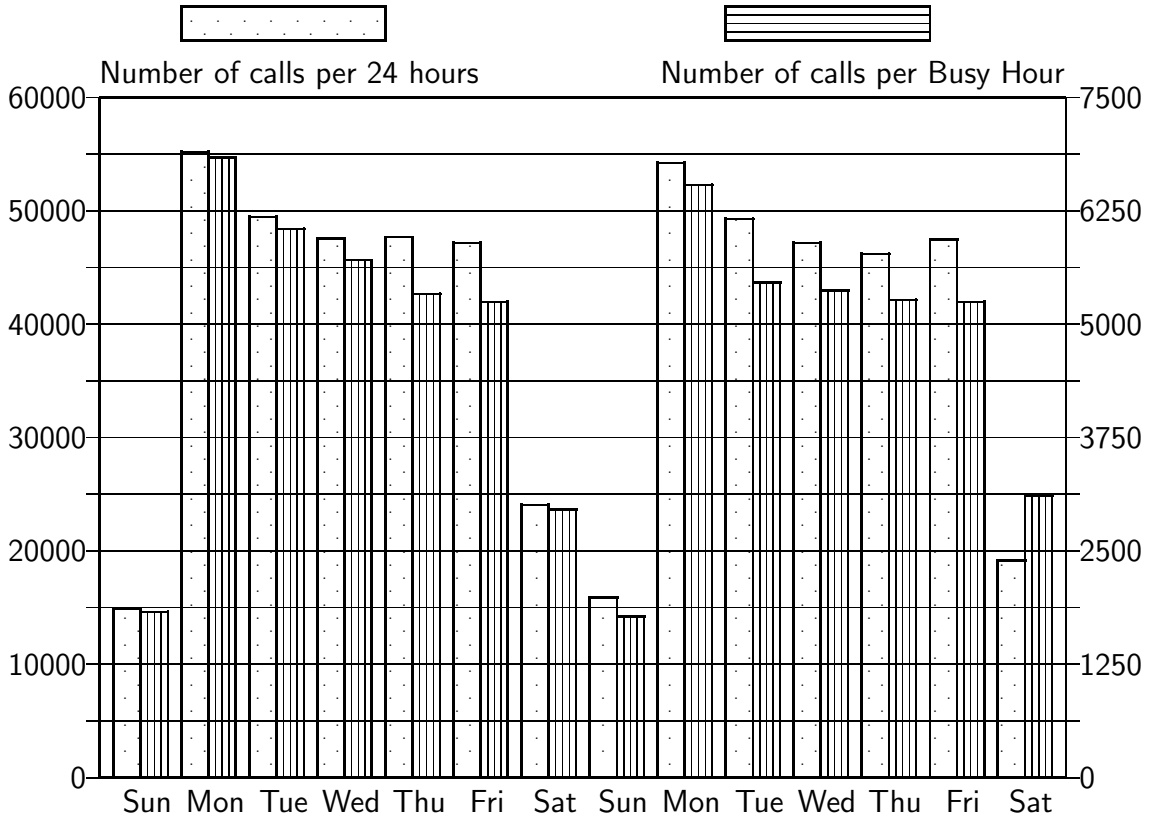


Figure 1.14: *Number of calls per 24 hours to a switching center (left scale). The number of calls during busy hour is shown for comparison at the right scale. We notice that the 24–hour traffic is approximately 8 times the busy hour traffic. This factor is called the traffic concentration (Iversen, 1973 [41]).*

These quantitative measures can for example be used to establish dimensioning principles for trunk groups.

When congestion is small it is possible with a good approximation to handle congestion in the different part of the system as being 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.

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

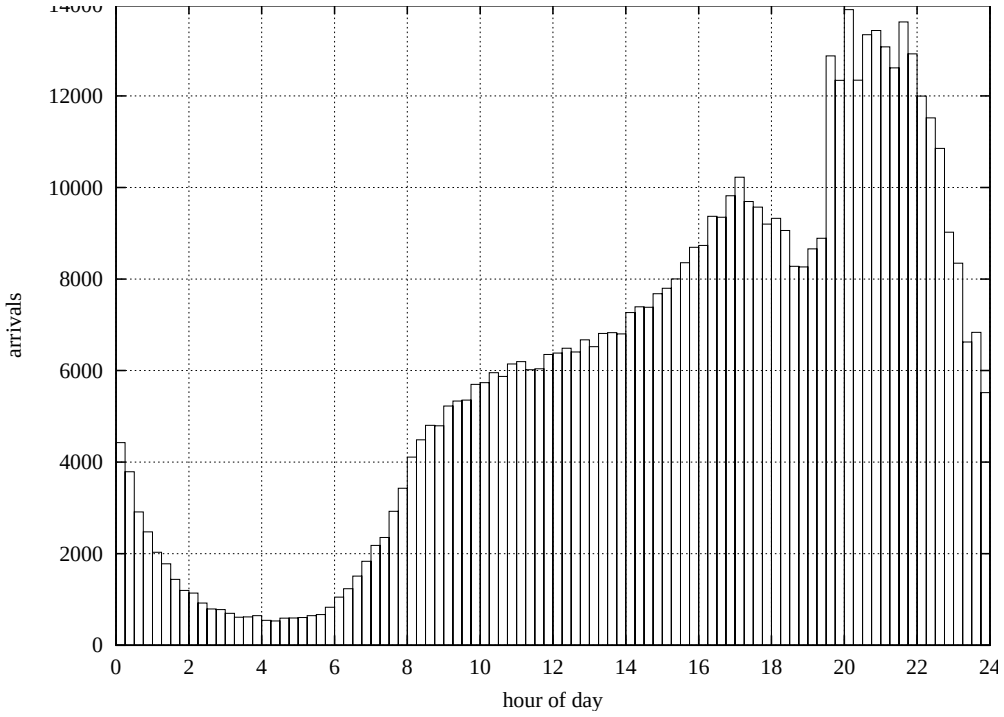


Figure 1.15: *Number of calls per 15 minutes to a modem pool of Tele Danmark Internet. Tuesday 1999.01.19.*

relation between inconvenience and waiting time.

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

1.10 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 1.1, 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 1.1 are misleading. To obtain the relevant probabilities, which are shown in Fig. 1.17, we shall only consider the calls arriving at the considered stage when calculating probabilities. Applying the notation

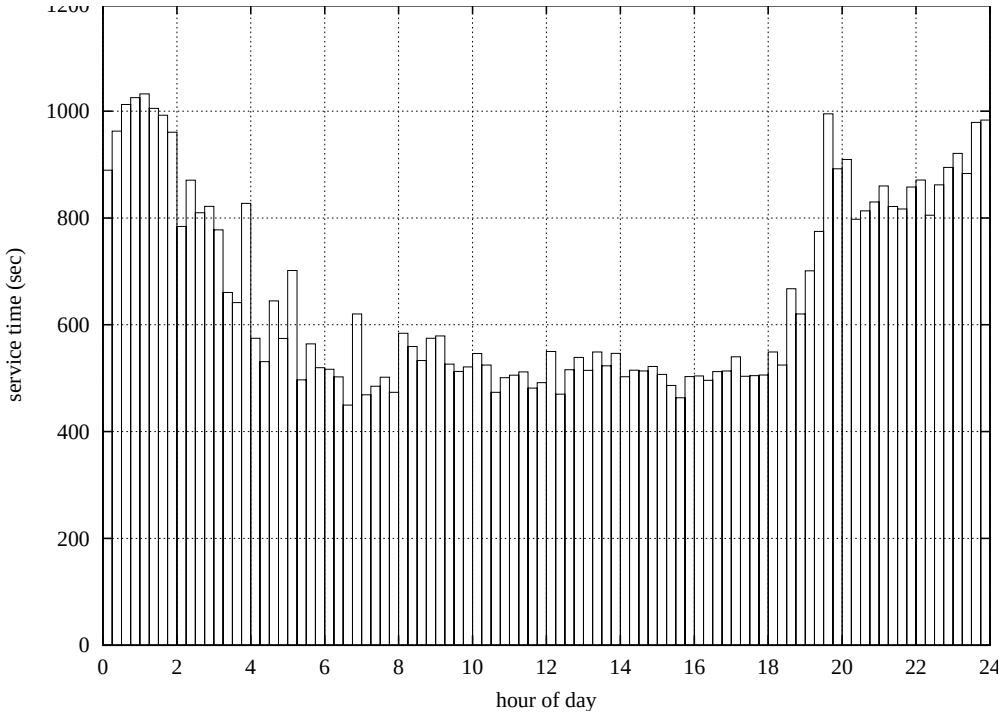


Figure 1.16: *Mean holding time in seconds as a function of time of day for calls arriving inside the period considered. Tele Denmark Internet. Tuesday 1999.01.19.*

in Fig. 1.17 we find the following probabilities for a call attempts (assuming independence):

$$p\{\text{A-error}\} = p_e \tag{1.5}$$

$$p\{\text{Congestion \& tech. errors}\} = (1-p_e)\cdot p_s \tag{1.6}$$

$$p\{\text{B–no answer}\} = (1-p_e)\cdot(1-p_s)\cdot p_n \tag{1.7}$$

$$p\{\text{B–busy}\} = (1-p_e)\cdot(1-p_s)\cdot p_b \tag{1.8}$$

$$p\{\text{B–answer}\} = (1-p_e)\cdot(1-p_s)\cdot p_a \tag{1.9}$$

Using the numbers from Table 1.1 we find the figures shown in Table 1.2. 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 mean call duration if we include all call–attempts. Fig. 1.18 shows an example with observed holding times.

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 1.1: *Typical outcome of a large number of call attempts during Busy Hour for Industrialized countries, respectively Developing countries.*

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 1.2: *The relevant probabilities for the individual outcomes of the call attempts calculated for Table 1.1*

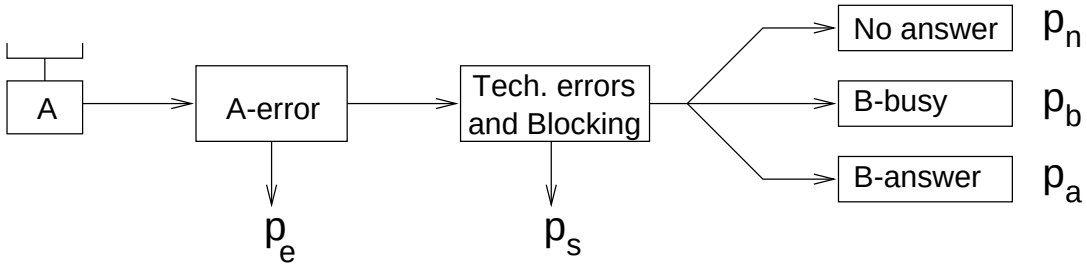


Figure 1.17: *When calculating the probabilities of events for a certain number of call attempts we have to consider the conditional probabilities.*

Example 1.10.1: 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 1.1:

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

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

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 1.4), 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 [61]) . Fig. 1.19 and Fig. 1.20 show some examples from measurements of subscriber behaviour.

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-

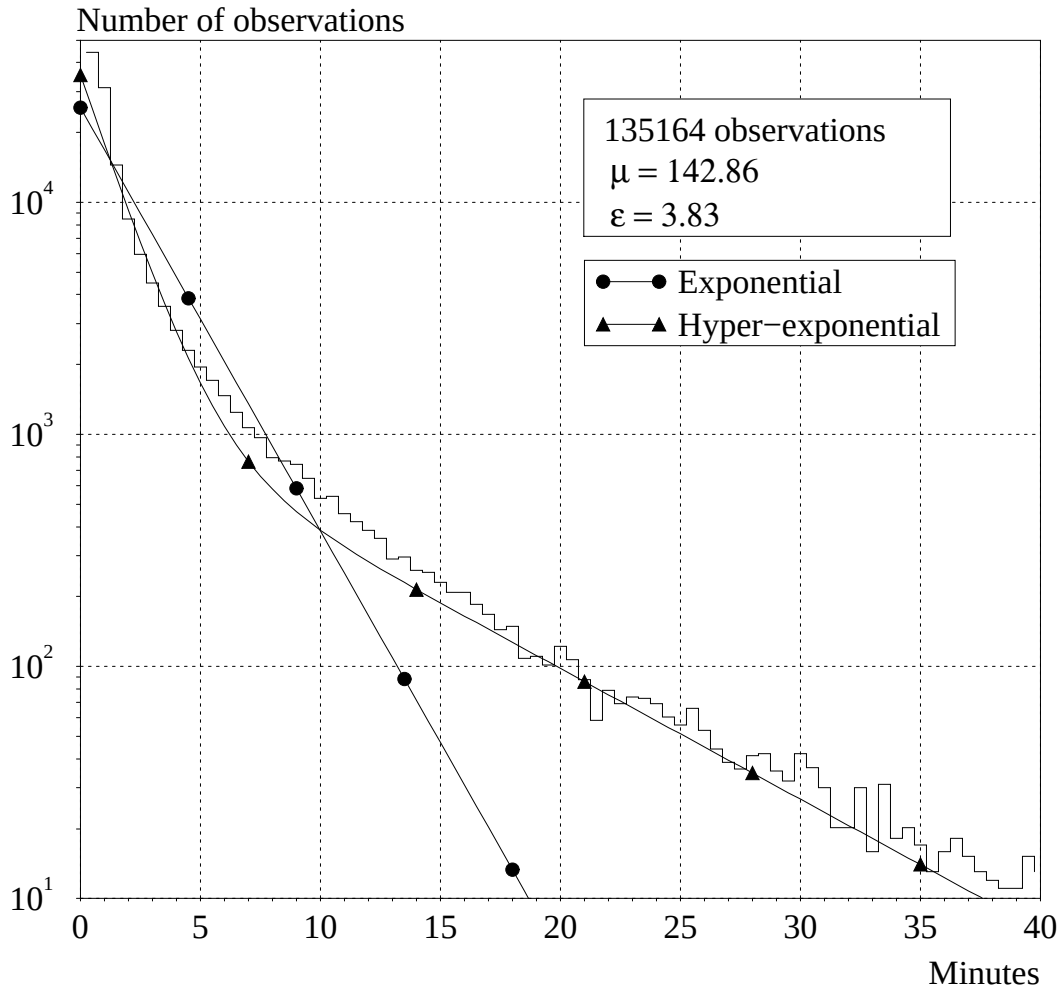


Figure 1.18: *Frequency function of holding times of trunks in a local switching center.*

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 industrialized countries the total probability of B-busy becomes of the same order of size as α (Table 1.1). 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 1.3.

	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 1.3: *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 gasworks (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. 10). 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.

Example 1.10.2: Repeated call attempt
This is an example of a simple model of repeated call attempts. Let us introduce the following

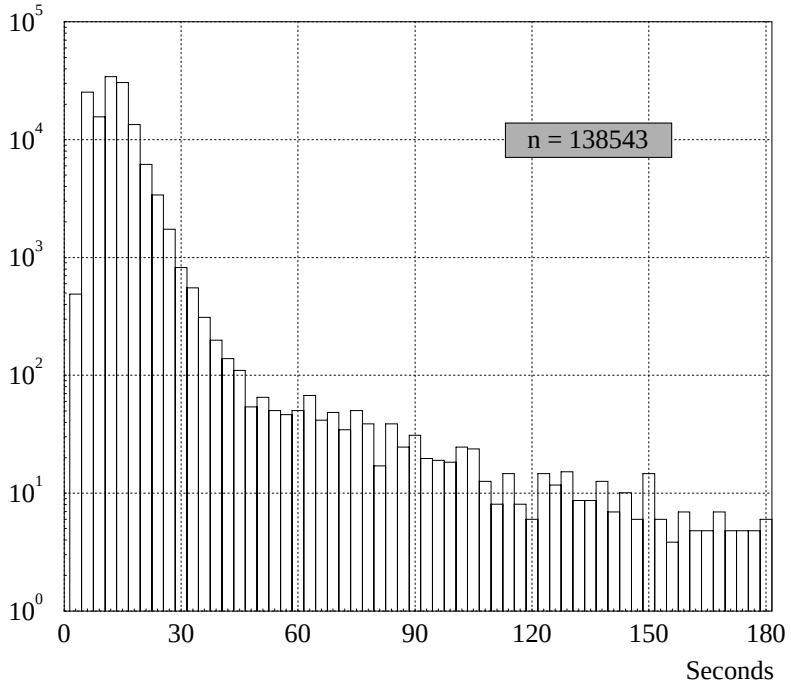


Figure 1.19: *Histogram for the time interval from occupation of register (dial tone) to B–answer for completed calls. The mean value is 13.60 s.*

notation:

$$b = \text{persistence} \tag{1.10}$$

$$B = \text{p\{non-completion\}} \tag{1.11}$$

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:

$$\text{p\{completion\}} = \frac{(1-B)}{(1-B\cdot b)} \tag{1.12}$$

$$\text{p\{non-completion\}} = \frac{B\cdot(1-b)}{(1-B\cdot b)} \tag{1.13}$$

$$\text{No. of call attempts per call intent} = \frac{1}{(1-B\cdot b)} \tag{1.14}$$

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

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	$\frac{(1-B)}{(1-B \cdot b)}$	$\frac{1}{(1-B \cdot b)}$	$\frac{B \cdot (1-b)}{(1-B \cdot b)}$

Table 1.4: *A single call intent results in a series of call attempts. The distribution of the number of attempts is geometrically distributed.*

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

$$Y = A \cdot \frac{1-B}{1-B \cdot b} \tag{1.15}$$

$$A = Y \cdot \frac{1-B \cdot b}{1-B} \tag{1.16}$$

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 1.3). If the unsuccessful calls have a positive mean holding time, then the carried traffic may become larger than the offered traffic.

1.11 Introduction to Grade-of-Service = GoS

The following section is based on (Veirø, 2001 [117]). 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).

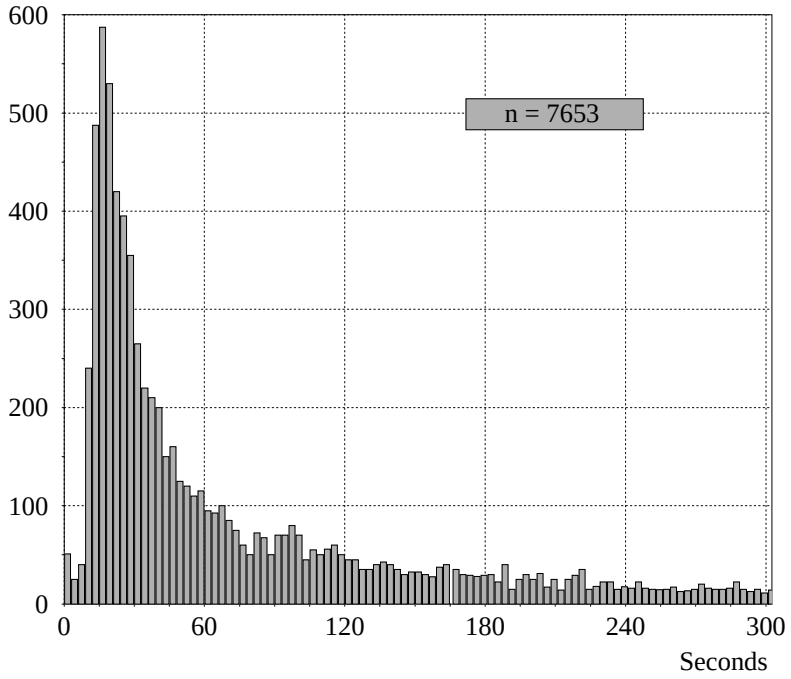


Figure 1.20: *Histogram for all call attempts repeated within 5 minutes, when the called party is busy.*

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 summarized 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 recommendation. 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 resources 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.

1.11.1 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 1.11.1:
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.

1.11.2 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.

1.11.3 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 organization 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.

1.11.4 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. 1.21.

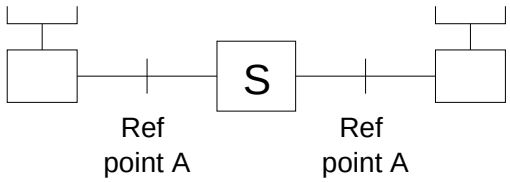


Figure 1.21: *Reference configuration for case 1.*

2. terminal → subscriber switch → transit switch → subscriber switch → terminal
   This is drawn as a reference configuration shown in Fig. 1.22.

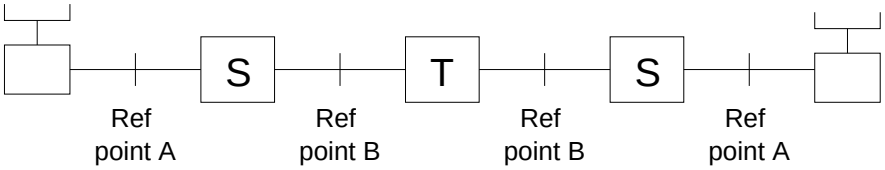


Figure 1.22: *Reference configuration for case 2.*

3. terminal → subscriber switch → transit switch → transit switch → subscriber switch → terminal
   This is drawn as a reference configuration shown in Fig. 1.23.

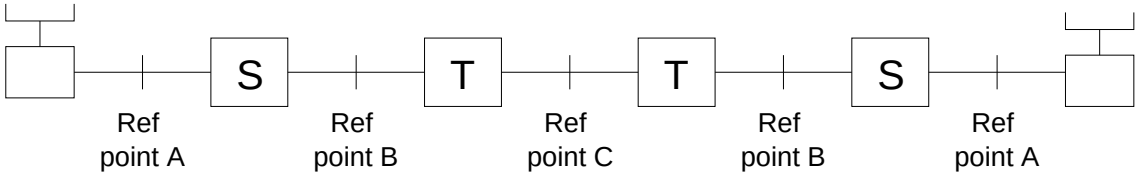


Figure 1.23: *Reference configuration for case 3.*

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 ITU-TRecommendation 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 2

Time interval modeling

Time intervals are non-negative, and therefore they can be expressed by non-negative *random variables*. 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 *life-times* and their distribution functions as *time distributions*. In this chapter we review the basic theory of probability and statistics relevant to teletraffic theory and illustrate the theory by the *(negative) exponential distribution* and generalizations of this. In this chapter we describe the continuous distributions which are most important in teletraffic. In next chapter we deal with the important discrete distributions.

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. The most fundamental characteristic of the exponential distribution is the *Markov property* which means *lack of memory* or *lack of age*. The future is independent of the past.

We can combine life-times in series (Sec. 2.3.1), in parallel (Sec. 2.3.2), or in a combination of the two (Sec. 2.4). In this way we get more parameters available for fitting the distribution to real observations. A *hypo-exponential or steep distribution* corresponds to a set of stochastic independent exponential distributions in series (Fig. 2.4), and a *hyper-exponential or flat distribution* corresponds to exponential distributions in parallel (Fig. 2.6). This structure corresponds naturally to the shaping of traffic processes in telecommunication and data networks. By combination of steep and flat distribution we get Cox-distributions, which can be approximated to any observed distribution with any degree of accuracy. By using a graphical approach, *phase-diagrams*, we are able to derive decomposition properties of importance for later applications.
We also mention a few other time distributions which are employed in teletraffic theory (Sec. 2.5), and finally we review some observations of real life-times in Sec. 2.6.

2.1 Distribution functions

A time interval can be described by a random variable T. This is characterized by a *cumulative distribution function* (*cdf*) denoted by $F(t)$, which is the probability that the duration of a time interval is less than or equal to t:

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

In general, we assume that the derivative of $F(t)$, *the probability density function* (*pdf*) denoted by $f(t)$, exists:

$$\mathrm{d}F(t) = f(t) \cdot \mathrm{d}t = p\{t < T \leq t + \mathrm{d}t\}\,, \qquad t \geq 0\,. \tag{2.1}$$

As we only consider non-negative time intervals we have:

$$F(t) = \begin{cases} 0\,, & t < 0\,, \\ \displaystyle\int_{0-}^{t} \mathrm{d}F(u) = \int_{0-}^{t} f(u)\,\mathrm{d}u\,, & 0 \leq t < \infty\,, \end{cases} \tag{2.2}$$

In (2.2) we integrate from $0-$ to keep record of a possible discontinuity in $t = 0$. When we consider waiting time systems, there is often a positive probability of having waiting times equal to zero, i.e. $F(0) > 0$. On the other hand, when we look at the inter-arrival times, we usually assume $F(0) = 0$ (Sec. 3.2.3). The probability density function is also called the *frequency function*.

Sometimes it is easier to consider the *complementary distribution function*, also called the *survival distribution function*:

$$F^c(t) = 1 - F(t)\,. \tag{2.3}$$

Analytically, many calculations can be carried out for any time distribution.

2.1.1 Exponential distribution

This is the most fundamental distribution in teletraffic theory, where it is called *the negative exponential distribution*.

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

$$F(t) = 1 - \mathrm{e}^{-\lambda t}\,, \quad \lambda > 0\,, \quad t \geq 0\,, \tag{2.4}$$

$$f(t) = \lambda \mathrm{e}^{-\lambda t}\,, \qquad \lambda > 0\,, \quad t \geq 0\,. \tag{2.5}$$

The phase diagram of the exponential distribution is shown in Fig. 2.1. The density function is shown in Fig. 2.5 for $k = 1$.

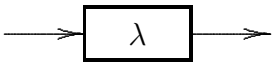


Figure 2.1: *Phase diagrams of an exponentially distributed time interval is shown as a box with the intensity λ. The box thus means that a customer arriving to the box is delayed an exponentially distributed time interval with mean value λ^{-1} before leaving the box.*

2.2 Characteristics of distributions

Times intervals are always non-negative and therefore their distribution functions have some useful properties.

2.2.1 Moments

The i'th *non-central moment*, which usually is called the i'th moment, is defined by:

$$E\{T^i\} = m_i = \int_0^\infty t^i \cdot f(t)\,\mathrm{d}t\,, \quad i = 1, 2, \ldots\,. \tag{2.6}$$

So far we assume that all moments exist. In general we always assume that at least the mean value exists. A distribution is uniquely defined by its moments. For life-time distributions we have the following relation, called *Palm's identity*:

$$m_i = \int_0^\infty t^i \cdot f(t)\,\mathrm{d}t = \int_0^\infty i \cdot t^{i-1} \cdot \{1 - F(t)\}\,\mathrm{d}t\,, \quad i = 1, 2, \ldots\,. \tag{2.7}$$

It was first proved by (Palm, 1943 [94]) as follows:

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

The order of integration can be inverted because the integrand is non-negative. Thus we have proved (2.6). In particular we find the first two moments:

$$m_1 = \int_0^\infty t \cdot f(t)\,\mathrm{d}t = \int_0^\infty \{1 - F(t)\}\mathrm{d}t = E\{T\}\,, \tag{2.8}$$

$$m_2 = \int_0^\infty t^2 \cdot f(t)\,\mathrm{d}t = \int_0^\infty 2t \cdot \{1 - F(t)\}\,\mathrm{d}t\,. \tag{2.9}$$

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

$$E\{(T - m_1)^i\} = \int_0^\infty (t - m_1)^i \cdot f(t)\,\mathrm{d}t\,. \tag{2.10}$$

In advanced teletraffic we also use *cumulants*, *Binomial moments*, and *factorial moments*. They are uniquely related to the above moments, but has some advantages when dealing with special problems, for example overflow systems (Chapter 6).

For characterizing random variables we use the following parameters related to the first two moments:

- Mean value or expected value = $E\{T\}$. This is the first moment:

$$m_1 = E\{T\}\,. \tag{2.11}$$

- *Variance*. This is the *2nd central moment:*

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

  It is easy to show that:

$$\sigma^2 = m_2 - m_1^2 \qquad \text{or} \tag{2.12}$$

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

- *Standard deviation*. This is the square root of the variance and thus equal to σ.

- *Coefficient of variation* is a normalized measure for the irregularity (dispersion) of a distribution. It is defined as the ratio between the standard deviation and the mean value:

$$CV = \text{Coefficient of Variation} = \frac{\sigma}{m_1}\,. \tag{2.13}$$

  This quantity is dimensionless, and later we use it to characterize discrete distributions (state probabilities).

- *Palm's form factor* ε is another normalized second order measure of irregularity which is defined as follows:

$$\varepsilon = \frac{m_2}{m_1^2} = 1 + \left(\frac{\sigma}{m_1}\right)^2 \geq 1\,. \tag{2.14}$$

The form factor ε as well as $(\sigma/m_1 = CV)$ 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, The form factor has its minimum value equal to one for constant time intervals ($\sigma = 0$). It is used to characterize continuous distributions, for example time intervals.

- *Median.* Sometimes we also use the *median* to characterize a distribution. The median is the value of t for which $F(t) = 0.5$. Thus half the observations will be smaller than the median and half will be bigger. For a symmetric probability density function

6. *Index of Dispersion for Intervals, IDI.*
The *Index of Dispersion for Intervals, IDI* is defined as:

$$IDI = \frac{\mathrm{Var}\{T_i\}}{\mathrm{E}\{T_i\}^2} = \varepsilon - 1\,, \tag{2.15}$$

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 (2.14).the mean equals the median. For the exponential distribution the median is 0.6931 times the mean value.

- *Percentiles.* More generally, we characterize a distribution by *percentiles* (*quantiles* or *fractiles*): If

$$P(T \leq t_p) = p_t\,,$$

then t_p is the $p_t \cdot 100$ % percentile. The median is the 50% percentile.

When estimating parameters of a distribution from observations, we are usually satisfied by knowing the first two moments (m_1 and σ) as higher order moments require extremely many observations to obtain reliable estimates.

Time distributions can also be characterized in other ways, for example by properties related to the traffic. We consider some important characteristics in the following sections.

Example 2.2.1: Exponential distribution
The following integral is very useful:

$$\int t \cdot \mathrm{e}^{-\lambda t}\mathrm{d}t = -\frac{\mathrm{e}^{-\lambda t}}{\lambda^2}\,(\lambda t + 1) \tag{2.16}$$

For the exponential distribution (Sec. 2.1.1) we find:

$$\begin{aligned} m_1 &= \int_0^\infty t \cdot \lambda \, \mathrm{e}^{-\lambda t} \, \mathrm{d}t = \frac{1}{\lambda} \,, \\ m_2 &= \int_0^\infty t^2 \cdot \lambda \, \mathrm{e}^{-\lambda t} \, \mathrm{d}t \\ &= \int_{t=0}^\infty 2\, t \cdot \mathrm{e}^{-\lambda t} \, \mathrm{d}t = \frac{2}{\lambda^2} \,. \end{aligned}$$

where the last equation is obtained using (2.7). The gamma function is defined by:

$$\Gamma(n+1) = \int_0^\infty t^n \cdot \mathrm{e}^{-t} \, \mathrm{d}t = n! \tag{2.17}$$

If we replace t by λt, then we get the i'th moment (2.7) of the exponential distribution:

$$\begin{aligned} &i\text{'th moment:} && m_i &&= \frac{i\,!}{\lambda^i} \,, \\ &\text{Mean value:} && m_1 &&= \frac{1}{\lambda} \,, \\ &\text{Second moment:} && m_2 &&= \frac{2}{\lambda^2} \,, \\ &\text{Variance:} && \sigma^2 &&= \frac{1}{\lambda^2} \,, \\ &\text{Form factor:} && \varepsilon &&= 2 \,. \end{aligned} \tag{2.18}$$

□

Example 2.2.2: Constant time interval
For a constant time interval of duration h we have: $m_i = h^i$. □

2.2.2 Residual life-time

If an event b has occurred, i.e. $p(b > 0)$, then the probability that the event a also occurs is given by the *conditional probability*. This is denoted by $p(a \mid b)$, the conditional probability of a, given b. Denoting the joint probability that both a and b take place by $p(a \cap b)$ we have:

$$p(a \cap b) = p(b) \cdot p(a \mid b) = p(a) \cdot p(b \mid a) \,.$$

Under the assumption that $p(b) > 0$, we thus have:

$$p(a \mid b) = \frac{p(a \cap b)}{p(b)} \,. \tag{2.19}$$

For time distributions we are interested in $F(x + t \mid x)$, the distribution of the residual life time t, given that a certain age $x \geq 0$ has already been obtained. The random variable of the total life time is T.

Assuming $p\{T>x\} > 0$ and $t \geq 0$ we get:

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

and thus:

$$\begin{aligned} F(x+t \mid x) &= p\{T \leq x+t \mid T > x\} \\ &= \frac{F(x+t) - F(x)}{1 - F(x)}\,, \end{aligned} \tag{2.20}$$

$$f(t+x \mid x) = \frac{f(x+t)}{1 - F(x)}\,, \quad t \geq 0\,, \quad x \geq 0\,. \tag{2.21}$$

Fig. 2.2 illustrates these calculations graphically.

The mean value $m_{1,r}$ of the residual life-time can be written as (2.8):

$$m_{1,r}(x) = \frac{1}{1 - F(x)} \cdot \int_{t=0}^{\infty} \{1 - F(x+t)\}\,\mathrm{d}t\,, \qquad x \geq 0\,. \tag{2.22}$$

The *death rate at time* x, i.e. the probability, that the considered life-time terminates within an interval $(x, x+\mathrm{d}x)$, under the condition that age x has been achieved, is obtained from (2.20) by letting $t = \mathrm{d}x$:

$$\begin{aligned} \mu(x) \cdot \mathrm{d}x &= \frac{F(x+\mathrm{d}x) - F(x)}{1 - F(x)} \\ &= \frac{\mathrm{d}F(x)}{1 - F(x)} \qquad (2.23) \\ &= \frac{f(x)\,\mathrm{d}x}{1 - F(x)}\,. \qquad (2.24) \end{aligned}$$

The conditional density function $\mu(x)$ is also called the *hazard function*. If this function is given by (2.23), then $F(x)$ may be obtained as the solution to the following differential equation:

$$\frac{\mathrm{d}F(x)}{\mathrm{d}x} + \mu(x) \cdot F(x) - \mu(x) = 0\,.$$

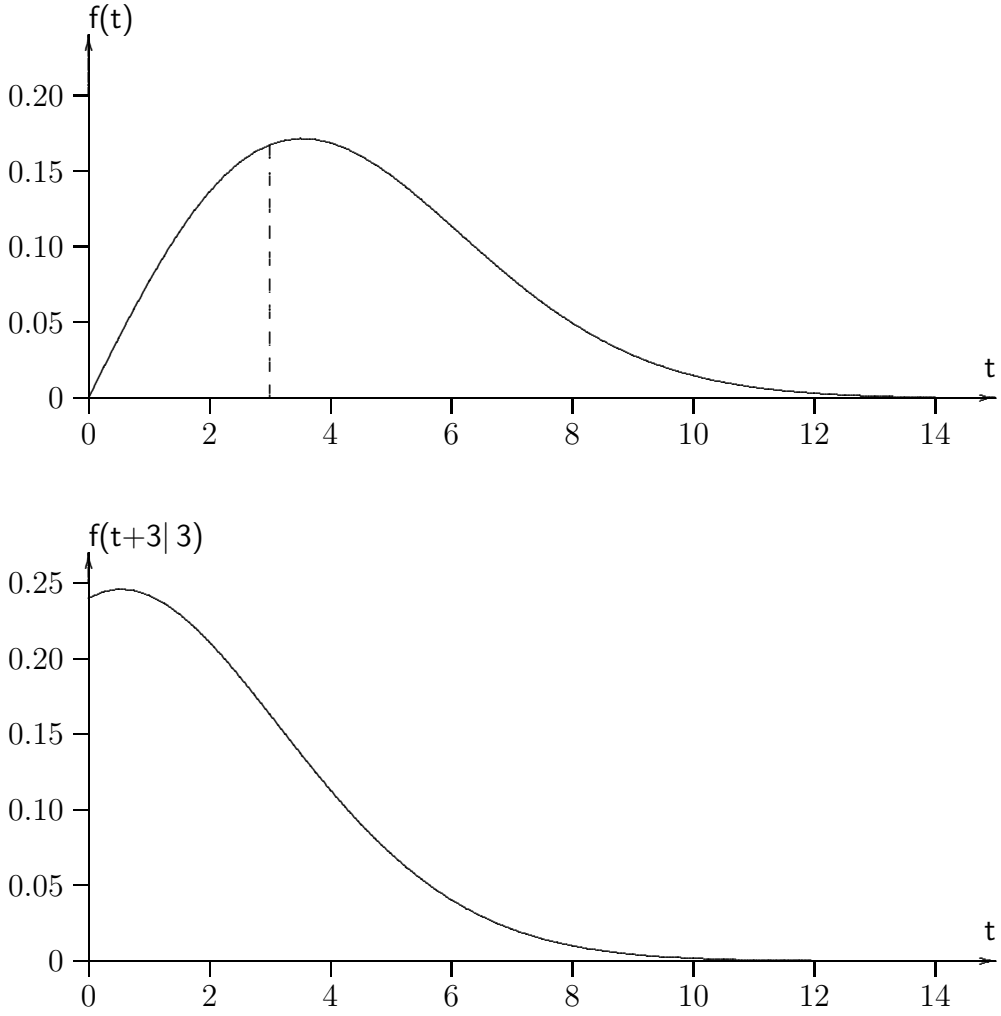


Figure 2.2: *The density function of the residual life time conditioned by a given age x (2.21). The example is based on a Weibull distribution We(2,5) (2.101), where $x = 3$ and $F(3) = 0.3023$.*

Assuming $F(0) = 0$ we get the solution:

$$F(t) = 1 - \exp\left\{-\int_0^t \mu(u)\,\mathrm{d}u\right\}, \tag{2.25}$$

$$f(t) = \mu(t) \cdot \exp\left\{-\int_0^t \mu(u)\,\mathrm{d}u\right\}. \tag{2.26}$$

The death rate $\mu(t)$ is constant if and only if the life-time is exponentially distributed. This is a fundamental characteristic of the exponential distribution which is called the Markovian property (*lack of memory or age*). The probability of terminating at time t is independent of the actual age t.

One would expect that the *mean residual life-time* $m_{1,r}(x)$ (2.22) decreases for increasing x, so that the expected *residual life-time* decreases when the age x increases. Often it is not so.

For an exponential distribution with form factor $\varepsilon = 2$ we have $m_{1,r}(x) = m_1$ as it has no memory. For steep distributions ($1 \leq \varepsilon < 2$) we have $m_{1,r}(x) < m_1$ (Sec. 2.3.1), whereas for flat distributions ($2 < \varepsilon < \infty$), we have $m_{1,r}(x) > m_1$ (Sec. 2.3.2).

Example 2.2.3: Exponential distribution
We assume duration of telephone calls is exponentially distributed. The distribution of the residual time is then independent of the actual duration of the conversation, and it is equal to the distribution of the total life-time (2.21):

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

If we remove the probability mass of the interval $(0, x)$ from the density function and normalize 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. Therefore, the mean value of the residual life-time is $m_{1,r}(x) = m_1$, and the probability of observing a life–time in the interval $(x, x + \mathrm{d}t)$, given that it has obtained the age x, is given by (2.23)

$$\begin{aligned} p\{x < X \leq x + \mathrm{d}t \mid X > t\} &= \frac{f(x)\,\mathrm{d}t}{1 - F(x)} \\ &= \lambda\,\mathrm{d}t\,. \end{aligned} \tag{2.27}$$

Thus it depends only upon λ and $\mathrm{d}t$, but it is independent of the actual age x. An example where this property is not valid is shown in Fig. 2.2 for the Weibull distribution (2.101) when $k \neq 1$. For $k = 1$ the Weibull distribution becomes identical with the exponential distribution. □

Example 2.2.4: Waiting-time distribution
Let us consider a queueing system with infinite queue where no customers are blocked. 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 are served immediately without any delay. We thus have $W_s(0) > 0$. The waiting time distribution $W_+(t)$ for customers having positive waiting times then becomes (2.20):

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

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

$$D \cdot \{1 - W_+(t)\} = 1 - W_s(t)\,. \tag{2.28}$$

For the probability density function (pdf) we have (2.21):

$$D \cdot w_+(t) = w_s(t)\,. \tag{2.29}$$

For mean values we get:

$$D \cdot w = W\,, \tag{2.30}$$

where the mean waiting time for all customers is denoted by W, and the mean waiting time for the delayed customers is denoted by w. These formulæ are valid for any queueing system with infinite queue. □

2.2.3 Load from holding times of duration less than x

So far we have attached the same importance to all life-times independently of their duration. The importance of a life-time 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 to a life time allocate a weight factor proportional to its duration, then the average weight of all time intervals is equal to the mean value:

$$m_1 = \int_0^\infty t \cdot f(t)\,\mathrm{d}t\,, \tag{2.31}$$

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

We are interested in calculating the proportion of the mean value which is due to contributions from life-times of duration less than x:

$$\varrho_x = \frac{\int_0^x t \cdot f(t)\,\mathrm{d}t}{m_1}\,. \tag{2.32}$$

Often relatively few service times make up a relatively large proportion of the total load. From Fig. 2.3 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 utilized to give priority to short tasks without delaying long tasks very much (Chap. 10).

Example 2.2.5: Exponential distribution
For the exponentially distributed jobs with mean value $m_1 = 1/\lambda$ we find the relative load from jobs of duration $t \le x$ from (2.32), using (2.16):

$$\begin{aligned} \varrho_x &= \lambda \cdot \int_0^x t \cdot f(t)\,\mathrm{d}t \\ &= \int_0^x \lambda t \cdot \lambda\,\mathrm{e}^{-\lambda t}\,\mathrm{d}t \\ &= 1 - \mathrm{e}^{-\lambda x}(\lambda x + 1)\,. \end{aligned} \tag{2.33}$$

This result is used later when we look at shortest-job-first queueing discipline (Sec. 10.6.4). □

2.2.4 Forward recurrence time

The residual life-time from a random point of time is called the *forward recurrence time*. In this section we shall derive some formulæ of importance for applications. To formulate the

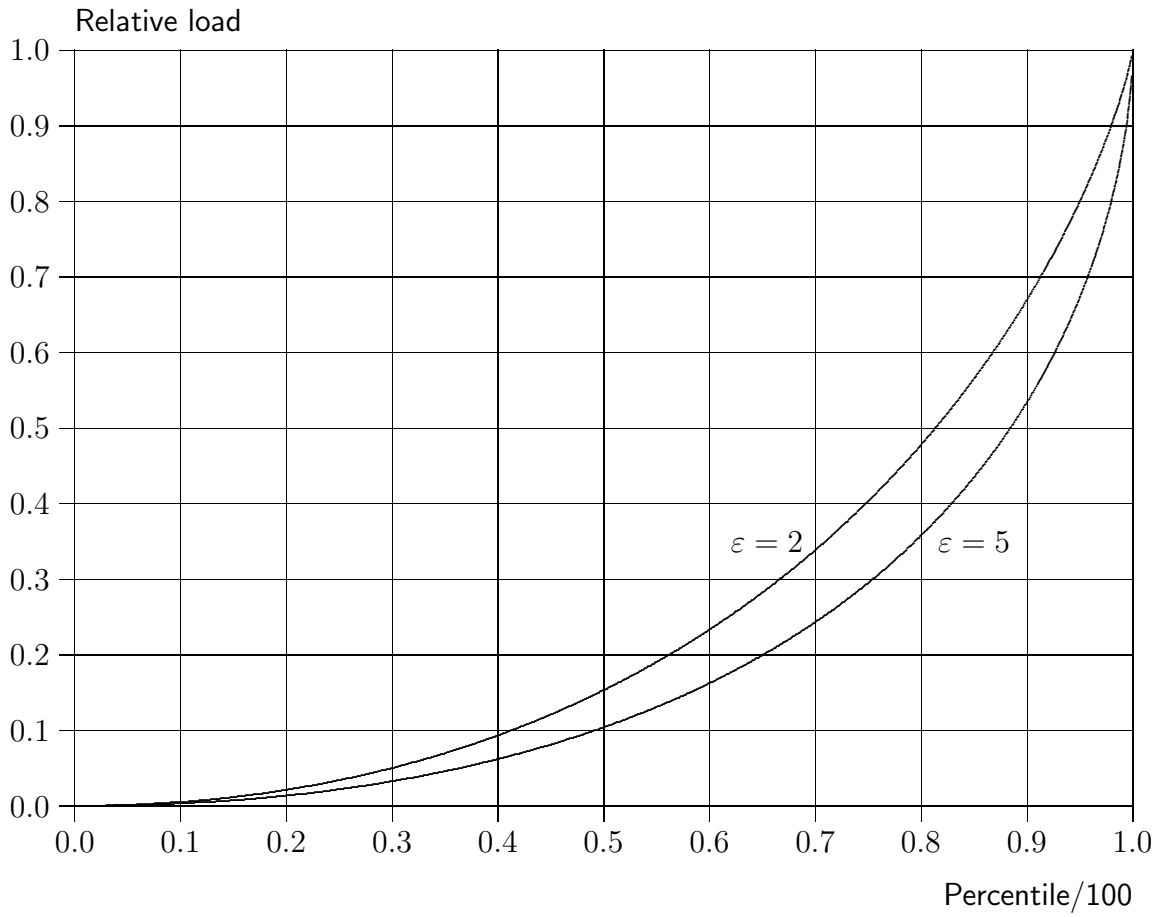


Figure 2.3: *Example of the relative traffic load from holding times shorter than a given value given by the percentile of the holding time distribution (2.32). Here $\varepsilon = 2$ corresponds to an exponential distribution and $\varepsilon = 5$ corresponds to a Pareto-distribution. We note that the 10% largest holding times contributes with 33%, respectively 47%, of the load (cf. customer averages and time averages in Chap. 3).*

problem we consider an example. We wish to investigate the life-time 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 the probability of choosing a certain car is proportional to the total life-time of that car. The distribution of the remaining residual life-time will be identical with the already achieved life-time.

By sampling in this way, the probability of choosing a car is proportional with the life-time of this car, i.e. we will preferably choose cars with longer life-times (length-biased sampling). The probability of choosing a car having a total life-time x is given by (cf. the derivation of (2.32)):

$$\frac{x \cdot f(x)\,\mathrm{d}x}{m_1}\,.$$

As we consider a random point of time, the distribution of the remaining life-time will be

uniformly distributed in $(0, x\,]$:

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

The probability density function (pdf) of the remaining life-time at a random point of time becomes:

$$\begin{aligned} v(t) &= \int_t^\infty \frac{1}{x} \cdot \frac{x \cdot f(x)\,\mathrm{d}x}{m_1}\,, \\ v(t) &= \frac{1 - F(t)}{m_1}\,. \end{aligned} \tag{2.34}$$

where $F(t)$ is the distribution function of the total life-time and m_1 is the mean value. By applying the identity (2.6), we note that the i'th moment of $v(t)$ is given by the $(i+1)$'th moment of $f(t)$:

$$\begin{aligned} m_{i,v} &= \int_0^\infty t^i \cdot v(t)\,\mathrm{d}t \\ &= \int_0^\infty t^i \cdot \frac{1 - F(t)}{m_1}\,\mathrm{d}t \\ &= \frac{1}{i+1} \cdot \frac{1}{m_1} \cdot \int_0^\infty (i+1)\,t^i \cdot \{1 - F(t)\}\,\mathrm{d}t\,, \\ m_{i,v} &= \frac{1}{i+1} \cdot \frac{1}{m_1} \cdot m_{i+1,f}\,. \end{aligned} \tag{2.35}$$

In particular, we obtain the mean value:

$$m_{1,v} = \frac{m_1}{2} \cdot \varepsilon\,, \tag{2.36}$$

where m_1 is the mean value and ε the form factor of the life-time distribution considered. These formulæ are also valid for discrete time distributions.

Example 2.2.6: Exponential distribution
For the exponential distribution we get (2.4):

$$m_{i,v} = \frac{1}{i+1} \cdot \frac{1}{m_1} \cdot \frac{(i+1)!}{\lambda^{i+1}} = \frac{i\,!}{\lambda^i} = m_i\,.$$

In particular, we have $m_{1,v} = m_1$. The mean remaining life-time from a random point of view is equal to the mean value of the life-time distribution, because the exponential distribution is without memory. Furthermore, the mean value of the actual life-time is also m_1 as we choose a random point of time. Thus the mean value of the total life-time becomes $2\,m_1$. □

2.2.5 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 be given by:

$$\begin{aligned} p\{j\text{'th largest} \leq t\} &= \sum_{i=0}^{j-1} \binom{k}{i} \{1 - F(t)\}^i \; F(t)^{k-i} \qquad (2.37) \\ &= 1 - \sum_{i=j}^{k} \binom{k}{i} \{1 - F(t)\}^i \; F(t)^{k-i} \, . \end{aligned}$$

as at most $j-1$ variables may be larger than t (but they may eventually all be less than t). The right-hand side is obtained using the Binomial theorem:

$$(a+b)^n = \sum_{i=0}^{n} \binom{n}{i} a^i \cdot b^{n-i} \, . \qquad (2.38)$$

The smallest one (or k'th largest, $j=k$) has the distribution function:

$$F_{\min}(t) = 1 - \{1 - F(t)\}^k \, , \qquad (2.39)$$

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

$$F_{\max}(t) = F(t)^k \, . \qquad (2.40)$$

If the random variables has individual distribution functions $F_i(t)$, we get an expression more complex than (2.37). For the smallest and the largest we get:

$$\begin{aligned} F_{\min}(t) &= 1 - \prod_{i=1}^{k} \{1 - F_i(t)\} \; , \qquad (2.41) \\ F_{\max}(t) &= \prod_{i=1}^{k} F_i(t) \, . \qquad (2.42) \end{aligned}$$

Example 2.2.7: Minimum of N exponentially distributed random variables
We assume that two random variables T_1 and T_2 are mutually independent and exponentially distributed with intensities λ_1 and λ_2, respectively. A new random variable T is defined as:

$$T = \min\{T_1, T_2\} \; .$$

The distribution function of T is (2.39):

$$p\{T \leq t\} = 1 - \mathrm{e}^{-(\lambda_1+\lambda_2)t} \, . \qquad (2.43)$$

Thus this distribution function 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 T_1 is realized first (i.e. takes places in this interval and the other takes place later) is given by:

$$\begin{aligned} p\{T_1 < T_2 \mid t\} &= \frac{P\{t < T_1 \leq t + \mathrm{d}t\} \cdot P\{T_2 > t\}}{P\{t < T \leq t + \mathrm{d}t\}} \\ &= \frac{\lambda_1 \, \mathrm{e}^{-\lambda_1 t} \, \mathrm{d}t \cdot \mathrm{e}^{-\lambda_2 t}}{(\lambda_1 + \lambda_2) \, \mathrm{e}^{-(\lambda_1+\lambda_2)t} \, \mathrm{d}t} \\ &= \frac{\lambda_1}{\lambda_1 + \lambda_2} \,, \end{aligned} \tag{2.44}$$

i.e. independent of t. These results can easily be generalized to N variables and make up the basic principle of the simulation technique called the *roulette method*, a Monte Carlo simulation methodology. □

2.3 Combination of random variables

Combining exponential distributed time intervals in series, we get a class of distributions called Erlang distributions (Sec. 2.3.1). Combining them in parallel, we obtain hyper–exponential distribution (Sec. 2.3.2). Combining exponential distributions both in series and in parallel, possibly with feedback, we obtain phase-type distributions, which is a very general class of distributions. One important sub–class of phase-type distributions is Cox-distributions (Sec. 2.4.2). 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.

2.3.1 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:

$$m_1 = \sum_{i=1}^{k} m_{1,i} \,, \tag{2.45}$$

$$\sigma^2 = \sum_{i=1}^{k} \sigma_i^2 \,. \tag{2.46}$$

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

The density function $f(t)$ of the sum is obtained by the convolution:

$$f(t)_{12\cdots k} = f_1(t) \otimes f_2(t) \otimes \cdots \otimes f_k(t)\,,$$

where $\otimes$ is the convolution operator:

$$\begin{aligned} f_{12}(t) &= f_1(t) \otimes f_2(t) \\ &= \int_0^t f_1(x) \cdot f_2(t-x)\,\mathrm{d}x \end{aligned} \tag{2.47}$$

Example 2.3.1: Non-homogenous Erlang-2 distribution
We consider two exponentially distributed independent time intervals T_1 and T_2 with intensities λ_1, respectively $\lambda_2 \neq \lambda_1$. The sum $T_{12} = T_1 + T_2$ is a random variable and the probability density function is obtained by convolution:

$$\begin{aligned} f_{12}(t) &= p(t < T_{12} \leq t + \mathrm{d}t) \\ &= \int_0^t f_1(x) \cdot f_2(t-x)\,\mathrm{d}x \\ &= \int_0^t \lambda_1\,\mathrm{e}^{-\lambda_1 x} \cdot \lambda_2\,\mathrm{e}^{-\lambda_2(t-x)}\,\mathrm{d}x \\ &= \lambda_1\,\lambda_2 \cdot \mathrm{e}^{-\lambda_2 t} \int_0^t \mathrm{e}^{-(\lambda_1-\lambda_2)x}\,\mathrm{d}x \\ &= \frac{\lambda_1\,\lambda_2}{\lambda_1-\lambda_2} \cdot \mathrm{e}^{-\lambda_2 t} \int_0^t (\lambda_1-\lambda_2) \cdot \mathrm{e}^{-(\lambda_1-\lambda_2)x}\,\mathrm{d}x \\ &= \frac{\lambda_1\,\lambda_2}{\lambda_1-\lambda_2} \cdot \mathrm{e}^{-\lambda_2 t} \left(1 - \mathrm{e}^{-(\lambda_1-\lambda_2)t}\right) \\ f_{12}(t) &= \frac{\lambda_1\,\lambda_2}{\lambda_1-\lambda_2} \cdot \mathrm{e}^{-\lambda_2 t} - \frac{\lambda_1\,\lambda_2}{\lambda_1-\lambda_2} \cdot \mathrm{e}^{-\lambda_1 t}\,, \quad \lambda_1 \neq \lambda_2\,. \end{aligned} \tag{2.48}$$

For the case $\lambda_1 = \lambda_2$ we get an Erlang-2 distribution considered in the following. □

Hypo-exponential or steep distributions

Steep distributions are also called hypo–exponential distributions or generalized Erlang distributions. They have a form factor within the interval $1 < \varepsilon \leq 2$. This distribution is obtained by convolving k exponential distributions (Fig. 2.4).

Figure 2.4: *By combining k exponential distributions in series we get a steep distribution with formfactor $\varepsilon \leq 2$). If all k distributions are identical ($\lambda_i = \lambda$), then we get an Erlang–k distribution.*

Erlang-k distributions

We consider the case where all k exponential distributions are identical. The distribution obtained $f_k(t)$ is called the *Erlang-k* distribution, as it was widely used by A.K. Erlang.

For $k = 1$ we of course get the exponential distribution. The distribution $f_k(t)$, $k > 0$, is obtained by convolving $f_{k-1}(t)$ and $f_1(t)$. If we assume that the expression (2.49) is valid for $f_{k-1}(t)$, then we have by *convolution*:

$$\begin{aligned} f_k(t) &= \int_0^t f_{k-1}(t-x)\, f_1(x)\, \mathrm{d}x \\ &= \int_0^t \frac{\{\lambda(t-x)\}^{k-2}}{(k-2)!}\, \lambda\, \mathrm{e}^{-\lambda(t-x)}\, \lambda\, \mathrm{e}^{-\lambda x}\, \mathrm{d}x \\ &= \frac{\lambda^k}{(k-2)!}\, \mathrm{e}^{-\lambda t} \int_0^t (t-x)^{k-2}\, \mathrm{d}x \\ f_k(t) &= \frac{(\lambda t)^{k-1}}{(k-1)!} \cdot \lambda \cdot \mathrm{e}^{-\lambda t}\,, \qquad \lambda > 0\,, \quad t > 0\,, \quad k = 1, 2, \ldots\,. \end{aligned} \tag{2.49}$$

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 *cdf* (cumulative distribution function) is obtained by repeated partial integration or as shown in a simple way later (3.21):

$$F_k(t) = \int_0^t f_k(t) = \sum_{j=k}^{\infty} \frac{(\lambda t)^j}{j\,!} \cdot \mathrm{e}^{-\lambda t} = 1 - \sum_{j=0}^{k-1} \frac{(\lambda t)^j}{j\,!} \cdot \mathrm{e}^{-\lambda t}\,. \tag{2.50}$$

The following moments can be found by using (2.45) and (2.46):

$$m_1 = \frac{k}{\lambda}\,, \tag{2.51}$$

$$\sigma^2 = \frac{k}{\lambda^2}\,, \tag{2.52}$$

$$\varepsilon = 1 + \frac{\sigma^2}{m^2} = 1 + \frac{1}{k}\,, \tag{2.53}$$

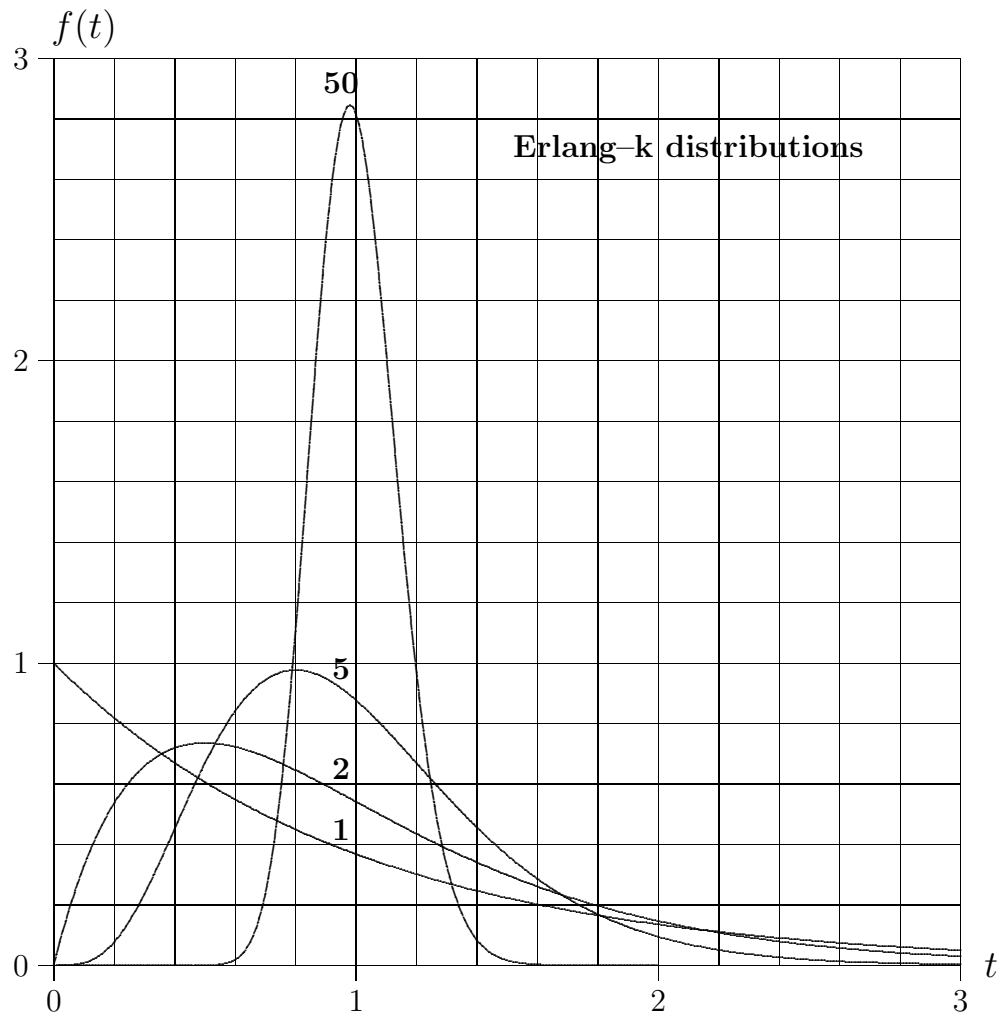


Figure 2.5: *Erlang–k distributions with mean value equal to one. The case $k = 1$ corresponds to an exponential distribution (density functions).*

The i'th non-central moment is:

$$m_i = \frac{(i+k-1)!}{(k-1)!} \cdot \left(\frac{1}{\lambda}\right)^i . \tag{2.54}$$

In particular, we have

$$m_2 = \frac{k\,(k+1)}{\lambda^2} . \tag{2.55}$$

The mean residual life–time $m_{1,r}(x)$ for $x \geq 0$ will be less than the mean value:

$$m_{1,r}(x) \leq m_1 , \qquad x \geq 0 .$$

Using 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, we normalize all Erlang–k distributions to the same mean value as the Erlang–1 distribution, i.e.

the exponential distribution with mean value $m_1 = 1/\lambda$, by replacing t by $k\,t$ or λ by $k\,\lambda$:

$$f_k(t)\,\mathrm{d}t = \frac{(\lambda k t)^{k-1}}{(k-1)\,!}\,\mathrm{e}^{-\lambda k t}\,k\lambda\,\mathrm{d}t\,, \tag{2.56}$$

$$m_1 = \frac{1}{\lambda}\,, \tag{2.57}$$

$$\sigma^2 = \frac{1}{k\,\lambda^2}\,, \tag{2.58}$$

$$\varepsilon = 1 + \frac{1}{k}\,. \tag{2.59}$$

Notice that the form factor is independent of time scale. The density function (2.56) is illustrated in Fig. 2.5 for different values of k with mean value $m_1 = 1$. The case $k = 1$ corresponds to the exponential distribution. When $k \to \infty$ we get a constant time interval ($\varepsilon = 1$). By solving for $f'(t) = 0$ we find the maximum value at:

$$\lambda t = \frac{k-1}{k}\,. \tag{2.60}$$

Steep distributions are named so because their distribution functions increase faster from 0 to 1 than the exponential distribution do.

2.3.2 Random variables in parallel

We combine ℓ independent time intervals (random variables) by choosing the i'th time interval with probability (weight factor) p_i,

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

The random variable of the weighted sum is called a *compound distribution*. The j'th (non-central) moment is obtained by weighting the (non-central) moments of the random variables:

$$m_j = \sum_{i=1}^{\ell} p_i \cdot m_{j,i}\,, \tag{2.61}$$

where $m_{j,i}$ is the j'th (non-central) moment of the distribution of the i'th interval. The mean value becomes:

$$m_1 = \sum_{i=1}^{\ell} p_i \cdot m_{1,i}\,. \tag{2.62}$$

The second moment is:

$$m_2 = \sum_{i=1}^{\ell} p_i \cdot m_{2,i}\,,$$

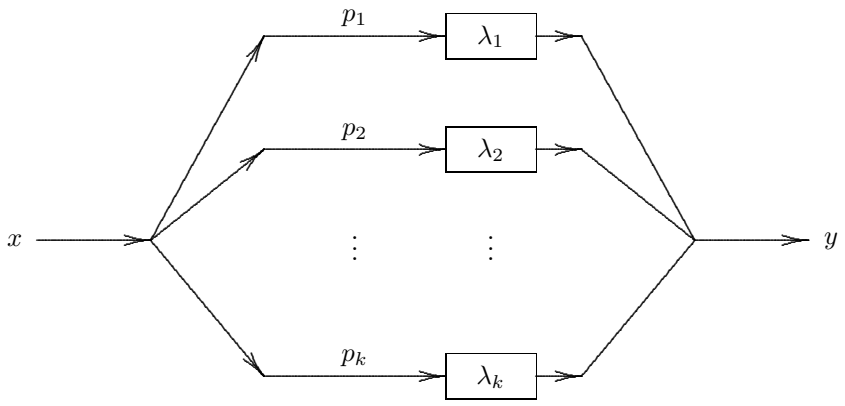


Figure 2.6: *By combining k exponential distributions in parallel and choosing branch number i with the probability p_i, we get a hyper–exponential distribution, which is a flat distribution ($\varepsilon \geq 2$).*

and from this we get the variance:

$$\sigma^2 = m_2 - m_1^2 = \sum_{i=1}^{\ell} p_i \cdot (\sigma_i^2 + m_{1,i}^2) - m_1^2 \,, \tag{2.63}$$

where σ_i^2, is the variance of the i'th distribution.

The distribution function is as follows:

$$F(t) = \sum_{i=1}^{\ell} p_i \cdot F_i(t) \,. \tag{2.64}$$

A similar formula is valid for the density function:

$$f(t) = \sum_{i=1}^{\ell} p_i \cdot f_i(t) \,.$$

Hyper-exponential distribution

Let k different exponential distributions have the following intensities:

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

and let us choose distribution i with probability p_i where

$$\sum_{i=1}^{k} p_i = 1 \,. \tag{2.65}$$

In this case (2.64) becomes:

$$F(t) = 1 - \sum_{i=1}^{k} p_i \cdot \mathrm{e}^{-\lambda_i t}\,, \quad t \geq 0\,. \tag{2.66}$$

This distribution class corresponds to a parallel combination of the exponential distributions. (Fig. 2.6). The mean values and form factor is obtained from (2.62) and (2.63) ($\sigma_i = m_{1,i} = 1/\lambda_i$):

$$m_1 = \sum_{i=1}^{k} \frac{p_i}{\lambda_i}\,, \tag{2.67}$$

$$\varepsilon = \left\{\sum_{i=1}^{k} p_i \frac{2}{\lambda_i^2}\right\} \Bigg/ \left\{\sum_{i=1}^{k} \frac{p_i}{\lambda_i}\right\}^2 \geq 2\,. \tag{2.68}$$

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

The distribution is called flat because its distribution function increases more slowly from 0 to 1 than the exponential distribution.

It is difficult to estimate more than one or two parameters (typically mean and variance) from real observations. The most common case in practise is $k = 2$ ($p_1 = p,\ p_2 = 1 - p$):

$$F(t) = 1 - p \cdot \mathrm{e}^{-\lambda_1 t} - (1-p) \cdot \mathrm{e}^{-\lambda_2 t}\,. \tag{2.69}$$

Statistical problems arise even when we have to estimate 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:

$$F(t) = 1 - p \cdot \mathrm{e}^{-2\lambda p t} - (1-p) \cdot \mathrm{e}^{-2\lambda(1-p)t}\,. \tag{2.70}$$

The mean value and form factor (assuming $p > 0$) becomes:

$$m_1 = \frac{1}{\lambda}\,,$$

$$\varepsilon = \frac{1}{2\,p\,(1-p)} > 2\,. \tag{2.71}$$

For this choice of parameters the two branches have the same contribution to the mean value. Fig. 2.7 shows an example of observed holding times approximated by a hyper-exponential distribution with these restrictions.

Flat distributions

A general hyper-exponential distribution is obtained using a a general distribution $\mathrm{d}W(\lambda) = w(\lambda)\mathrm{d}\lambda$ as weight function. The distribution becomes a compound distribution) with a form

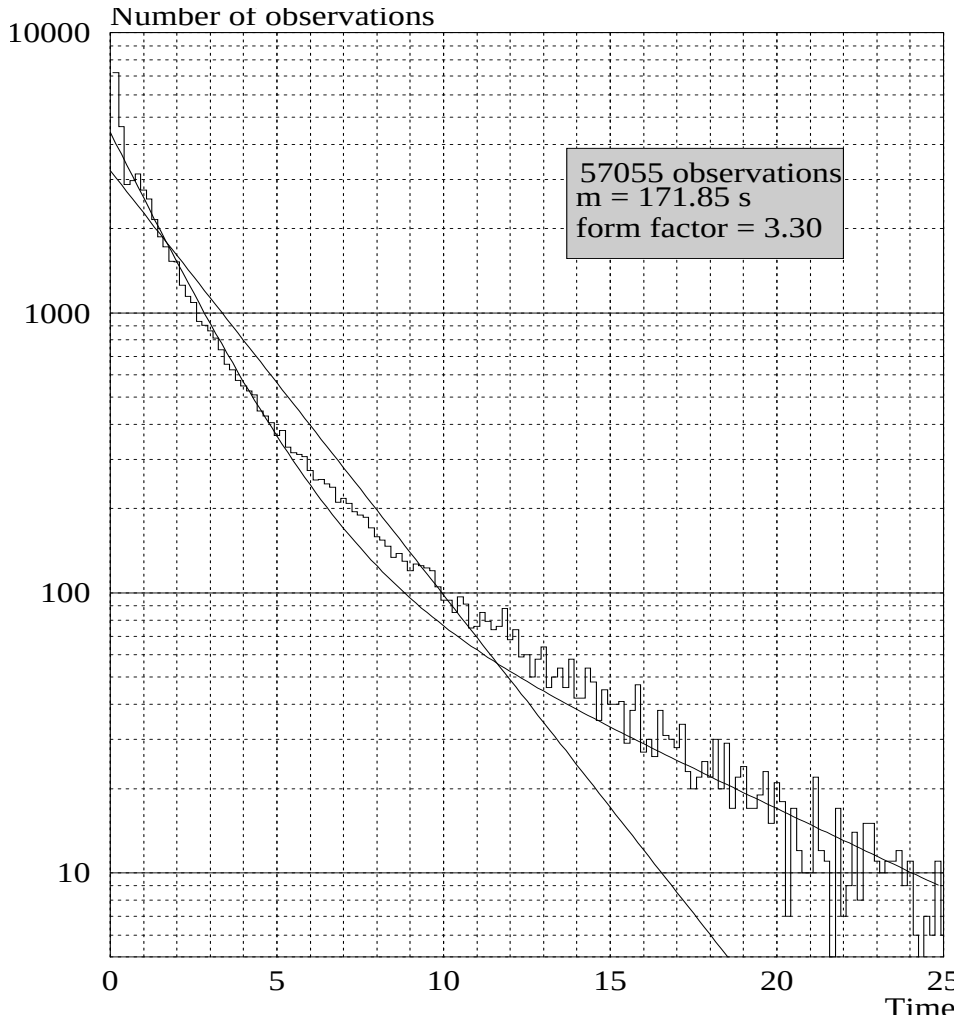


Figure 2.7: *Probability density function for holding times observed on lines in a local exchange during busy hours.*

factor $\varepsilon \geq 2$:

$$F(t) = \int_0^\infty \left(1 - \mathrm{e}^{-\lambda t}\right) \mathrm{d}W(\lambda)\,, \qquad \lambda > 0\,, \quad t \geq 0\,, \tag{2.72}$$

where the weight function may be discrete or continuous (Stieltjes integral). The density function is called complete monotone due to the alternating signs of the derivatives of the probability density function (Palm, 1957 [97]):

$$(-1)^\nu \cdot f^{(\nu)}(t) \geq 0\,. \tag{2.73}$$

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

$$m_{1,r}(x) \geq m_1\,, \qquad x \geq 0\,. \tag{2.74}$$

Pareto distribution and Palm's normal forms

In the most important case elaborated by Conny Palm (1943 [94]), $W(\lambda)$ is chosen to be gamma-distributed with mean value and form factor as follows:

$$\begin{aligned} m_1 &= \frac{1}{\lambda} \\ \varepsilon &= 1 + \eta_0/\lambda \end{aligned}$$

This corresponds to $\lambda = 1/\eta_0$ and $k = \lambda/\eta_0$ in (2.56) as the Erlang-k distribution is a special gamma-distribution (k integer). We then get

$$\mathrm{d}W(x) = \frac{1}{\eta_0} \cdot \frac{\left(\frac{x}{\eta_0}\right)^{\frac{\lambda}{\eta_0}-1}}{\Gamma\left(\frac{\lambda}{\eta_0}\right)} \cdot e^{-\frac{x}{\eta_0}} \cdot \mathrm{d}x\,, \tag{2.75}$$

and from (2.72) it can be shown that we get:

$$F(t) = 1 - (1+\eta_0 t)^{-\left(1+\frac{\lambda}{\eta_0}\right)}\,. \tag{2.76}$$

This distribution is called the *Pareto-distribution*. With the above choice of parameters the mean value and form factor of $F(t)$ becomes:

$$\begin{aligned} m_1 &= \frac{1}{\lambda}\,, \\ \varepsilon &= \frac{2\lambda}{\lambda - \eta_0}, \qquad 0 < \eta_0 < \lambda\,. \end{aligned} \tag{2.77}$$

Note that the variance does not exist for $\lambda \leq \eta_0$, and the distribution is called heavy-tailed (Sec. 2.5). This model is called *Palm's first normal form*, which has only two parameters (λ, η_0). As a special case, letting $\eta_0 \to 0$, the gamma-distribution (2.75) becomes a constant and (2.76) becomes an exponential distribution.

By weighting once more again using a gamma-distribution (κ) the result is a time distribution with three parameters which is called *Palm's second normal form:*

$$F(t) = 1 - \frac{1}{1+\eta_0 t}\left\{1 + \kappa\frac{\lambda}{\eta_0}\ln(1+\eta_0 t)\right\}^{-\left(1+\frac{1}{\kappa}\right)}, \quad \eta_0 > 0\,,\ \kappa > 0\,,\ t \geq 0\,. \tag{2.78}$$

The Pareto-distribution (2.76) is obtained from (2.78) by letting $\kappa \to 0$, or $\eta_0 \to 0$. If both κ and η tend to zero, we get the exponential distribution. We return to the normal forms in Sec. 3.6.

2.4 Phase-type distributions

By combining exponential random variables in both series and parallel we get an almost general class of distributions. By weak convergence it can be shown that we in this way can approximate any distribution function with any degree of accuracy. For the derivations it is useful first to consider the concept a stochastic sum (random sum).

2.4.1 Stochastic sum

By a stochastic sum we understand the sum of a stochastic number of random variables (Feller, 1950 [32]). Let us consider a system, 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 characterized by:

$$\begin{aligned} N: \quad & \text{density function } \; p(i) = p\{N = i\}\,, \quad i = 0, 1, 2, \ldots\,, \\ & \text{mean value } \; m_{1,n}\,, \\ & \text{variance } \; \sigma_n^2\,, \end{aligned} \tag{2.79}$$

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 characterized by:

$$\begin{aligned} T: \quad & \text{density function } \; f(t) = p(t < T \leq t + \mathrm{d}t)\,, \quad t \geq 0\,, \\ & \text{mean value } \; m_{1,t}\,, \\ & \text{variance } \; \sigma_t^2\,, \end{aligned} \tag{2.80}$$

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

$$S_T = T_1 + T_2 + \cdots + T_N\,. \tag{2.81}$$

In the following we assume that T_i and N are independent. This will be fulfilled in a system with many channels where calls are served without loss or delay. 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

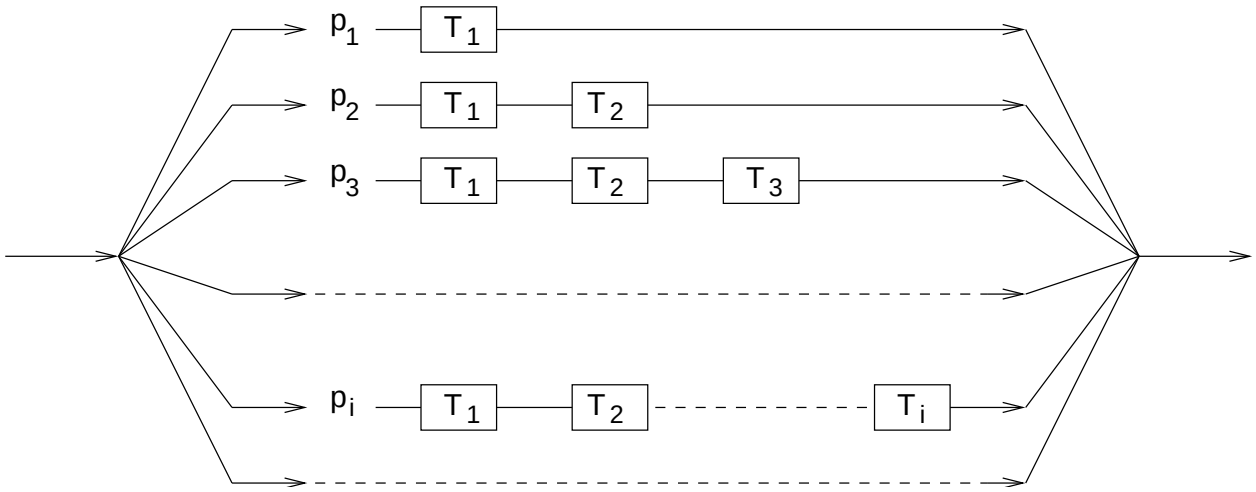


Figure 2.8: *A stochastic sum may be interpreted as a series/parallel combination of random variable.*

parallel as shown in Fig. 2.8 and dealt with in Sec. 2.3. For a given branch i we find (Fig. 2.8):

$$m_{1,i} = i \cdot m_{1,t}\,, \tag{2.82}$$

$$\sigma_i^2 = i \cdot \sigma_t^2\,, \tag{2.83}$$

$$m_{2,i} = i \cdot \sigma_t^2 + (i \cdot m_{1,t})^2\,. \tag{2.84}$$

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

$$\begin{aligned} m_{1,s} &= \sum_{i=1}^{\infty} p(i) \cdot m_{1,i} \\ &= \sum_{i=1}^{\infty} p(i) \cdot i \cdot m_{1,t}\,, \\ &= m_{1,t} \cdot m_{1,n}\,, \end{aligned} \tag{2.85}$$

$$m_{2,s} = \sum_{i=1}^{\infty} p(i) \cdot m_{2,i}$$

$$
\begin{aligned}
m_{2,s} &= \sum_{i=1}^{\infty} p(i) \cdot \left\{ i \cdot \sigma_t^2 + (i \cdot m_{1,t})^2 \right\}, \\
&= m_{1,n} \cdot \sigma_t^2 + m_{1,t}^2 \cdot m_{2,n}, \qquad (2.86) \\
\sigma_s^2 &= m_{1,n} \cdot \sigma_t^2 + m_{1,t}^2 \cdot m_{2,n} - (m_{1,t} \cdot m_{1,n})^2, \\
&= m_{1,n} \cdot \sigma_t^2 + m_{1,t}^2 \cdot \sigma_n^2 . \qquad (2.87)
\end{aligned}
$$

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

$$
\begin{aligned}
m_{1,s} &= n \cdot m_{1,t}, \\
\sigma_s^2 &= n \cdot \sigma_t^2 . \qquad (2.88)
\end{aligned}
$$

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

$$
\begin{aligned}
m_{1,s} &= m_{1,n} \cdot t, \\
\sigma_s^2 &= t^2 \cdot \sigma_n^2 . \qquad (2.89)
\end{aligned}
$$

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. Thus the variance/mean ratio becomes $m_{1,t}$ times bigger. □

Example 2.4.3: 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. □

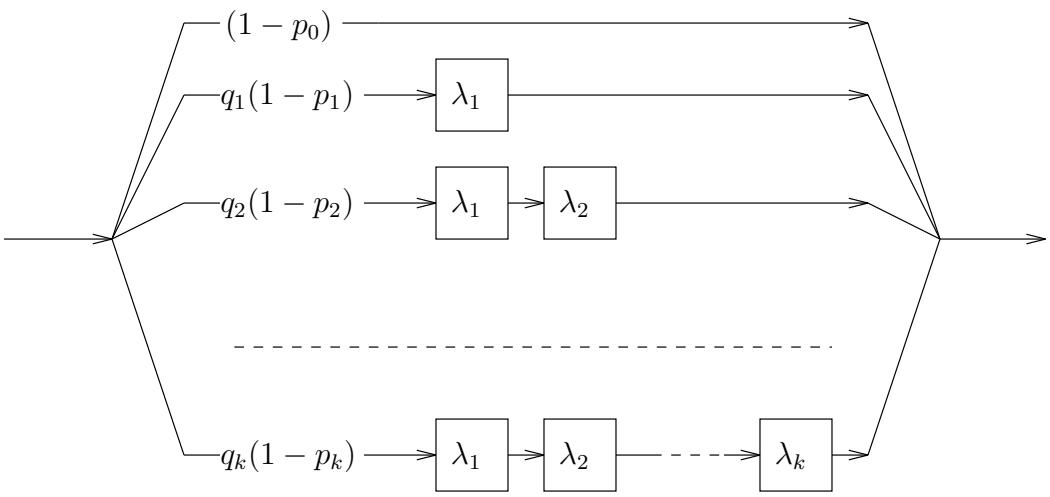


Figure 2.9: *A Cox–distribution is a generalized Erlang–distribution having exponential distributions in both parallel and series. The phase-diagram is equivalent to Fig. 2.10.*

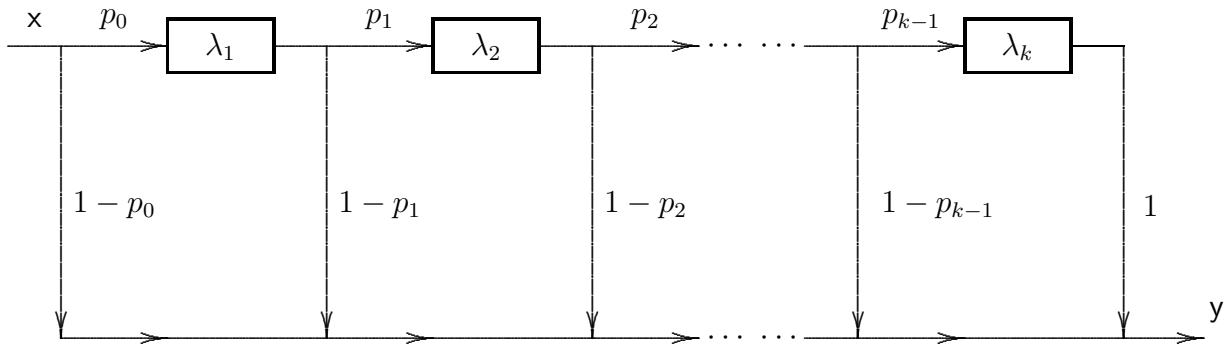


Figure 2.10: *The phase diagram of a Cox distribution, cf. Fig. 2.9.*

2.4.2 Cox distributions

By combining the steep and flat distributions we obtain a general class of distributions, Cox 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 theory of Markov processes, for which we have powerful tools as the phase-method. In the more general case with phase-type distributions we can allow for loop back betwee the phases.

We shall only consider *Cox-distributions* as shown in Fig. 2.9 (Cox, 1955 [18]). These also appear under the name of *branching Erlang* distributions. The mean value and variance of this Cox distribution (Fig. 2.10) are found from the formulae in Sec. 2.3 for random variables in series and parallel as shown in fig. 2.9:

$$m_1 = \sum_{i=1}^{k} q_i \,(1-p_i) \left\{ \sum_{j=1}^{i} \frac{1}{\lambda_j} \right\} , \tag{2.90}$$

where

$$q_i = p_0 \cdot p_1 \cdot p_2 \cdot \cdots \cdot p_{i-1} \,. \tag{2.91}$$

The term $q_i(1-p_i)$ is the probability of leaving after phase number i. It can be shown that the mean value can be expressed by the simple form:

$$m_1 = \sum_{i=1}^{k} \frac{q_i}{\lambda_i} = \sum_{i=1}^{k} m_{1,i} \,, \tag{2.92}$$

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

$$\begin{aligned} m_2 &= \sum_{i=1}^{k} \{q_i \,(1-p_i) \cdot m_{2,i}\} \\ &= \sum_{i=1}^{k} \left\{ q_i \,\, (1-p_i) \cdot \left\{ \sum_{j=1}^{i} \frac{1}{\lambda_j^2} + \left(\sum_{j=1}^{i} \frac{1}{\lambda_j} \right)^2 \right\} \right\} \,, \end{aligned} \tag{2.93}$$

where $m_{2,i}$ is obtained from (2.12): $m_{2,i} = \sigma_{2,i}^2 + m_{1,i}^2$. It can be shown that this can be written as:

$$m_2 = 2 \cdot \sum_{i=1}^{k} \left\{ \left(\sum_{j=1}^{i} \frac{1}{\lambda_j} \right) \cdot \frac{q_i}{\lambda_i} \right\} \,. \tag{2.94}$$

From this we get the variance (2.12):

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

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:

$$1 - F(t) = \sum_{i=1}^{k} c_i \cdot \mathrm{e}^{-\lambda_i t} \quad \text{where} \quad 0 \le \sum_{i=1}^{k} c_i \le 1 \,, \quad -\infty < c_i < +\infty \,. \tag{2.95}$$

2.4.3 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 this point is within phase i with probability:

$$\varrho_i = \frac{m_i}{m_1} \,, \qquad i = 1, 2, \ldots, k \,. \tag{2.96}$$

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):

$$p\{y_1, y_2, \ldots, y_k \mid y\} = \binom{y}{y_1,\, y_2,\, \ldots,\, y_k} \cdot \varrho_1^{y_1} \cdot \varrho_2^{y_2} \cdot \ldots \cdot \varrho_k^{y_k} \,, \tag{2.97}$$

where

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

and

$$\binom{y}{y_1,\, y_2,\, \ldots,\, y_k} = \frac{y!}{y_1! \cdot y_2! \cdot \ldots \cdot y_k!}\,. \tag{2.98}$$

This (2.98) is called the *multinomial coefficient*. 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.

By the *multinomial theorem* we have by summation over all possible states:

$$(\varrho_1 + \varrho_2 + \ldots + \varrho_k)^{\,y} = 1 = \sum_{\sum y_i = y} \binom{y}{y_1,\, y_2,\, \ldots,\, y_k} \cdot \varrho_1^{\,y_1} \cdot \varrho_2^{\,y_2} \cdot \ldots \cdot \varrho_k^{\,y_k}\,. \tag{2.99}$$

The multinomial theorem is also valid for $\sum_i \varrho_i \neq 1$. It is a generalization of the binomial theorem (2.38).

2.4.4 Decomposition principles

Phase–diagrams are a useful tool for analyzing Cox distributions. The following is a fundamental characteristic of the exponential distribution (Iversen & Nielsen, 1985 [47]):

Theorem 2.1 *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 (Fig. 2.11).*

According to Theorem 2.1 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. 2.13.

We have another property of Cox distributions (Iversen & Nielsen, 1985 [47]):

Theorem 2.2 *The phases in any Cox distribution can be ordered such as $\lambda_i \geq \lambda_{i+1}$.*

Theorem 2.1 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. 2.11). We notice that the branching probabilities are constant. Fig. 2.12 corresponds to a weighted sum of Erlang–k distributions where the weighting factors are geometrically distributed.

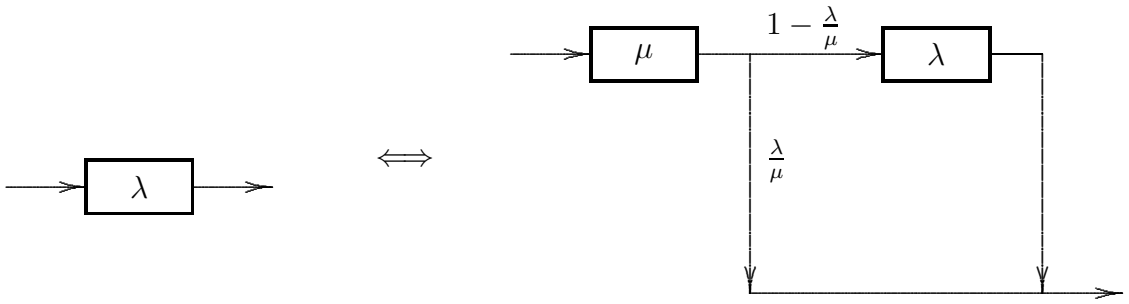


Figure 2.11: *An exponential distribution with rate λ is equivalent to the shown Cox–2 distribution (Theorem 2.1).*

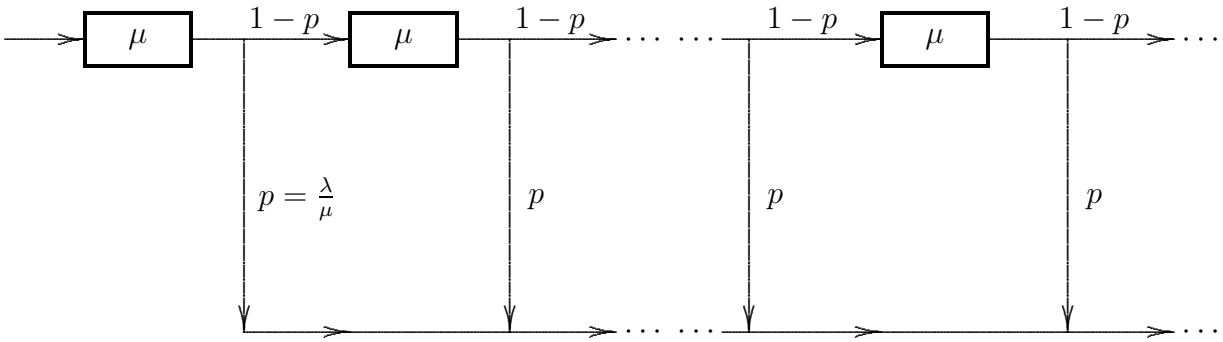


Figure 2.12: *An exponential distribution with rate λ is by successive decomposition transformed into a compound distribution of homogeneous Erlang–k distributions with rates $\mu > \lambda$, where the weighting factors follows a geometric distribution (quotient $p = \lambda/\mu$).*

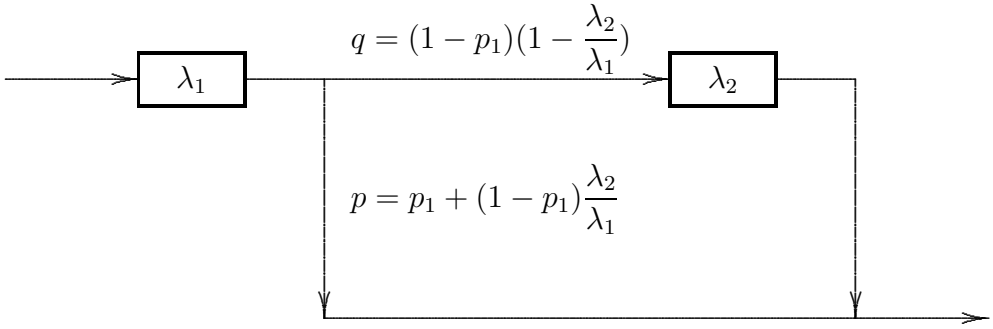


Figure 2.13: *A hyper–exponential distribution (Fig. 2.6) with two phases ($\lambda_1 > \lambda_2$, $p_2 = 1 - p_1$) can be transformed into a Cox–2 distribution.*

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. 2.14 we 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.

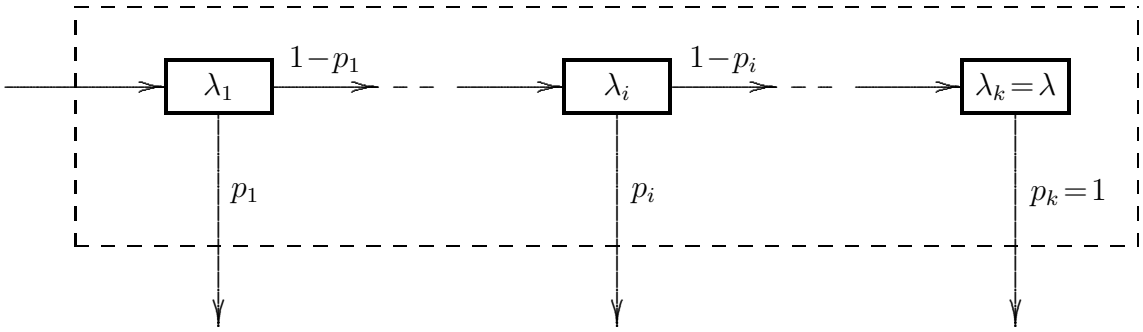


Figure 2.14: *This phase-type distribution is equivalent to a single exponential when* $p_i \cdot \lambda_i = \lambda$. *Thus* $\lambda_i \geq \lambda$ *as* $0 < p_i \leq 1$.

2.4.5 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 analyzed 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.

2.5 Other time distributions

In principle, every distribution which has non–negative values, may be used as a time distribution to describe the time intervals. For distributions which are widely applied in the queueing theory, we have the following abbreviated notations (cf. Sec. 10.1):

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.

Gamma distribution

If we suppose the parameter k in Erlang-k distribution (2.49) takes non-negative real values, then we obtain the *gamma distribution*:

$$f(t) = \frac{1}{\Gamma(k)}(\lambda t)^{k-1} \cdot \mathrm{e}^{-\lambda t} \cdot \lambda\,, \qquad \lambda > 0\,, \quad t \geq 0\,. \tag{2.100}$$

The mean value and variance are given in (2.51) and (2.52).

Weibull distribution

A distribution also known in teletraffic theory is the *Weibull distribution* $We(k, \lambda)$:

$$F(t) = 1 - \mathrm{e}^{-(\lambda t)^k}\,, \qquad t \geq 0\,, \quad k > 0\,, \quad \lambda > 0\,. \tag{2.101}$$

This distribution has a time-dependent death intensity (2.23):

$$\begin{aligned} \frac{\mathrm{d}F(t)}{1 - F(t)} = \mu(t) &= \frac{\lambda \mathrm{e}^{-(\lambda t)^k} \cdot k\,(\lambda t)^{k-1}\,\mathrm{d}t}{\mathrm{e}^{-(\lambda t)^k}} \\ &= \lambda k\,(\lambda t)^{k-1}\,. \end{aligned} \tag{2.102}$$

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

Heavy-tailed distributions

To describe data with big variations we often 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 (2.76) is heavy-tailed in strict sense.

Sometimes distributions with a tail more heavy than the exponential distribution are also classified as heavy-tailed. Examples are hyper-exponential, Weibull, and log-normal distributions. Another class of distributions is sub-exponential distribution. These subjects are dealt with in the literature.

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 (decrease) or tiredness (increase). In the same way, electro–mechanical systems work more slowly during periods of high load because the voltage decreases.

2.6 Observations of life-time distribution

Fig. 2.7 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. 3.4 shows observation and inter–arrival times of incoming calls to a transit telephone exchange during one hour. A particular outcome of a random variable is called a *random variate*. Thus observations of holding times are variates of a random variable we want to model.

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

Erlang (1920, [12]) reports a measurement where 2461 conversation times were recorded in a telephone exchange in Copenhagen in 1916. Palm (1943 [94]) analyzed 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 [41]). The importance of using discrete values of time when observing values is dealt with in Chapter 13. 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.

More recent extensive measurements have been performed and modeled using self-similar traffic models (Jerkins & al., 1999 [60]). These subjects are dealt with in more advanced chapters. For more advanced modelling Laplace transforms and Z-transforms are widely used.

Updated: 2011-05-03

Chapter 3

Arrival Processes

Arrival processes, such as telephone calls arriving to a switching system or messages arriving to a server are mathematically described as *stochastic point processes*. For a point process, we have to be able to distinguish two arrivals from each other. Information 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 fields. It was mathematically refined by Khintchine ([73], 1968), and is widely applicable in many fields.

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 generalizations 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 modeled by a Poisson process.

Due to its great importance in practice, we shall study the Poisson process in detail in this chapter. First (Sec. 3.5) 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. 3.6). Finally, in Sec. 3.7 we consider the interrupted Poisson process and the Batch Poisson process as examples of generalization.

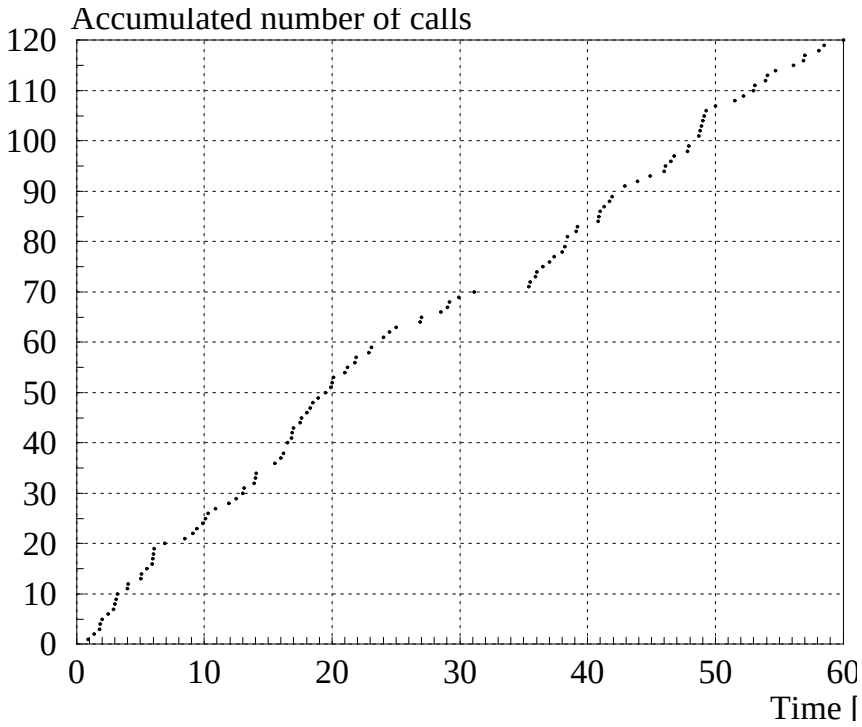


Figure 3.1: *The call arrival process at the incoming lines of a transit exchange.*

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

$$0 = T_0 < T_1 < T_2 < \ldots < T_i < T_{i+1} < \ldots . \tag{3.1}$$

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.

The time distance between two successive arrivals is:

$$X_i = T_i - T_{i-1}, \qquad i = 1, 2, \ldots . \tag{3.2}$$

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

Corresponding to the two random variables N_t and X_i, a point process can be characterized 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 n 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:

$$N_t < n \qquad \text{if and only if} \qquad T_n = \sum_{i=1}^{n} X_i \geq t\,, \quad n = 1, 2, \ldots \tag{3.3}$$

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

$$p\{N_t < n\} = p\{T_n \geq t\}\,, \qquad n = 1, 2, \ldots \tag{3.4}$$

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 on a per-call basis. Number representation has no parallel in time series analysis. The statistics we obtain are averaged over time and we get *time averages*, i.e. statistics on a per time unit basis (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.

3.1.1 Basic properties of number representation

There are three properties which are of 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*:*

$$H(t_1, t_2) = E\{N_{t_2} - N_{t_1}\}\,. \tag{3.5}$$

2. *The density* of arriving calls at time t (time average) is:

$$\lambda_t = \lim_{\Delta t \to 0} \frac{N_{t+\Delta t} - N_t}{\Delta t} = N_t'\,. \tag{3.6}$$

   We assume that λ_t exists and is finite. We may interpret λ_t as the *intensity = rate* by which arrivals occur at time t (cf. Sec. 2.2.2). For *simple or ordinary point processes*, we have:

$$p\{N_{t+\Delta t} - N_t \geq 2\} = o(\Delta t)\,, \tag{3.7}$$

$$p\{N_{t+\Delta t} - N_t = 1\} = \lambda_t \Delta t + o(\Delta t)\,, \tag{3.8}$$

$$p\{N_{t+\Delta t} - N_t = 0\} = 1 - \lambda_t \Delta t + o(\Delta t)\,, \tag{3.9}$$

where by definition:

$$\lim_{\Delta t \to 0} \frac{o(\Delta t)}{\Delta t} = 0\,. \tag{3.10}$$

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:

$$IDC = \frac{\mathrm{Var}\{N_t\}}{\mathrm{E}\{N_t\}}\,. \tag{3.11}$$

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*,indexpeakedness which we later introduce to characterize the number of busy channels in a traffic process (4.7).

3.1.2 Basic properties of interval representation

Also here we have three properties of interest.

4. The probability density function $f(t)$ of inter-arrival time intervals X_i (3.2), and by convolving this distribution by itself $i-1$ times, the distribution of the time until the i'th arrival.

$$F_i(t) = p\{X_i \le t\}\,, \tag{3.12}$$

$$E\{X_i\} = m_{1,i}\,. \tag{3.13}$$

The mean inter-arrival time 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, i.e. $m_{1,i} = m_1$ (IID = Identically and Independently Distributed).

5. The distribution function (pdf) $V(t)$ of the time interval from a random point (epoch) of time 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:

$$IDI = \frac{\mathrm{Var}\{X_i\}}{\mathrm{E}\{X_i\}^2} = \varepsilon - 1\,, \tag{3.14}$$

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 (2.14). 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. 13).

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

Example 3.1.1: 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 3.1.2: 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 equipment only collects data about 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 3.1.3: 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 3.1.4: Called party busy (B-Busy)
At a telephone exchange typically 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 [84]) analyzed 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 $b \cdot \varepsilon$. □

3.2 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.

3.2.1 Stationarity (Time homogeneity)

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:

$$p\{N_{t_1+t_2} - N_{t_1} = k\} = p\{N_{t_1+t_2+t} - N_{t_1+t} = k\} \ . \tag{3.15}$$

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.

3.2.2 Independence

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

$$p\{N_{t_2} - N_{t_1} = k | N_{t_1} - N_{t_0} = n\} = p\{N_{t_2} - N_{t_1} = k\} \tag{3.16}$$

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 3.2.1: 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 scans 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 acceptance of new calls for service depends on the instant when the first call of the busy period started service.

□

3.2.3 Simplicity or ordinarity

We have already mentioned (3.7) that we exclude processes with multiple arrivals.

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

$$p\left\{N_{t+\Delta t} - N_t \geq 2\right\} = o(\Delta t)\,. \tag{3.17}$$

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 (2.2):

$$F(0+) = 0 \tag{3.18}$$

Example 3.2.2: 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. □

3.3 Little's theorem

This is the only general result that is valid for all queueing systems. It was first published by Little (1961 [86]). The proof below was shown by applying the theory of stochastic process in (Eilon, 1969 [25]).

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. 3.2, 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 *statistical equilibrium* at initial time $t = 0$. We use the following notation (Fig. 3.2):

$N(T)$ = number of arrivals in period T.

$A(T)$ = the total service times of all customers in the period T
= the shadowed area between curves
= the carried traffic volume.

$\lambda(T)$ = $\frac{N(T)}{T}$ = the average call intensity in the period T.

$W(T)$ = $\frac{A(T)}{N(T)}$ = mean holding time in system per call in the period T.

$L(T)$ = $\frac{A(T)}{T}$ = the average number of calls in the system in the period T.

We have the important relation among these variables:

$$L(T) = \frac{A(T)}{T} = \frac{W(T) \cdot N(T)}{T} = \lambda(T) \cdot W(T) \tag{3.19}$$

If the limits of

$$\lambda = \lim_{T\to\infty} \lambda(T) \quad \text{and} \quad W = \lim_{T\to\infty} W(T)$$

exist, then the limiting value of $L(T)$ also exists and it becomes:

$$L = \lambda \cdot W \qquad \text{(Little's theorem)}. \tag{3.20}$$

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. 9–12.

Example 3.3.1: 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. 1.7. □

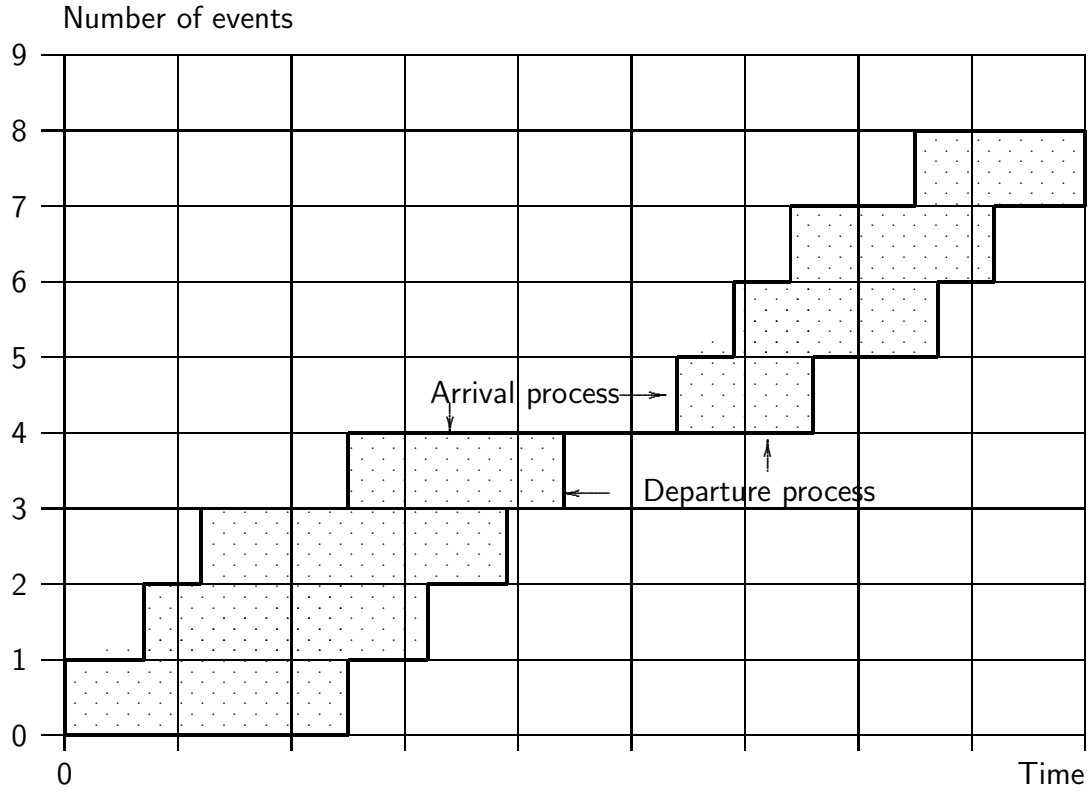


Figure 3.2: *A queueing system with arrival and departure of customers. The vertical distance between the two curves is equal to the actual number of customers being served. The customers in general don't depart in the the same order as they arrive, so the horizontal distance between the curves don't describe the actual time in the system of a customer.*

3.4 Characteristics of the Poisson process

The fundamental properties of the Poisson process are defined in Sec. 3.2:

a. *Stationary*,

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

c. Simple.

(*b*) and (*c*) are fundamental properties, whereas (*a*) can be relaxed. 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 (3.2) between consecutive events is *exponentially distributed*.

In this case using (2.49) and (2.50) *Feller–Jensen's identity* (3.4) shows the fundamental relationship between the cumulated Poisson distribution and the Erlang-k distribution (Sec. 3.5.2):

$$\sum_{j=0}^{k-1} \frac{(\lambda t)^j}{j!} \cdot \mathrm{e}^{-\lambda t} = \int_{x=t}^{\infty} \frac{(\lambda x)^{k-1}}{(k-1)!}\, \lambda \cdot \mathrm{e}^{-\lambda x}\, \mathrm{d}x = 1 - F(t)\,. \tag{3.21}$$

This formula can also be obtained by repeated partial integration.

3.5 Distributions of the Poisson process

In this section we consider the Poisson process in a dynamical and physical way (Fry, 1928 [35]) & (Jensen, 1954 [12]). The derivations are based on a simple physical model and focus upon 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.

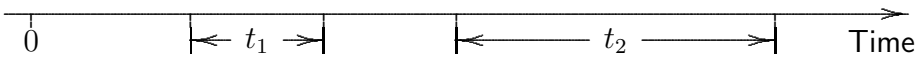


Figure 3.3: *When deriving the Poisson process, we consider arrivals within two non–overlapping time intervals of duration t_1 and t_2, respectively.*

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:* Let t_1 and t_2 be two non–overlapping intervals (Fig. 3.3), then because of the independence assumption we have:

$$p\,(0,t_1)\cdot p\,(0,t_2) = p\,(0,t_1+t_2)\ . \tag{3.22}$$

2. We notice that (3.22) implies that the event *"no arrivals within the interval of length 0"* has the probability one:

$$p(0,0) = 1\,. \tag{3.23}$$

3. The mean value of the time interval between two successive arrivals is $1/\lambda$ (2.8):

$$\int_0^\infty p(0,t)\,\mathrm{d}t = \frac{1}{\lambda}, \qquad 0 < \frac{1}{\lambda} < \infty\,. \tag{3.24}$$

   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 function). The mean value (3.24) is obtained directly from (2.8). Formula (3.24) can also be interpreted as the area under the curve $p(0,t)$, which is a non-increasing function decreasing from 1 to 0.

4. We also notice that (3.24) implies that the probability of *"no arrivals within a time interval of length ∞"* is zero as it never takes place:

$$p(0,\infty) = 0\,. \tag{3.25}$$

3.5.1 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 $\{1-p(0,t) = F(t)\}$ is an exponential distribution (cf. Sec. 2.1.1).

From (3.22) we have:

$$\ln p\,(0,t_1) + \ln p\,(0,t_2) = \ln p\,(0,t_1+t_2)\ . \tag{3.26}$$

Letting $\ln p(0,t) = f(t)$, (3.26) can be written as:

$$f\,(t_1) + f\,(t_2) = f\,(t_1+t_2)\ . \tag{3.27}$$

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

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

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

$$f(t) = a + b\,t\,. \tag{3.28}$$

By inserting (3.28) into (3.27), we obtain $a = 0$. Therefore $p(0,t)$ has the form:

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

From (3.24) 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:

$$p(0,t) = \mathrm{e}^{-\lambda t}\,. \tag{3.29}$$

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. 2.1.1):

$$1 - p(0,t) = F(t) \;=\; 1 - \mathrm{e}^{-\lambda t}, \qquad \lambda > 0\,, \quad t \geq 0\,, \tag{3.30}$$

$$F'(t) = \;f(t) \;=\; \lambda \cdot \mathrm{e}^{-\lambda t}\,, \qquad \lambda > 0\,, \quad t \geq 0\,. \tag{3.31}$$

We have the following mean value and variance (2.18):

$$\begin{aligned} m_1 &= \frac{1}{\lambda}\,, \\ \sigma^2 &= \frac{1}{\lambda^2}\,. \end{aligned} \tag{3.32}$$

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

$$\begin{aligned} f(t)\,\mathrm{d}t &= \lambda \mathrm{e}^{-\lambda t}\,\mathrm{d}t \\ &= p(0,t)\,\lambda\,\mathrm{d}t\,, \end{aligned} \tag{3.33}$$

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* $\mathrm{d}t$ (2.26).

Because λ is independent of the actual *age* t, the exponential distribution has no memory (cf. Secs. 2.1.1 & 2.2.2). 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 (3.6). The exponential distribution is in general a very good model of call inter-arrival times when the traffic is generated by human beings (Fig. 3.4).

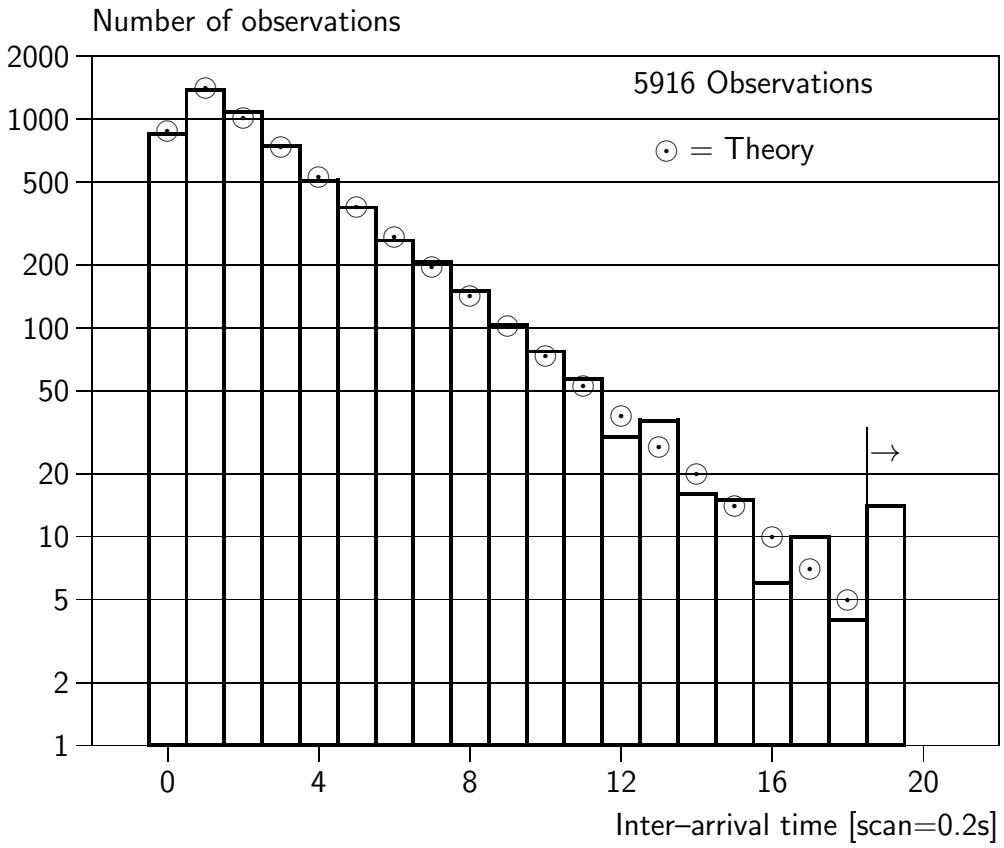


Figure 3.4: *Inter–arrival time distribution of calls at a transit exchange. The theoretical values are based on the assumption of exponentially distributed inter–arrival times. Due to the measuring principle (scanning method) the continuous exponential distribution is transformed into a discrete Westerberg distribution (13.14) (χ^2-test = 18.86 with 19 degrees of freedom, percentile = 53).*

3.5.2 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. 2.3.1) and the density is given by (2.49):

$$f_k(t)\,\mathrm{d}t = \lambda\,\frac{(\lambda t)^{k-1}}{(k-1)!}\,\mathrm{e}^{-\lambda t}\,\mathrm{d}t\,, \qquad \lambda > 0\,, \quad t \geq 0\,, \quad k = 1, 2, \ldots\,. \tag{3.34}$$

The mean value and the variance are obtained in (2.51) – (2.55): from (3.32):

$$\begin{aligned} m_1 &= \frac{k}{\lambda}\,, \\ \sigma^2 &= \frac{k}{\lambda^2}\,, \\ \varepsilon &= 1 + \frac{1}{k}\,. \end{aligned} \tag{3.35}$$

Example 3.5.1: Call statistics from an SPC-system (cf. Example 3.1.2)
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. □

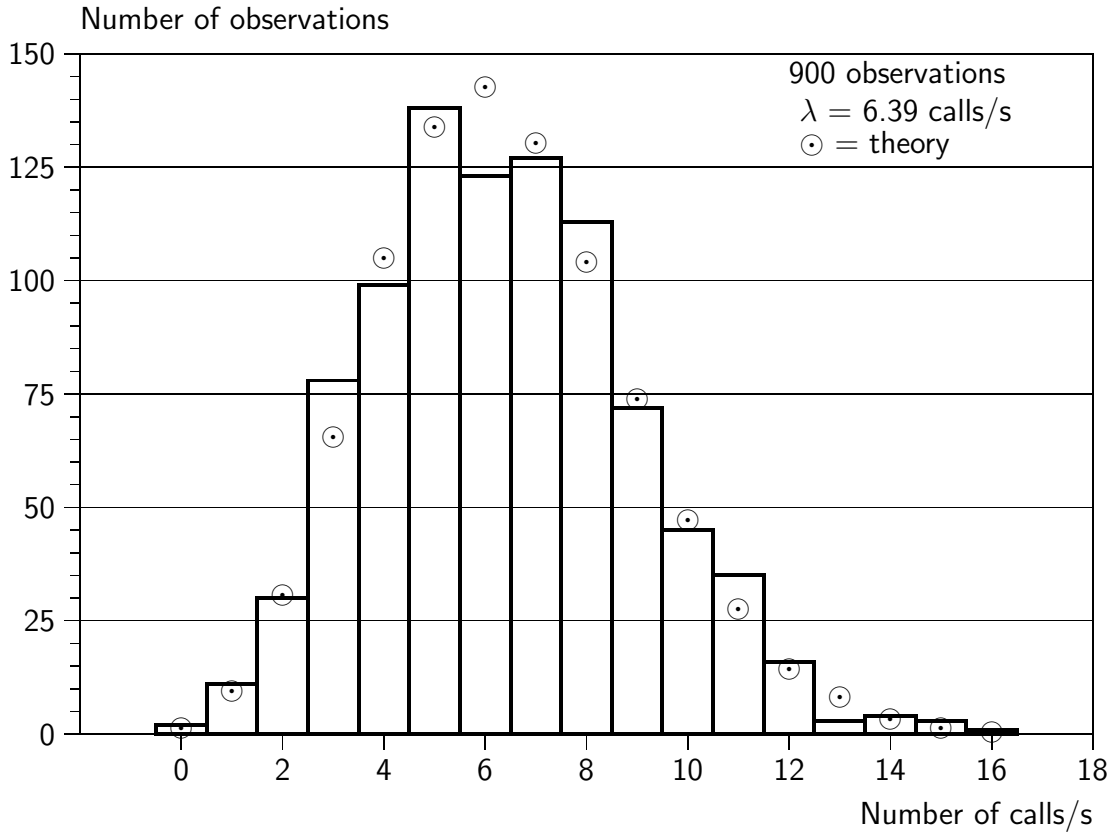


Figure 3.5: *Number of Internet dial-up calls per second. The theoretical values are based on the assumption of a Poisson distribution. A statistical test accepts the hypothesis of a Poisson distribution.*

3.5.3 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 $i = 1$ (3.29). 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 integrating over all possible values of t_1 as a product of the following three independent 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:

$$\begin{aligned} p(i,t) &= \int_0^t \frac{(\lambda t_1)^{i-1}}{(i-1)!}\,\mathrm{e}^{-\lambda t_1} \cdot \lambda\,\mathrm{d}t_1 \cdot \mathrm{e}^{-\lambda(t-t_1)} \\ &= \frac{\lambda^i}{(i-1)!}\,\mathrm{e}^{-\lambda t} \int_0^t t_1^{i-1}\,\mathrm{d}t_1\,, \\ p(i,t) &= \frac{(\lambda t)^i}{i!} \cdot \mathrm{e}^{-\lambda t}\,, \qquad i = 0,1,\ldots \quad \lambda > 0\,. \end{aligned} \tag{3.36}$$

This is the Poisson distribution which we thus have obtained from (3.29) by induction. The mean value and variance are:

$$m_1 = \lambda \cdot t\,, \tag{3.37}$$

$$\sigma^2 = \lambda \cdot t\,. \tag{3.38}$$

The Poisson distribution is in general a very good model for the number of calls in a telecommunication system (Fig. 3.5) or jobs in a computer system.

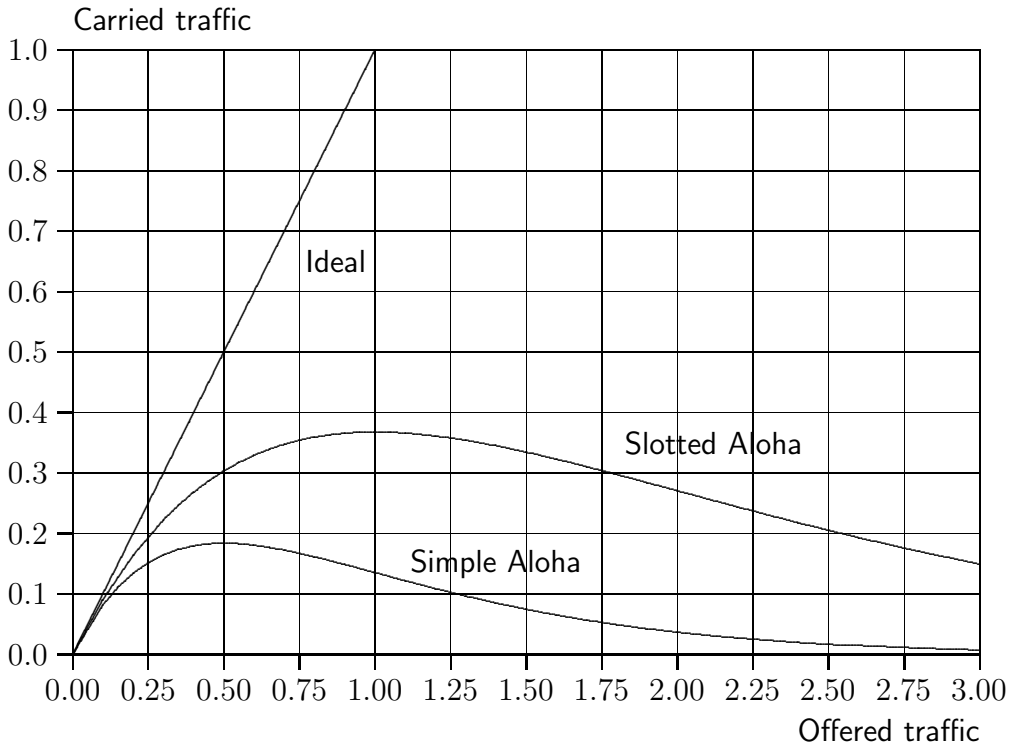


Figure 3.6: *The carried traffic in a slotted Aloha system has a maximum throughput twice the maximum throughput of the simple Aloha system (example 3.5.2). The Simple Aloha protocol is dealt with in example 4.2.1.*

Example 3.5.2: Slotted Aloha Satellite System
Let us consider a digital satellite communication system with constant packet length h. The satellite 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 synchronized 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.

$$p(i) = \frac{(\lambda h)^i}{i!} \cdot \mathrm{e}^{-\lambda h}. \tag{3.39}$$

The probability of a correct transmission is:

$$p(1) = \lambda h \cdot \mathrm{e}^{-\lambda h}. \tag{3.40}$$

This corresponds to the proportion of the time axes which is utilized effectively. This function, which is shown in Fig. 3.6, has an optimum when the derivative with respect to λh is zero:

$$p'_{\lambda h}(1) = \mathrm{e}^{-\lambda h} \cdot (1 - \lambda h) = 0\,, \tag{3.41}$$

$$\lambda h = 1\,.$$

Inserting this value in (3.40) we get:

$$\max\{p(1)\} = e^{-1} = 0.3679\,. \tag{3.42}$$

We thus have a maximum utilization 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. □

3.5.4 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 $n \cdot p$ 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 3.1.

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 3.1.

3.6 Properties of the Poisson process

In this section we shall show some fundamental properties of the Poisson process. From the physical model in Sec. 3.5 we have seen that the Poisson process is the most random

BINOMIAL PROCESS Discrete time Probability of success: $p\,, \quad 0 < p < 1$	POISSON PROCESS Continuous time Intensity of success: $\lambda\,, \quad \lambda > 0$
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 $p(n) = p \cdot (1-p)^{n-1}\,, \quad n = 1, 2, \ldots$ $m_1 = \frac{1}{p}\,, \qquad \sigma^2 = \frac{1-p}{p^2}$	EXPONENTIAL DISTRIBUTION $f(t) = \lambda \cdot \mathrm{e}^{-\lambda t}\,, \quad t \geq 0$ $m_1 = \frac{1}{\lambda}\,, \qquad \sigma^2 = \frac{1}{\lambda^2}$
Number of attempts to get k successes	Time interval until k'th success
PASCAL = NEGATIVE BINOMIAL DISTR. $p(n \mid k) = \binom{n-1}{k-1} p^k (1-p)^{n-k}\,, \; n \geq k$ $m_1 = \frac{k}{p}\,, \qquad \sigma^2 = \frac{k(1-p)}{p^2}$	ERLANG–K DISTRIBUTION $f_k(t) = \frac{(\lambda t)^{k-1}}{(k-1)!} \cdot \lambda \cdot \mathrm{e}^{-\lambda t}\,, \qquad t \geq 0$ $m_1 = \frac{k}{\lambda}\,, \qquad \sigma^2 = \frac{k}{\lambda^2}$
Number of successes in n attempts	Number of successes in a time interval t
BINOMIAL DISTRIBUTION $p(x \mid n) = \binom{n}{x} p^x (1-p)^{n-x}\,, \quad x = 0, 1, \ldots$ $m_1 = p\,n\,, \qquad \sigma^2 = p\,n \cdot (1-p)$	POISSON DISTRIBUTION $f(x,t) = \frac{(\lambda t)^x}{x!} \cdot \mathrm{e}^{-\lambda t}\,, \quad t \geq 0$ $m_1 = \lambda\,t\,, \qquad \sigma^2 = \lambda\,t$

Table 3.1: *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.*

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. In a Poisson process events occur at random during time and therefore call averages and time averages are identical. This is the so-called *PASTA*–property: Poisson Arrivals See Time Averages.

3.6.1 Palm-Khintchine 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 (the central limit theorem).

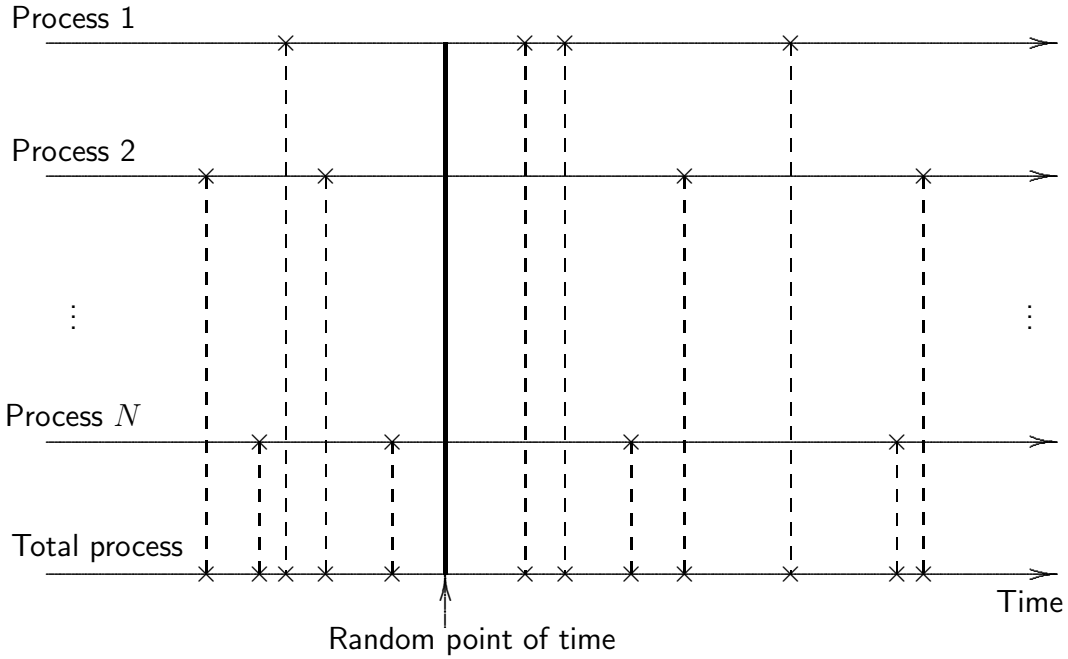


Figure 3.7: *By superposition of N independent point processes we obtain under certain assumptions a process which locally is a Poisson process.*

Theorem 3.1 *Palm-Khintchine 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 (2.34).

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 (2.34):

$$p\{T \leq t\} = 1 - \prod_{i=1}^{N} \left\{ 1 - V_i\left(\frac{t}{N}\right) \right\} . \tag{3.43}$$

If all sub-processes are identical, we get:

$$p\{T \leq t\} = 1 - \left\{ 1 - V\left(\frac{t}{N}\right) \right\}^N . \tag{3.44}$$

From (2.34) and (3.18) we find (letting $m_1 = 1$):

$$\lim_{\Delta t \to 0} v(\Delta t) = 1 \,,$$

and thus:

$$V(\Delta t) = \int_0^{\Delta t} 1 \, \mathrm{d}t = \Delta t \,. \tag{3.45}$$

Therefore, we get from (3.44) by letting the number of sub-processes increase to infinity:

$$\begin{aligned} p\{T \leq t\} &= \lim_{N \to \infty} \left\{ 1 - \left(1 - \frac{t}{N} \right)^N \right\} \\ &= 1 - \mathrm{e}^{-t} \,. \end{aligned} \tag{3.46}$$

which is the exponential distribution. We have thus shown that by superposition of N identical processes we locally get a Poisson process. In a similar way we may superpose non-identical processes and locally obtain a Poisson process.

Example 3.6.1: 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. 8). 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-Khintchine theorem we see that the life-time of the route tends to be exponentially distributed. □

Corollary to Palm-Khintchine theorem *(Poisson superposition theorem): By superposition of N independent Poisson processes we obtain a Poisson process.*

This is the only case we obtain an exact Poisson process. It can be proven (1) by remembering that the smallest of N exponential distributions is itself an exponential distribution (Example 2.2.7) (interval representation) or (2) by observing that the sum of N Poisson distributions is a Poisson distribution (number representation).

3.6.2 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 3.2 *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.*

This is also indicated by the following general result. If we generate a sub-process by random splitting of a point process choosing an event with probability $p_i, \{i = 1, 2, \ldots, N\}$, then the sub-process has the form factor ε_i:

$$\varepsilon_i = 2 + p_i \cdot (\varepsilon - 2)\,, \tag{3.47}$$

where ε is the form factor of the original process. When p_i approaches zero the form factor becomes 2 as for the exponential distribution. The result is only exact when the original process is a Poisson process:

Corollary to Raikov's theorem *(Poisson splitting theorem): By splitting a Poisson process into N sub-processes, each sub-process will be an independent Poisson processes.*

This can be shown both by interval representation and number representation.

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 for example is a good description of the arrival processes to a local exchange which usually is generated by many independent local subscribers.

3.6.3 Uniform distribution – a conditional property

In Sec. 3.5 we have seen that *a uniform distribution in a very large interval* corresponds to a Poisson process. The inverse property is also valid (proof left out):

Theorem 3.3 *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. 13). This can be shown both from the Poisson distribution (number representation) and from the exponential distribution (interval presentation).

3.7 Generalization of the stationary Poisson process

The Poisson process has been generalized in many ways. In this section we only consider the interrupted Poisson process, but further generalizations are *MMPP* (Markov Modulated Poisson Processes) and *MAP* (Markov Arrival Processes).

3.7.1 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 flexible enough to describe a real arrival process as it has only one parameter. Kuczura (1973 [80]) proposed a generalization which has been widely used.

The idea of generalisation comes from the overflow problem (Fig. 3.8 & Sec. 6.4). 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 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. 3.9). As a 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 3.10. It can be shown that the parameters are related as follows:

$$\begin{aligned} \lambda &= p\,\lambda_1 + (1-p)\lambda_2\,, \\ \lambda \cdot \omega &= \lambda_1 \cdot \lambda_2\,, \\ \lambda + \gamma + \omega &= \lambda_1 + \lambda_2\,. \end{aligned} \tag{3.48}$$

Because a hyper–exponential distribution with two phases can be transformed into a *Cox–2* distribution (Sec. 2.4.4), the *IPP* arrival process is a *Cox-2* arrival processes as shown in

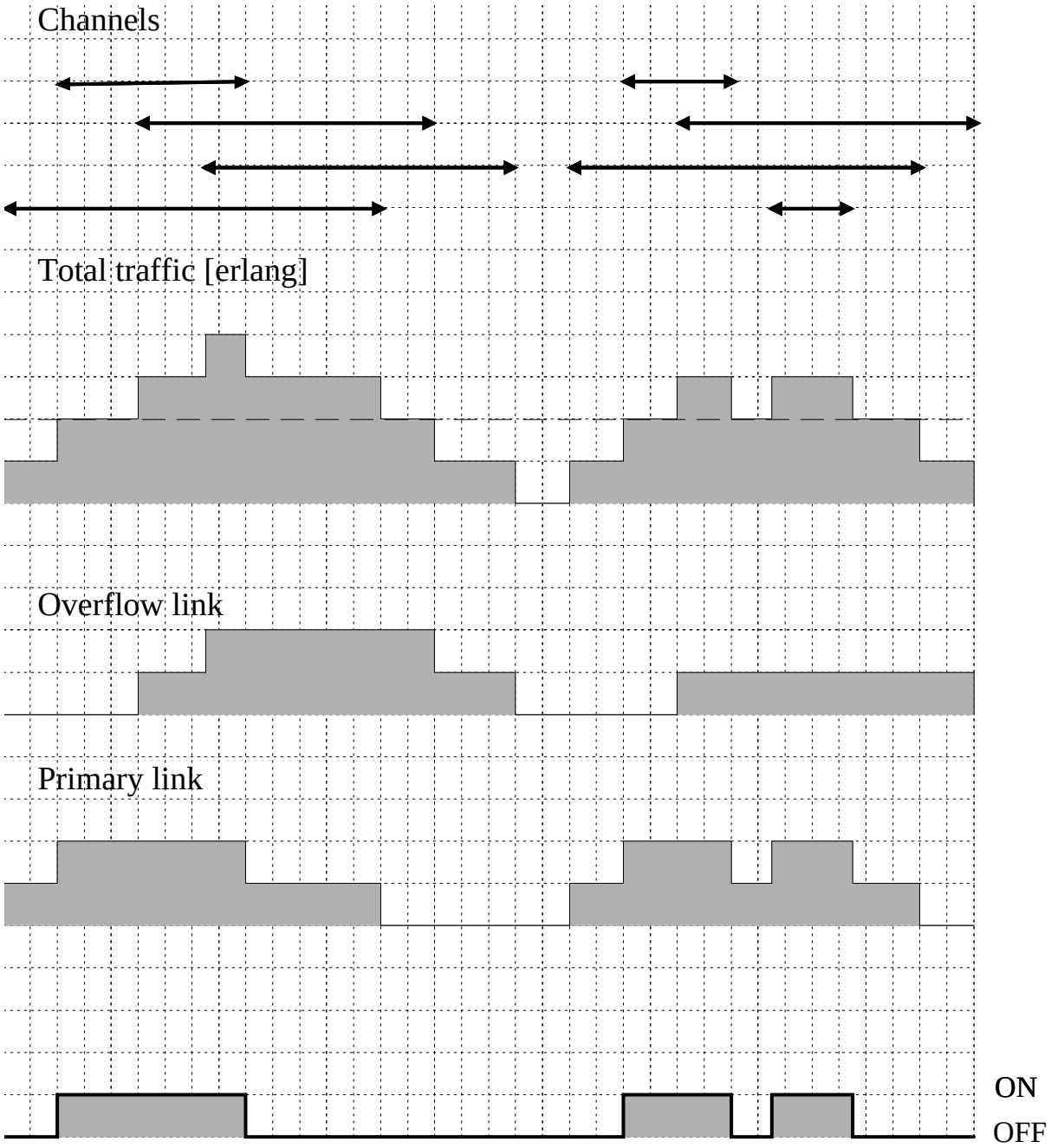


Figure 3.8: *Overflow system with Poisson arrival process (intensity λ). Normally, calls arrive to the primary group. During periods when all n trunks in the primary group are busy, all calls are offered to the overflow group.*

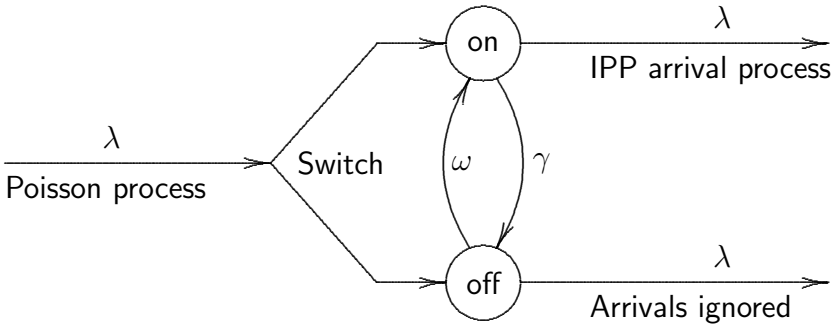


Figure 3.9: *Illustration of the interrupted Poisson process (IPP) (cf. Fig. (3.8)). The position of the switch is controlled by a two-state Markov process.*

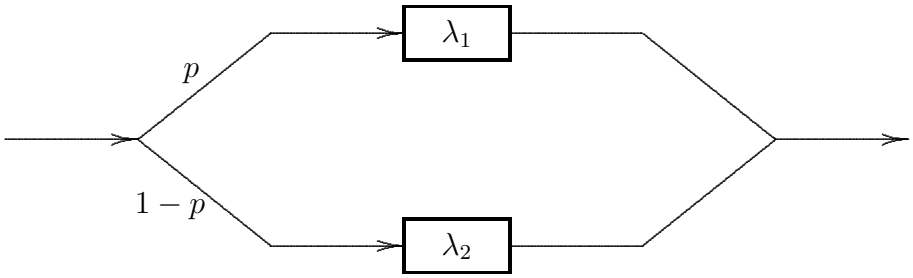


Figure 3.10: *The interrupted Poisson process is equivalent to a hyper–exponential arrival process (3.48).*

Fig. 2.13. We have three parameters available, whereas the Poisson process has only one parameter. This makes it more flexible for modelling empirical data.

3.7.2 Batch Poisson process

We consider an arrival process where events occur according to a Poisson process with rate λ. At each event a batch of calls (packets, jobs) arrive simultaneously. The distribution of the batch size is in the general case a discrete distribution $p(i)\,, (i = 1, 2, \ldots)$. The batch size is at least one. In the Poisson arrival process the batch size is always one. We choose the simplest case where the distribution is a geometric distribution (Tab. 3.1, p. 92):

$$p(i) = p\,(1-p)^{i-1}\,, \quad i = 1, 2, \ldots\,. \tag{3.49}$$

$$m_1 = \frac{1}{p}\,, \tag{3.50}$$

$$\sigma^2 = \frac{1-p}{p^2}\,. \tag{3.51}$$

The number of events during a time interval t then becomes a stochastic sum (Sec. 2.4) where N (2.79) is a Poisson distribution with mean value and variance $\lambda\, t$ and T (2.80) is the

geometric distribution given above. The mean value of the number of events during a time interval t is (2.85):

$$m_{1,s} = \lambda\, t \cdot \frac{1}{p}\,, \tag{3.52}$$

and the variance is (2.87):

$$\begin{aligned} \sigma_s^2 &= \lambda\, t \cdot \frac{1-p}{p^2} + \left(\frac{1}{p}\right)^2 \lambda\, t \\ &= \lambda\, t \cdot \frac{2-p}{p^2}\,. \end{aligned} \tag{3.53}$$

The index of dispersion of counts (3.11) becomes:

$$IDC = \frac{\sigma_s^2}{m_{1,s}} = \frac{2-p}{p}\,. \tag{3.54}$$

For $p = 1$ the geometric distribution always takes the value one and we get a Poisson process. For $p < 1$ the process is more bursty than the Poisson process.

Chapter 4

Erlang's loss system and B–formula

In this and the following chapters we consider the classical teletraffic theory developed by Erlang (Denmark), Engset (Norway) and Fry & Molina (USA). It has successfully been applied for more than 80 years. In this chapter we consider the fundamental Erlang-B formula. In Sec. 4.1 we specify the assumptions for the model. Sec. 4.2 deals with infinite capacity, which results in a Poisson distributed number of busy channels. In Sec. 4.3 we consider a limited number of channels and obtain the truncated Poisson distribution and Erlang's B-formula. Sec. 4.4 describes a standard procedure for dealing with state transition diagrams (*STD*) which are the key to classical teletraffic theory. We also derive an accurate recursive formula for numerical evaluation of Erlang's B-formula (Sec. 4.5). In Sec. 4.6 properties of Erlang's B-formula are studied. Thus we consider non-integral number of channels, insensitivity, derivatives, inverse formulæ, and approximations. Sec. 4.7 considers the Blocked Calls Held model, which is useful for many applications. Finally, in Sec. 4.8 we study the basic principles of dimensioning, where we balance Grade–of–Service (GoS) against costs of the system.

4.1 Introduction

Erlang's B-formula is based on the following model, described by the three elements *structure, strategy, and traffic* (Fig. 1.1):

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 at least one 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 and dependability. If all channels are busy the system

is congested and call attempts are blocked. A blocked (= rejected, lost) call attempt disappears without any after-effect as it may be accepted by an alternative route. This strategy is the most important one and has been applied with success for many years. This is called *Erlang's loss model* or the *Blocked Calls Cleared* = *BCC*–model. Usually, we assume that the service time is independent of both the arrival process and other service times.

Within a full accessible group we may look for an idle channel in different ways:

- *Random hunting:* we choose a random channel among the idle channels. On average every channel will carry the same traffic.
- *Ordered hunting:* the channels are numbered $1, 2, \ldots n$, and we search for an idle channel in this order, always starting with channel one (ordered hunting with homing). This is also called *sequential hunting*. A channel will on the average carry more traffic than the following channels.
- *Cyclic hunting:* this is similar to ordered hunting, but without homing. We continue hunting for an idle channel starting from the position where we ended last time. Also in this case every channel will on the average carry the same traffic.

The hunting takes place momentarily. If all channels are busy a call attempts is blocked. The blocking probability is independent of the hunting mode.

c. *Traffic:* In the following we assume that:

- The arrival process is a Poisson process with rate λ, and
- The service times are exponentially distributed with intensity μ (corresponding to a mean value $1/\mu$).

This type of traffic is called *Random Traffic* (*RT*) or *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 is infinite (1.2). 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:

$$A = \lambda \cdot \frac{1}{\mu} = \frac{\lambda}{\mu}\,. \tag{4.1}$$

Scenarios: We consider two cases:

1. $n = \infty$: Poisson distribution (Sec. 4.2),
2. $n < \infty$: Truncated Poisson distribution (Sec. 4.3).

Insensitivity: We shall later see that this 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 as described in Sec. 1.9. They are identical for Erlang's loss model because of the Poisson arrival process (*PASTA*–property: Poisson Arrivals See Time Averages).

4.2 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).

4.2.1 State transition diagram

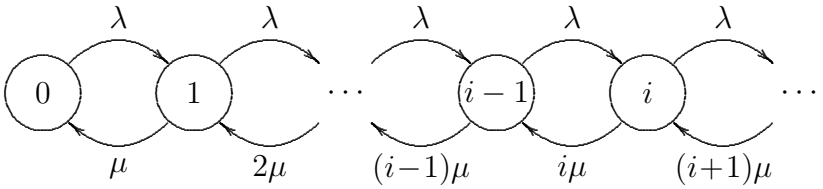


Figure 4.1: *The Poisson distribution. State transition diagram for a system with infinitely many channels, Poisson arrival process (λ), and exponentially distributed holding times (μ).*

We define the state of the system, $[\,i\,]$, as the number of busy channels i ($i = 0, 1, 2, \ldots$). In Fig. 4.1 all states of the system are shown as circles (bubbles), 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 (Sec. 3.2.3), we only have transitions to neighboring states. If we assume the system is in *statistical equilibrium*, then the system will be in state $[\,i\,]$ the proportion of time $p(i)$, 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 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. When i channels are busy, each channel will terminate calls with rate μ so that the total service rate is $i \cdot \mu$ (Palm-Khintchine theorem 3.1). The future development of the traffic process only depends upon the present state, not upon how the process came to this state (the Markov-property).

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 spends in state $[\,i\,]$. The average number of jumps from state $[\,0\,]$ to state $[\,1\,]$ is $\lambda \cdot p(0)$ per time unit, and the average number of jumps from state $[\,1\,]$ to state $[\,0\,]$ is $\mu \cdot p(1)$ per time unit. Thus we have for state $i = 0$:

$$\lambda \cdot p(0) = \mu \cdot p(1)\,, \qquad i = 0\,. \tag{4.2}$$

For state $i > 0$ we get the following *equilibrium or balance equation*:

$$\lambda \cdot p(i-1) + (i+1)\,\mu \cdot p(i+1) = (\lambda + i\,\mu) \cdot p(i)\,, \quad i > 0\,. \tag{4.3}$$

Node equations are always applicable, also for state transition diagrams more dimensions, which we will consider in later chapters.

b. *Cut equations*
In many cases we may exploit a simple structure of the state transition diagram. If for example we put a fictitious cut 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:

$$\lambda \cdot p(i-1) = i\,\mu \cdot p(i)\,, \qquad i = 1, 2, \ldots\,. \tag{4.4}$$

Cut equations are easy to apply for one-dimensional state transition diagrams, whereas node equations are applicable for any diagram.

As the system always will be in some state, we have the normalization restriction:

$$\sum_{i=0}^{\infty} p(i) = 1\,, \qquad p(i) \geq 0\,. \tag{4.5}$$

We notice that node equations (4.3) involve three state probabilities, whereas cut equations (4.4) only involve two. Therefore, it is easier to solve the cut equations. Loss system will always be able to enter statistical equilibrium because we have a limited number of states. We do not specify the mathematical conditions for statistical equilibrium in this chapter.

4.2.2 Derivation of state probabilities

For one-dimensional state transition diagrams the application of cut equations is the most appropriate approach. From Fig. 4.1 we get the following balance equations:

$$\begin{aligned}
\lambda \cdot p(0) &= \mu \cdot p(1)\,, \\
\lambda \cdot p(1) &= 2\,\mu \cdot p(2)\,, \\
\dots &\quad \dots \\
\lambda \cdot p(i-2) &= (i-1)\,\mu \cdot p(i-1)\,, \\
\lambda \cdot p(i-1) &= i\,\mu \cdot p(i)\,, \\
\lambda \cdot p(i) &= (i+1)\,\mu \cdot p(i+1)\,, \\
\dots &\quad \dots\,.
\end{aligned}$$

Expressing all state probabilities by $p(0)$ and introducing the offered traffic $A = \lambda/\mu$ we get:

$$\begin{aligned}
p(0) & & & = & p(0)\,, \\
p(1) & & & = & A \cdot p(0)\,, \\
p(2) &= \frac{A}{2} \cdot p(1) & & = & \frac{A^2}{2} \cdot p(0)\,, \\
\dots & \quad \dots & & \quad \dots & \\
p(i-1) &= \frac{A}{i-1} \cdot p(i-2) & & = & \frac{A^{i-1}}{(i-1)!} \cdot p(0)\,, \\
p(i) &= \frac{A}{i} \cdot p(i-1) & & = & \frac{A^i}{i\,!} \cdot p(0)\,, \\
p(i+1) &= \frac{A}{i+1} \cdot p(i) & & = & \frac{A^{i+1}}{(i+1)!} \cdot p(0)\,, \\
\dots & \quad \dots & & \quad \dots &
\end{aligned}$$

The normalization constraint (4.5) implies:

$$\begin{aligned}
1 &= \sum_{j=0}^{\infty} p(j) \\
&= p(0) \cdot \left\{ 1 + A + \frac{A^2}{2\,!} + \dots + \frac{A^i}{i\,!} + \dots \right\} \\
&= p(0) \cdot \mathrm{e}^{A}\,, \\
p(0) &= \mathrm{e}^{-A}\,.
\end{aligned}$$

Thus the state probabilities become Poisson distributed:

$$p(i) = \frac{A^i}{i\,!} \cdot \mathrm{e}^{-A}\,, \quad i = 0, 1, 2, \ldots\,. \tag{4.6}$$

The number of busy channels at a random point of time is thus Poisson distributed with both mean value (3.37) and variance (3.38) equal to the offered traffic A. We have earlier shown that the number of calls in a fixed time interval also is Poisson distributed (3.36). Thus the Poisson distribution is valid both in time and in space.

We would, of course, obtain the same solution by using node equations.

4.2.3 Traffic characteristics of the Poisson distribution

From a dimensioning point of view, the system with unlimited capacity is of little interest in practise. The traffic characteristics of this system become:

$$\begin{aligned}
&\text{Time congestion:} && E &&= 0\,,\\
&\text{Call congestion:} && B &&= 0\,,\\
&\text{Carried traffic:} && Y &&= \sum_{i=1}^{\infty} i \cdot p(i) = A\,,\\
&\text{Lost traffic:} && A_\ell &&= A - Y = 0\,,\\
&\text{Traffic congestion:} && \mathrm{C} &&= 0\,.
\end{aligned}$$

Only ordered hunting makes sense in this case, and traffic carried by the i'th channel is later given in (4.14).

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 (3.11)). For the Poisson distribution we find (3.37) & (3.38):

$$Z = \frac{\sigma^2}{m_1} = \frac{A}{A} = 1\,. \tag{4.7}$$

The peakedness has dimension [*number of channels*] and is different from the coefficient of variation which has no dimension (2.13).

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. 2.2.7):

$$f_i(t) = (\lambda + i\,\mu)\mathrm{e}^{-(\lambda\,+\,i\,\mu)\,t}\,, \quad t \geq 0\,.$$

Example 4.2.1: Simple Aloha protocol
In example 3.5.2 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 traffic case resulting in a Poisson distribution which can be shown to be valid also for constant holding times. The state probabilities are Poisson distributed (4.6) with $A = \lambda\, h$. 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 utilized efficiently. It has an optimum for $\lambda h = A = 1/2$, where the derivative with respect to A equals zero:

$$\frac{\partial A_{correct}}{\partial A} = \mathrm{e}^{-2\, A} \cdot (1 - 2\, A)\,,$$

$$\max\{A_{correct}\} = \frac{1}{2e} = 0.1839\,. \tag{4.8}$$

We thus obtain a maximum utilization equal to 0.1839 when we offer 0.5 erlang. This is half the value we obtained for a slotted system by synchronizing the satellite transmitters. The models are compared in Fig. 3.6. □

4.3 Truncated Poisson distribution

We still assume *Pure Chance Traffic Type I (PCT-I)* as in Sec. 4.2. 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. 4.2.

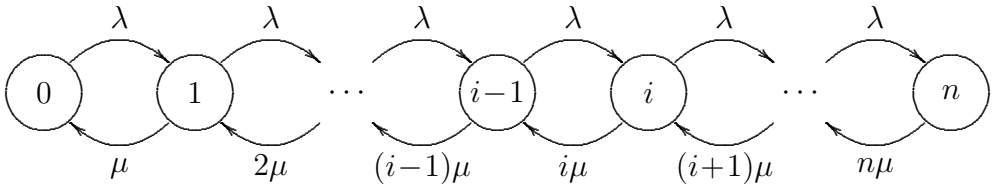


Figure 4.2: *The truncated Poisson distribution. State transition diagram for a system with a limited number of channels (n), Poisson arrival process (λ), and exponential service times (μ).*

4.3.1 State probabilities

We get similar cut equations as for the Poisson case, but the state space is limited to $\{0, 1, \ldots, n\}$ and the normalization condition (4.5) now becomes:

$$p(0) = \left\{ \sum_{j=0}^{n} \frac{A^j}{j!} \right\}^{-1} .$$

We get the so-called *truncated Poisson distribution:*

$$p(i) = \frac{\dfrac{A^i}{i!}}{\displaystyle\sum_{j=0}^{n} \frac{A^j}{j!}} , \qquad 0 \leq i \leq n . \tag{4.9}$$

The name *truncated* means cut-off and is due to the fact that the solution may be interpreted as a conditional Poisson distribution $p(i \,|\, i \leq n)$. This is seen by multiplying both numerator and denominator by e^{-A}. It is not a trivial fact that we are allowed to truncate the Poisson distribution, so that the relative ratios between the state probabilities are unchanged.

4.3.2 Traffic characteristics of Erlang's B-formula

Knowing the state probabilities, we are able to find all 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 (4.9) for $i = n$:

$$E_n(A) = p(n) = \frac{\dfrac{A^n}{n!}}{1 + A + \dfrac{A^2}{2!} + \cdots + \dfrac{A^n}{n!}} . \tag{4.10}$$

This is *Erlang's famous B-formula* (1917, [12]). It is denoted by $E_n(A) = E_{1,n}(A)$, where the fist index refers to the alternative name *Erlang's first formula*.

Call congestion:

The probability that a random call attempt will be lost is equal to the proportion of call attempts blocked. If we consider one time unit, we find by summation over all possible states:

$$B_n(A) = \frac{\lambda \cdot p(n)}{\displaystyle\sum_{i=0}^{n} \lambda \cdot p(i)} = p(n) = E_n(A) . \tag{4.11}$$

The denominator is the average number of call attempts per time unit, and the numerator is the average number of blocked calls per time unit.

Carried traffic:

If we use the cut equation between states $[\,i-1\,]$ and $[\,i\,]$ we get:

$$\begin{aligned} Y_n(A) &= \sum_{i=1}^{n} i \cdot p(i) = \sum_{i=1}^{n} \frac{\lambda}{\mu} \cdot p(i-1) = A \cdot \{1 - p(n)\}\,, \\ Y_n(A) &= A \cdot \{1 - E_n(A)\}\,, \end{aligned} \tag{4.12}$$

where A is the offered traffic. The carried traffic will be less than both A and n.

Lost traffic:

$$A_\ell = A - Y_n(A) = A \cdot E_n(A)\,, \qquad 0 \le A < \infty\,.$$

Traffic congestion:

$$C_n(A) = \frac{A - Y}{A} = E_n(A)\,, \qquad 0 \le Y < n\,.$$

We thus have $E = B = C$ because the arrival intensity λ is independent of the state. This is called the *PASTA*–property, *Poisson Arrivals See Time Averages*, which is valid for all systems with Poisson arrival processes. In all other cases at least two of the three congestion measures will be different. Erlang's B-formula is shown graphically in Fig. 4.3 for some selected values of the parameters.

Traffic carried by the i'th channel (utilization y_i of channel i):

1. *Random hunting and cyclic hunting:* In this case all channels on the average carry the same traffic. The total carried traffic is independent of the hunting strategy and we find the utilization:
$$y_i = y = \frac{Y}{n} = \frac{A\,\{1 - E_n(A)\}}{n}\,. \tag{4.13}$$
This function is shown in Fig. 4.4. We observe that for a given congestion E we obtain the highest utilization 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:
$$y_i = A \cdot \{E_{i-1}(A) - E_i(A)\}\,. \tag{4.14}$$
It should be noticed that the traffic carried by channel i is independent of the number of channels after i in the hunting order. Thus channels after channel i have no influence upon the traffic carried by channel i. There is no feed-back. As the total carried traffic is independent of the hunting mode we have:
$$Y = \sum_{i=1}^{n} y_i\,.$$

Improvement function:

This denotes the increase in carried traffic when the number of channels is increased by one from n to $n+1$:

$$\begin{aligned} F_n(A) &= Y_{n+1} - Y_n\,, \\ &= A\{1 - E_{n+1}\} - A\{1 - E_n\}\,, \qquad (4.15) \\ F_n(A) &= A\left\{E_n(A) - E_{n+1}(A)\right\}\,. \qquad (4.16) \end{aligned}$$

We have that $0 \leq F_n(A) < 1$, as one channel at most can carry one erlang. The improvement function $F_n(A)$ is tabulated in *Moe's Principle* (Arne Jensen, 1950 [59]) and shown in Fig. 4.5. In Sec. 4.8.2 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* (3.11). For the truncated Poisson distribution it can be shown that:

$$Z = Z_n(A) = \frac{\sigma^2}{m} = 1 - A\left\{E_{n-1}(A) - E_n(A)\right\} = 1 - y_n < 1\,, \qquad (4.17)$$

where we have used (4.14). The dimension is [*channels*]. In a group with ordered hunting we may thus estimate the peakedness from observation of the traffic carried by the last channel.

Duration of state $[\,i\,]$*:*

The total intensity for leaving state $[\,i\,]$ is equal to $(\lambda + i\,\mu)$, and therefore the duration of the time in state $[\,i\,]$ (sojourn time) is exponentially distributed with probability density function (pdf):

$$\begin{aligned} f_i(t) &= (\lambda + i\,\mu) \cdot \mathrm{e}^{-(\lambda + i\,\mu)\,t}\,, \quad 0 \leq i < n\,, \\ f_n(t) &= (n\,\mu) \cdot \mathrm{e}^{-(n\,\mu)\,t}\,, \qquad\qquad i = n\,. \qquad (4.18) \end{aligned}$$

The fundamental assumption for the validity of Erlang's B-formula is the Poisson arrival process. According to *Palm-Khintchine theorem* this is fulfilled in ordinary telephone systems with many independent subscribers. As the state probabilities are independent of the holding time distribution, the model is very robust. The combined arrival process and 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.

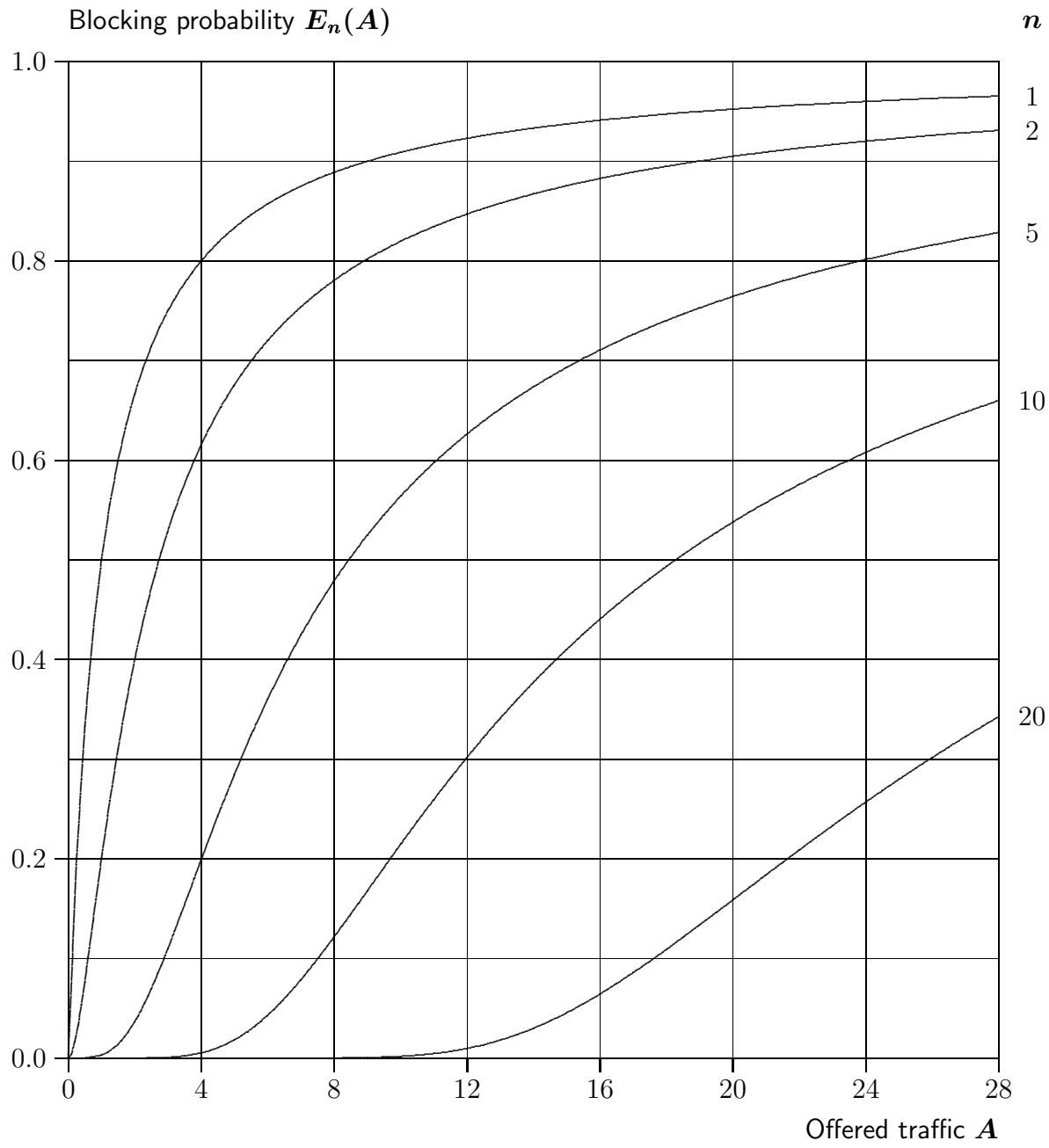


Figure 4.3: *Blocking probability $E_n(A)$ as a function of the offered traffic A for various values of the number of channels n (4.9).*

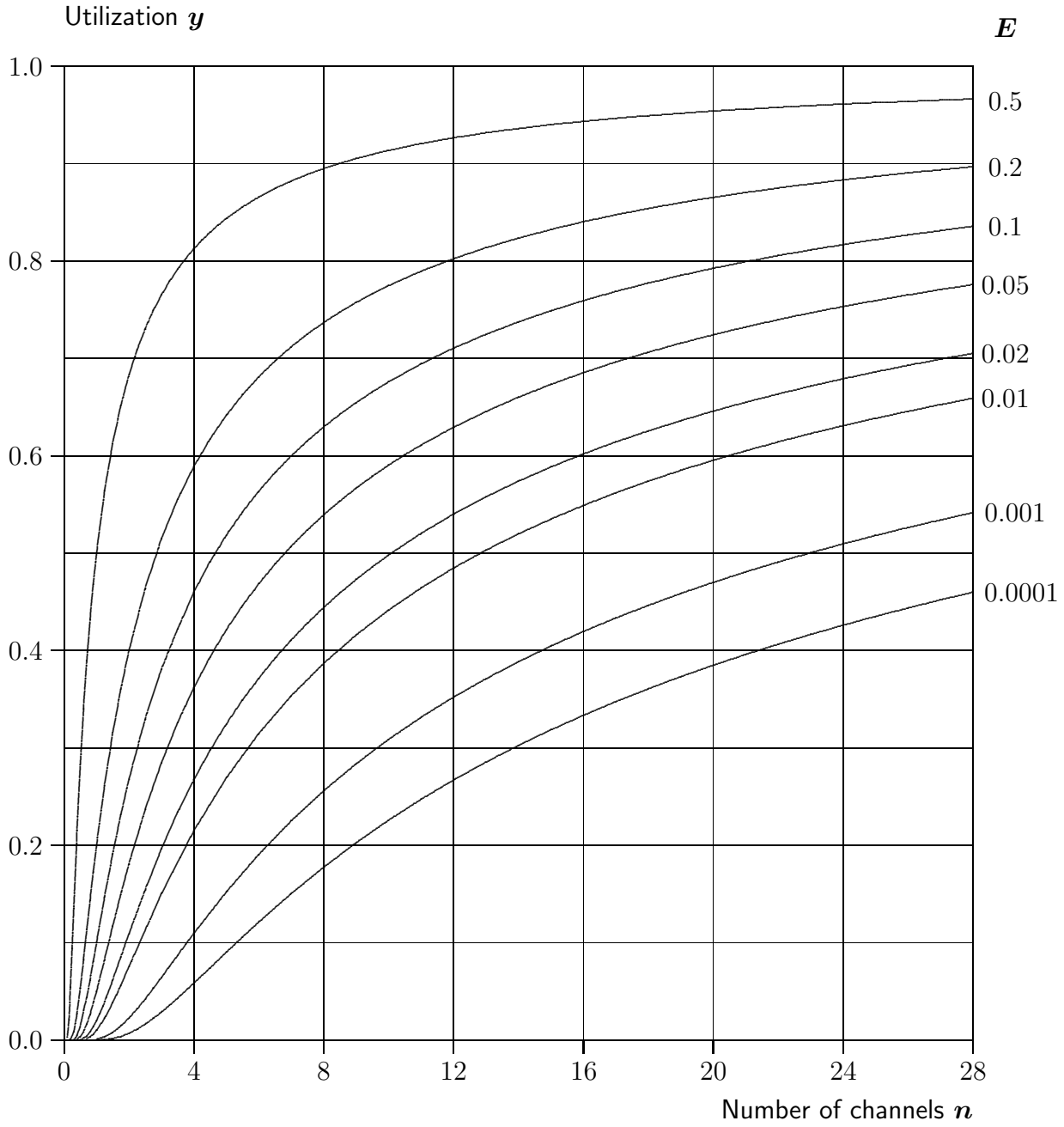


Figure 4.4: *The average utilization per channel y (4.13) as a function of the number of channels n for given values of the congestion E.*

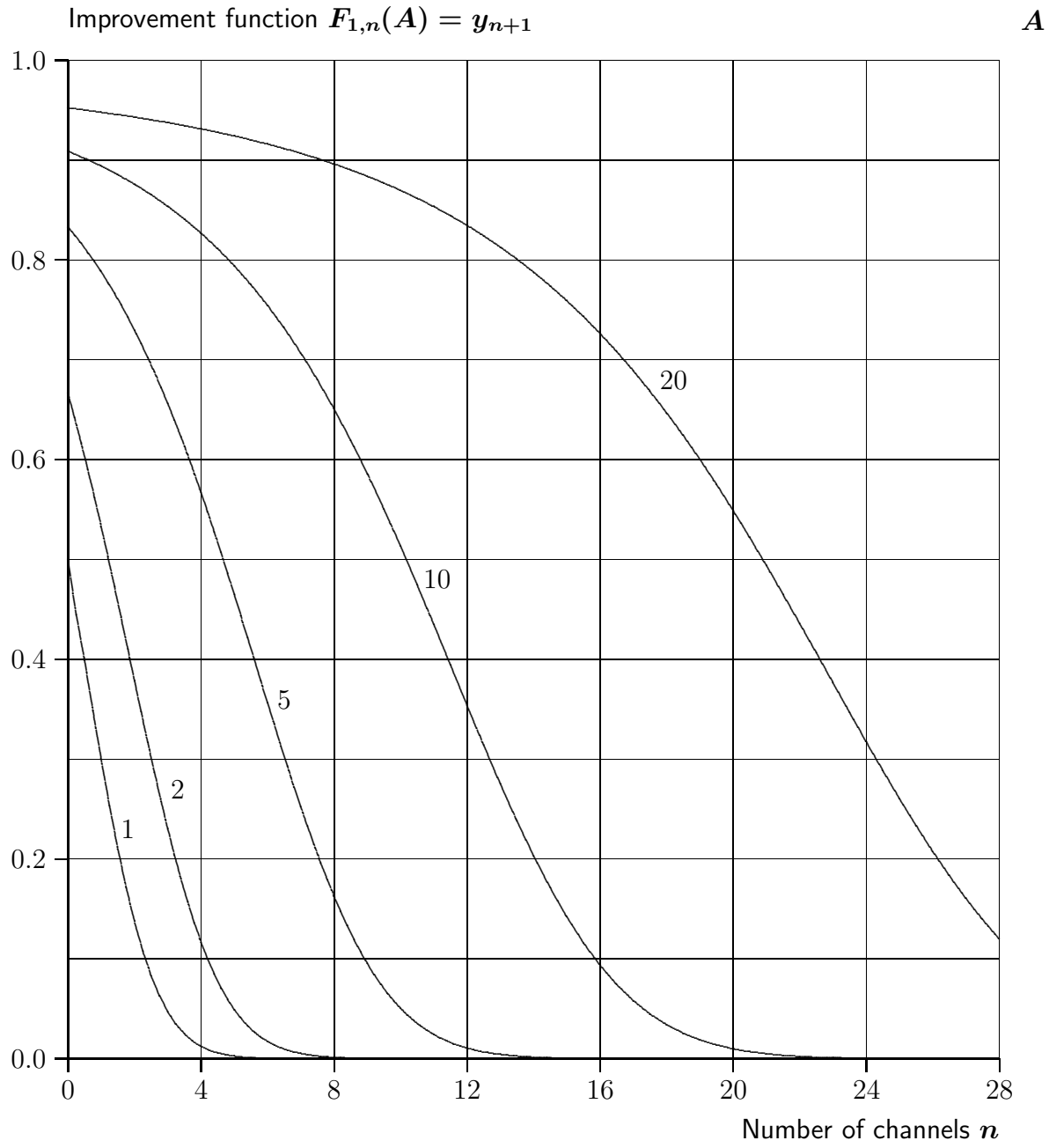


Figure 4.5: *Improvement function* $F_n(A)$ *(4.16) of Erlang's B–formula. By sequential hunting* $F_n(A)$ *equals the traffic* y_{n+1} *carried on channel number* $(n+1)$.

4.4 General procedure for state transition diagrams

The most important tool in teletraffic theory is formulation and solution of models by means of state transition diagrams (bubble diagram). 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 (bubble diagram).

   - Define the states of the system in an unique way,
   - Draw the states as circles (bubble),
   - Consider the states one at a time and draw all possible arrows for transitions away from the state due to:
     (a) the arrival process (new arrival or phase shift in the arrival process),
     (b) the departure (service) process (service time termination or phase shift).

   Remember that only one event takes place at a time. In this way we obtain the complete state transition diagram.

b. Set up the equations describing the system in equilibrium.

   - If the conditions for statistical equilibrium are fulfilled, the steady state equations can be obtained from:
     * node equations (general),
     * 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 normalization.

d. Calculate the performance measures expressed by the state probabilities.

For small values of n we let the non-normalized value of the state probability $q(0)$ equal to one, and then calculate the relative values $q(i), (i = 1, 2, \ldots)$. By normalizing we then find:

$$p(i) = \frac{q(i)}{Q_n}, \quad i = 0, 1, \ldots, n\,, \tag{4.19}$$

where

$$Q_n = \sum_{i=0}^{n} q(i)\,. \tag{4.20}$$

The time congestion becomes:

$$p(n) = \frac{q(n)}{Q_n} = 1 - \frac{Q_{n-1}}{Q_n}\,. \tag{4.21}$$

For large values of n we should use the procedure described below.

4.4.1 Recursion formula

If $q(i)$ becomes very large (e.g. 10^{10}), then we may as an intermediate normalization 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 [111]).

We may normalize the state probabilities after each step which implies more calculations, but ensures a higher accuracy. Let the normalized state probabilities for a system with $x-1$ channels be given by:

$$\underline{P_{x-1}} = \{p_{x-1}(0),\ p_{x-1}(1), \ldots,\ p_{x-1}(x-2),\ p_{x-1}(x-1)\}\,, \quad x = 1, 2, \ldots\,, \tag{4.22}$$

where index $(x-1)$ indicates that we consider state probabilities for a system with $(x-1)$ channels. Let us assume we have the following recursion formula for obtaining $q_x(x)$ from r previous state probabilities (often $r = 1$):

$$q_x(x) = f\big(p_{x-1}(x-1),\ p_{x-1}(x-2), \ldots, p_{x-1}(x-r)\big)\,, \quad x = 1, 2, \ldots\,, \tag{4.23}$$

where $q_x(x)$ will be a relative (non-normalized) state probability. We know the normalized state probabilities for $(x-1)$ channels (4.22), and we want to find the normalized state probabilities for a system with x channels. The *relative* values of state probabilities do not change when we increase number of channels by one, so we get:

$$q_x(i) = \begin{cases} p_{x-1}(i)\,, & i = 0, 1, 2, \ldots, x-1\,, \\ q_x(x)\,, & i = x\,. \end{cases} \tag{4.24}$$

The new normalization constant becomes:

$$Q_x = \sum_{i=0}^{x} q_x(i) = 1 + q_x(x)\,,$$

because in the previous step we normalized the state probabilities ranging from 0 to $x-1$ so they add to one. We thus get:

$$p_x(i) = \begin{cases} \dfrac{p_{x-1}(i)}{1 + q_x(x)}\,, & i = 0, 1, 2, \ldots, x-1\,, \\ \dfrac{q_x(x)}{1 + q_x(x)}\,, & i = x\,. \end{cases} \tag{4.25}$$

The initial value for the recursion is given by $p_0(0) = 1$. The recursion algorithm thus starts with this value, and we find the state probabilities of a system with one channel more by (4.24) and (4.25). The recursion is numerically very stable because we in (4.25) divide with a number greater than one.

Example 4.4.1: Calculating probabilities of the Poisson distribution
We may calculate the Poisson distribution (4.6) by the above approach by starting with class zero and stopping at a state i where for example $q(i) < 10^{-10}$. If we want to calculate the Poisson distribution for very large mean values $m_1 = A = \lambda/\mu$, then we may start with class m by letting $q(m) = 1$, 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 when for example $q(i) < 10^{-10}$ for increasing, respectively decreasing values (or when $i = 0$). We normalize the state probabilities in each step. In this way we avoid calculating many classes with state probability less than 10^{-10}, and we also avoid problems with underflow and overflow. □

Above we calculate all state probabilities. To calculate the time congestion for a loss system we need only store the latest state probability. Let us consider a system with simple birth and death traffic process with arrival rate λ_i and departure rate $i \cdot \mu$ in state i. Then $q_x(x)$ only depends on the previous state probability. By using the cut equation we get the following recursion formula:

$$q_x(x) = \frac{\lambda_{x-1}}{x\,\mu} \cdot p_{x-1}(x-1) = \frac{\lambda_{x-1}}{x\,\mu} \cdot E_{x-1}\,. \tag{4.26}$$

The time congestion for x channels is $E_x = p_x(x)$. Inserting (4.26) into (4.25) we get a simple recursive formula for the time congestion:

$$E_x = \frac{q_x(x)}{1 + q_x(x)} = \frac{\frac{\lambda_{x-1}}{x\,\mu} \cdot E_{x-1}}{1 + \frac{\lambda_{x-1}}{x\,\mu} \cdot E_{x-1}}$$

$$= \frac{\frac{\lambda_{x-1}}{\mu} \cdot E_{x-1}}{x + \frac{\lambda_{x-1}}{\mu} \cdot E_{x-1}}\,, \qquad E_0 = 1\,. \tag{4.27}$$

Introducing the inverse time congestion probability $I_x = E_x^{-1}$ we get:

$$I_x = 1 + \frac{x\,\mu}{\lambda_{x-1}} \cdot I_{x-1}\,, \qquad I_0 = 1\,. \tag{4.28}$$

This is a general recursion formula for calculating time congestion for all systems with state dependent arrival rates λ_i and homogeneous servers.

4.5 Evaluation of Erlang's B-formula

For numerical calculations the formula (4.10) is not very appropriate, since both $n!$ and A^n increase quickly so that overflow in the computer will occur. If we apply (4.27), then we get

the recursion formula:

$$E_x(A) = \frac{A \cdot E_{x-1}(A)}{x + A \cdot E_{x-1}(A)}\,, \qquad E_0(A) = 1\,. \tag{4.29}$$

From a manual calculation point of view, the inverse linear form (4.28) may be simpler:

$$I_x(A) = 1 + \frac{x}{A} \cdot I_{x-1}(A)\,, \qquad I_0(A) = 1\,, \tag{4.30}$$

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 [96]). 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 4.5.1: 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-normalized 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:

$$\textit{Time congestion:} \qquad E_6(2) = p(6) = 0.0121\,.$$

$$\textit{Traffic congestion:} \qquad C_6(2) = \frac{A - Y}{A} = \frac{2 - 1.9758}{2} = 0.0121\,.$$

$$\textit{Call congestion:} \qquad B_6(2) = \{\lambda(6) \cdot p(6)\} \Big/ \left\{ \sum_{i=0}^{6} \lambda(i) \cdot p(i) \right\} = \frac{0.0242}{2.0000} = 0.0121\,.$$

We notice that $E = B = C$ due to the *PASTA*–property.

By applying the recursion formula (4.29) 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 4.5.2: Recursion formula for Erlang-B
The recursion formulæ (4.29) and (4.30) are numerically very stable. For larger values of number of channels n, the initial value $E_0(A)$ in (4.30) has only minor influence. For example, for $A = 20$ erlang and $n = 10$ channels we find with 6 decimals accuracy the same blocking probability independent of whether we start the iteration with the correct value $E_0(A) = 1$ or the erroneous value $E_0(A) = 0$. If we choose $n = 20$ channels, then the first eight decimals are the same. Errors are eliminated when we iterate with increasing n. Upon the other hand, the recursion formula becomes inaccurate if we iterate with decreasing n because errors then accumulate. In general, if a recursion formula is accurate in one direction, then it will be inaccurate in the opposite direction. □

Example 4.5.3: Calculation of $E_x(A)$ for large x
By recursive application of (4.30) we find the inverse blocking probability of the B-formula:

$$\begin{aligned}
I_x(A) &= 1 + \frac{x}{A} + \frac{x\,(x-1)}{A^2} + \ldots + \frac{x(x-1)\ldots(x-j+1)}{A^j} + \ldots + \frac{x!}{A^x} \\
&= \sum_{j=0}^{x} \binom{x}{j} \frac{j!}{A^j}\,,
\end{aligned}$$

For small values of number of channels n we include all terms and get the exact value. For large values of n and A this formula can be applied for fast calculation of the B-formula, because we

may truncate the sum when the terms of summation become very small. This corresponds to use the general recursion formulæ (4.24) and (4.25) for calculating state probabilities (or inverse state probabilities) for decreasing x, starting with state n We get the next state $q(x-1)$ by multiplying the previous state $q(x)$ by $(x-j)/A$, and then normalize $q(n)$ and $q(x-1)$ by $(1+q(x-1))$. At some stage $(x-j) < A$ and the terms start decreasing. We may truncate the summation after $k+1$ terms when for example $q(x-1) < 10^{-10}$. This can be done not only for $I_x(A)$, but also for $E_x(A)$. The truncation level depends on the required accuracy. In this way we avoid calculating many lower states and can control the accuracy. In Example 4.5.2 we were unable to control the accuracy. □

4.6 Properties of Erlang's B-formula

The literature on Erlang-B formula (by mathematician called function) is very comprehensive. In the following we describe the most important aspects for engineering use.

4.6.1 Continued Erlang-B formula

For practical applications of Erlang's B-formula (e.g. Sec. 6.4) we need to generalize Erlang's B-formula to non-integral values of the number of channels x. We define *Erlang's continued or extended B-formula* by:

$$E_x(A) = \frac{A^x \cdot e^{-A}}{\int_A^\infty t^x \cdot e^{-t}\,\mathrm{d}t} \tag{4.31}$$

$$= \frac{A^x \cdot e^{-A}}{\Gamma(x+1,A)}\,. \tag{4.32}$$

where x and A are real numbers and $A > 0$. The incomplete gamma function is defined as:

$$\Gamma(x,A) = \int_A^\infty t^{x-1} \cdot e^{-t}\,\mathrm{d}t\,, \tag{4.33}$$

where A is a non-negative real numbers and x is a real number, including negative values.. The number of channels may be any positive or negative number, the recursion formula (4.29) will still be valid. In Chap. 6 we shall see how we need to work with a negative and non-integral number of channels when evaluating overflow systems.

For integral values of x which we denote by n this can be rewritten as:

$$\begin{aligned}\int_A^\infty t^n \cdot e^{-t}\,\mathrm{d}t &= \int_0^{+\infty} (t+A)^n \cdot e^{-(t+A)}\mathrm{d}t \\ &= e^{-A}\sum_{j=0}^{n}\binom{n}{j}A^j\int_0^{+\infty} t^{n-j}\,e^{-t}\,\mathrm{d}t \\ &= e^{-A}\sum_{j=0}^{n}\frac{n\,!}{j!\,(n-j)!}\cdot A^j\cdot(n-j)! \\ &= n!\cdot e^{-A}\cdot\sum_{j=0}^{n}\frac{A^j}{j\,!}\end{aligned}$$

which inserted in (4.31) yields:

$$\begin{aligned}E_n(A) &= \frac{A^n\cdot e^{-A}}{n!\cdot e^{-A}\cdot\sum_{j=0}^{n}\frac{A^j}{j\,!}} \\ &= \frac{\dfrac{A^n}{n\,!}}{\displaystyle\sum_{j=0}^{n}\frac{A^j}{j\,!}}\,, \quad \text{q.e.d.}\end{aligned}$$

The recursion formula (4.29) will still be valid as we have:

$$\Gamma(x+1,A) = A^x\cdot e^{-A} + x\cdot\Gamma(x,A)\,. \tag{4.34}$$

Example 4.6.1: Erlang-B for non-integral number of channels
The recursion formula for Erlang-B (4.29) is valid for non-integral number of channels. To calculate $E_x(A)$ for any real value of x, we need to find the initial value of $E_{\{x\}}(A)$ for $0 < \{x\} < 1$, where $\{x\}$ is the fractional part of x. If we want to calculate $E_x(A)$ for large non-integral number of channels, then we will get the correct blocking probability by using the initial value $E_{\{x\}}(A) = 1$. For smaller values of x we may use an approximation given in Sec. 4.6.7. To get the exact blocking probability, we have to evaluate the incomplete gamma function in (4.32). □

4.6.2 Insensitivity

We have the following definition of insensitivity:

Insensitivity: A system is insensitive to the holding time distribution if the state probabilities of the system only depend on the mean value of the holding time.

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 (holding time = service time). The state probabilities for both the Poisson distribution (4.6) and the truncated Poisson distribution (4.9) 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.

4.6.3 Derivatives of Erlang-B formula and convexity

The Erlang-B formula is a function of the offered traffic A and the number of channels n which in general may be a real non-negative number. In some cases when we want to optimise systems we need the partial derivatives of Erlang-B formula.

4.6.4 Derivative of Erlang-B formula with respect to A

Erlang's B-formula is given by 4.10:

$$E_n(A) = \frac{\dfrac{A^n}{n!}}{1 + A + \dfrac{A^2}{2!} + \cdots + \dfrac{A^n}{n!}} = \frac{\dfrac{A^n}{n!}}{Q_n},$$

where $\{n, A\} > 0$ are non-negative real numbers, and Q_n denotes the denominator (normalizing constant). We find the derivative with respect to A:

$$\frac{\partial E_n(A)}{\partial A} = \frac{Q_n \cdot \frac{A^{n-1}}{(n-1)!} - \frac{A^n}{n!} \cdot \frac{\partial Q_n}{\partial A}}{Q_n^2} \tag{4.35}$$

where

$$\frac{\partial Q_n}{\partial A} = 1 + \frac{A}{1} + \ldots + \frac{A^{n-1}}{(n-1)!} = Q_{n-1}\,.$$

Thus Q_{n-1} is the normalizing constant of a system with $n-1$ channels.

From the recursion formula for Erlang-B (4.29) we have:

$$E_{x-1} = \frac{n}{A\,\{E_x(A) - 1\}}$$

From (4.35) we then get:

$$\begin{aligned}\frac{\partial E_n(A)}{\partial A} &= \frac{\frac{n}{A}\cdot\frac{A^n}{n!}}{Q_n} - E_n(A)\cdot\frac{Q_{n-1}}{Q_n} \\ &= \frac{n}{A}\cdot E_n(A) - E_n(A)\left\{1 - E_n(A)\right\} \\ \frac{\partial E_n(A)}{\partial A} &= \left(\frac{n}{A} - 1\right)\cdot E_n(A) + E_n(A)^2\,. \end{aligned} \tag{4.36}$$

In a similar way we may obtain higher order derivatives.

4.6.5 Derivative of Erlang-B formula with respect to n

It can be shown that:

$$\frac{\partial E_n(A)}{\partial n} = -E_n(A)^2\cdot A\cdot\int_0^\infty e^{-Ax}(1+x)^n\ln(1+x)\,\mathrm{d}x \tag{4.37}$$

Esteves & Craveirinha & Cardoso (1995 [30]) presents a numerical algorithm for the evaluation of (4.37). In a way similar to (4.29) for the Erlang-B formula, there is a recursive formula to calculate the derivative of order k of the Erlang-B formula for x channels from the value at $x-1$ channels. Let the inverse value of the derivative of order k be denoted by $I_k(A,x)$, we then have:

$$I_k(A,x) = \frac{k}{A}\cdot I_{k-1}(A,x-1) + \frac{x}{A}\cdot I_k(A,x-1)\,,\quad k = 1,2,3,\ldots. \tag{4.38}$$

where $I_0(A,x) = I_x(A)$ given by (4.30). It can be shown that the Erlang-B formula is convex for $n > 1$, as this is equivalent to the following requirement:

$$E_{n-1}(A) - E_n(A) > E_n(A) - E_{n+1}(A)\,. \tag{4.39}$$

If we multiply both sides by A, we observe that this corresponds to $y_n < y_{n+1}$ (4.14) & (4.16), which intuitively is obvious. The first explicit proof of this was given by Messerli (1972 [90]) for integral values of n. Jagers & van Doorn (1986 [56]) show that the Erlang B-formula is convex for all real positive values of the number of trunks. This property is e.g. exploited in Moe's principle (Sec. 4.8.2).

Example 4.6.2: Call admission control with moving window
Erlang's B-formula is valid for arbitrary service times. We may therefore assume that the holding time is equal to a constant h and consider a system with n channels. At an arbitrary instant t all calls accepted during the interval $(t-h,t)$ are still being served. We can at most have n calls being served simultaneously, therefore we may at most accept n calls during $(t-h,t)$. This is valid for any instant. Thus the system at most accepts n calls in any moving window of length h. This mechanism can be applied for control of cell arrival processes in *ATM* systems, i.e. for *CAC* (connection acceptance control). The mechanism works for any arrival process. For a Poisson-arrival process we can calculate the cell loss probability by Erlang's B-formula. □

4.6.6 Inverse Erlang-B formulæ

The inverse formulæ, i.e. n as a function of (A, E) and A as a function of (n, E), may be obtained by means of Newton-Raphson iteration (Szybicki, 1967 [114]).

From a given initial guess (x_0) we calculate i.a. sequence which converges to a fixed value which must satisfy a function $f(x) = 0$:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}\,. \tag{4.40}$$

The following functions should be used:

$$\begin{aligned} A(E,n): \qquad f(A) &= A\left\{E_n(A) - E\right\}\,, \\ A_0 &= \frac{n}{1-E}\,, \\ x(E,A): \qquad f(x) &= E_x(A) - E\,. \end{aligned}$$

The initial value of the number of channels x is chosen so that

$$n_0 - 1 < x \leq n_0, \quad \text{where} \quad E_{n_0}(A) \leq E < E_{n_0-1}(A).$$

The Figs. 4.3 and 4.4 show $E_n(A)$ for various values of the parameters. The derivatives of Erlang's B-formula are given in Sec. 4.2.2.

The numerical problems are also dealt with by Farmer & Kaufman (1978 [31]) and Jagerman (1984 [55]).

Example 4.6.3: Traffic carried by the last channel
In electro-mechanical telephone systems with rotating selectors sequential hunting with homing was often applied, and the quality of service could be monitored by measuring the carried traffic on the last channel (switch) (Brockmeyer, 1957 [10]). As mentioned above the improvement function $F_n(A)$ is equal to the additional traffic carried y_{n+1} when adding an extra channel $(n+1)$ for fixed offered traffic A. We also define the *marginal channel capacity* a_δ, as the additionally traffic carried (in the total system) by adding one channel and keeping the blocking probability E fixed. For $n = 20$ channels and $E = 1\%$ we find $A = 12.0306$ erlang. The above parameters then become: $y_n = 0.0817$ [erlang], $y_{n+1} = 0.0685$ [erlang], $a_\delta = 0.8072$ [erlang] and $y = 0.5955$ erlang. □

4.6.7 Approximations for Erlang-B formula

In the literature various approximations for the continued Erlang-B formula $E_x(A)$, $0 \leq x < 1$, are published. The following properties should be fulfilled:

(a) For integral values $x = n$ $(n = 0, 1, \ldots)$ we obtain the classical Erlang-B formula.

(b) The partial derivatives $\frac{\partial E_x(A)}{\partial x}$ and $\frac{\partial E_x(A)}{\partial A}$ are continuous.

(c) $\lim_{A \to 0} E_x(A) = 0 \quad \text{for} \quad x > 0\,.$

Yngve Rapp (1964 [102]) applies a parabola which satisfy (a) and (b), but not (c):

$$\begin{aligned} E_x(A) &\approx C_0 + C_1 \cdot x + C_2 \cdot x^2\,, \quad \text{where} \\ C_0 &= 1\,, \\ C_1 &= -\frac{A+2}{(1+A)^2 + A}\,, \\ C_2 &= \frac{1}{(1+A)\,((1+A)^2 + A)}\,. \end{aligned} \tag{4.41}$$

Another approximation using a hyperbola is published by Szybicki (1967 [114]). It satisfies (a) and (c), but not (b):

$$E_x(A) \approx \frac{(2-x) \cdot A + A^2}{x + 2A + A^2}\,, \qquad 0 \le x < 1, \tag{4.42}$$

Hedberg, 1981 [39]) has proposed a hyperbola which satisfy all three properties:

$$E_x(A) \approx \frac{1}{1+A} \left\{ \frac{C \cdot (1+C)}{x+C} + A - C \right\}\,, \quad \text{where} \tag{4.43}$$

$$C = \frac{1}{2} \cdot \left\{ A(3+A) + \sqrt{A^2(3+A)^2 + 4A} \right\}\,. \tag{4.44}$$

The above approximations are applicable for $0 \le x < 1$. Approximations to $E_x(A)$ for any value of $x \ge 0$ are developed by Störmer (1963 [112]) and Mejlbro (1994 [89]).

The most accurate values are obtain by using a continued-fraction method of the incomplete gamma function (Lévy-Soussan, 1968 [83]) or by calculating the incomplete gamma function by numerical integration. The extended B-formula can also be defined for negative values of number of trunks.

4.7 Fry-Molina's Blocked Calls Held model

In Fry-Molina's *BCH* (Blocked Calls Held) model (Fry, 1928 [35]), (Molina, 1922 [91], 1927 [92]) a call attempt, which finds all channels busy, will continue to demand service during a time

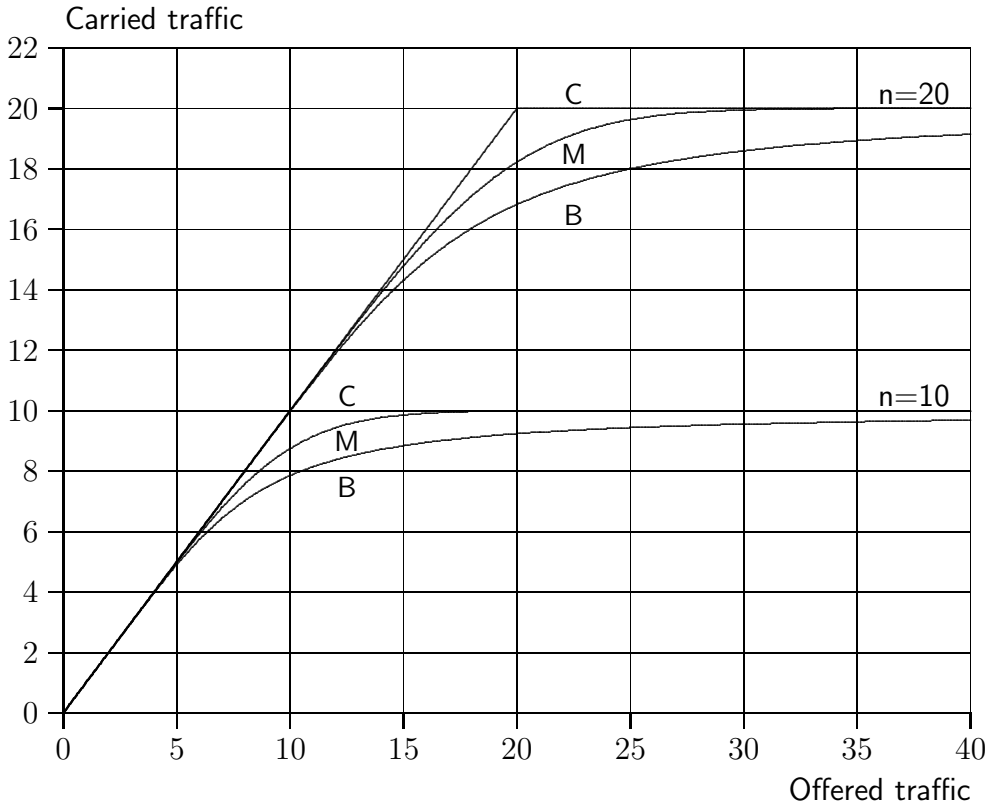


Figure 4.6: *The carried traffic as a function of the offered traffic for Erlang's LCC model (curve B), Fry-Molina's BCH model (curve M) and Erlang's waiting time system (curve C, Chap. 9). By Fry-Molina's model, which corresponds to rearrangement (call packing), we can increase the utilisation as compared with Erlang's B-formula.*

interval, which is equal to the service time it would have obtained, if it was accepted. If a channel becomes idle during this time interval, the call attempt will occupy the channel and keep it busy during the remaining time interval. This model has been applied in North America until the sixties, because is was observed to agree better with the real traffic observations than Erlang's Blocked Calls Cleared model. The explanation to this is maybe that USA for many years was dominated by step-by-step systems, where a blocked call attempt often will be repeated (Lost Call Held).

When applying alternative routing a call attempt, which is blocked on the direct route, will in general be carried on an alternative route, and therefore there will be no repeated call attempt to the direct route (Lost Call Cleared).

The model was already developed by Engset in 1915 in a for many years unknown report (Engset, 1915 [27]). By the introduction of intelligent digital systems with re-arrangement the model has again become of current interest to modelling of e.g. mobile communication systems and service-integrated broadband systems.

Fry-Molina's *BCH*-model is based upon the non-truncated state-dependent Poisson arrival processes, e.g. *BPP*–traffic (Binomial distribution (5.4), Poisson distribution (4.6), and Pascal distribution (5.65)). If we denote the relative state probabilities by $q(i)$ $(i = 0, 1, 2, \ldots)$, then we find the absolute state probabilities by a normalization:

$$p(i) = \frac{q(i)}{Q(\infty)}\,, \quad Q(\infty) = \sum_{i=0}^{\infty} q(i)\,. \tag{4.45}$$

For Fry-Molina's BCH model we get the following state probabilities $p_m(i)$:

$$p_m(i) = \begin{cases} p(i)\,, & 0 \le i < n\,, \\ \displaystyle\sum_{j=n}^{\infty} p(j)\,, & i = n\,. \end{cases} \tag{4.46}$$

The time congestion E is by definition the proportion of time all channels are busy:

$$E = p_m(n) = 1 - Q(n-1)\,. \tag{4.47}$$

The traffic congestion C is from a numerical point of view best obtained in the following way. As the offered traffic A per definition is equal to the traffic carried by in infinite trunk group we have:

$$A = \sum_{i=0}^{\infty} i \cdot p(i)\,. \tag{4.48}$$

The lost traffic is:

$$A_\ell = \sum_{i=n+1}^{\infty} (i-n) \cdot p(i)\,. \tag{4.49}$$

The traffic congestion therefore becomes:

$$C = \frac{A_\ell}{A}\,.$$

4.8 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 characterize 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 that time there were direct connections between all exchanges in the Copenhagen area which resulted in many small and big channel groups. If Erlang's B-formula were applied with a fixed blocking probability for dimensioning these groups, then the utilization in small groups would become low.

Kai Moe (1893–1949), 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 later done by P.A. Samuelson in his famous book, first published in 1947. On the basis of Moe's works the fundamental principles of dimensioning are formulated for telecommunication systems in *Moe's Principle* (Jensen, 1950 [59]).

4.8.1 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 (Cf. B–busy [61]).

$\boldsymbol{n}$	1	2	5	10	20	50	100
$\boldsymbol{A\ (E = 1\%)}$	0.010	0.153	1.361	4.461	12.031	37.901	84.064
$\boldsymbol{y}$	0.010	0.076	0.269	0.442	0.596	0.750	0.832
$\boldsymbol{F_{1,n}(A)}$	0.000	0.001	0.011	0.027	0.052	0.099	0.147
$\boldsymbol{A_1 = 1.2 \cdot A}$	0.012	0.183	1.633	5.353	14.437	45.482	100.877
$\boldsymbol{E}$ [%]	1.198	1.396	1.903	2.575	3.640	5.848	8.077
$\boldsymbol{y}$	0.012	0.090	0.320	0.522	0.696	0.856	0.927
$\boldsymbol{F_{1,n}(A_1)}$	0.000	0.002	0.023	0.072	0.173	0.405	0.617

Table 4.1: *Upper part: For a fixed value of the blocking probability $E = 1\%$ n trunks can be offered the traffic A. The average utilization of the trunks is y, and the improvement function is $F_{1,n}(A)$ (4.16). Lower part: The values of E, y and $F_{1,n}(A)$ are obtained for an overload of 20%.*

Tab. 4.1 shows the offered traffic for a fixed blocking probability $E = 1\%$ for some values of n. The table also gives the average utilization 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. 4.1 two features are observed:

a. The utilisation a per channel is, for a given blocking probability, highest in large groups (Fig. 4.4). At a blocking probability $E = 1$ % a single channel can at most be used 36 seconds per hour on the average!

b. Large channel groups are more sensitive to a given percentage overload than small channel groups. This is explained by the low utilization 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 utilization of the channels.

4.8.2 Improvement principle (Moe's principle)

As mentioned in Sec. 4.8.1 a fixed blocking probability results in a low utilization (bad economy) of small channel groups. If we replace the requirement of a fixed blocking probability with an economic requirement, then the improvement function $F_{1,n}(A)$ (4.16) 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. 4.2 we show the congestion for some values of n and an improvement value $F = 0.05$. We notice from the table that the utilization 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. 4.8. 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.

$\boldsymbol{n}$	1	2	5	10	20	50	100
$\boldsymbol{A\ (F_B = 0.05)}$	0.271	0.607	2.009	4.991	11.98	35.80	78.73
$\boldsymbol{y}$	0.213	0.272	0.387	0.490	0.593	0.713	0.785
$\boldsymbol{E_{1,n}(A)}$ [%]	21.29	10.28	3.72	1.82	0.97	0.47	0.29
$\boldsymbol{A_1 = 1.2 \cdot A}$	0.325	0.728	2.411	5.989	14.38	42.96	94.476
$\boldsymbol{E}$ {%}	24.51	13.30	6.32	4.28	3.55	3.73	4.62
$\boldsymbol{y}$	0.245	0.316	0.452	0.573	0.693	0.827	0.901
$\boldsymbol{F_{1,n}(A_1)}$	0.067	0.074	0.093	0.120	0.169	0.294	0.452

Table 4.2: *For a fixed value of the improvement function we have calculated the same values as in table 4.1.*

This service criteria will therefore in comparison with fixed blocking in Sec. 4.8.1 allocate more channels to large groups and fewer channels to small groups, which is the trend we were

looking for.

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 ($\Delta A/\Delta n = \text{constant}$). 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:

$$c_n = c_0 + c \cdot n \,. \tag{4.50}$$

The total costs for a given number of channels is then (a) cost of cable and (b) cost due to lost traffic (missing income):

$$C_n = g \cdot A\, E_{1,n}(A) + c_0 + c \cdot n \,, \tag{4.51}$$

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. 4.7):

$$C_{n-1} > C_n \quad \text{and} \quad C_n \leq C_{n+1} \,.$$

As $E_{1,n}(A) = E_n(A)$ we get:

$$A\,\{E_{n-1}(A) - E_n(A)\} > \frac{c}{g} \geq A\,\{E_n(A) - E_{n+1}(A)\} \,, \tag{4.52}$$

or:

$$F_{1,n-1}(A) > F_B \geq F_{1,n}(A) \,, \tag{4.53}$$

where:

$$F_B = \frac{c}{g} = \frac{\textit{cost per extra channel}}{\textit{income per extra channel}} \,. \tag{4.54}$$

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:

$$g \cdot A\,\{1 - E_n(A)\} > c_0 + c \cdot n \,. \tag{4.55}$$

Fig. 4.8 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.

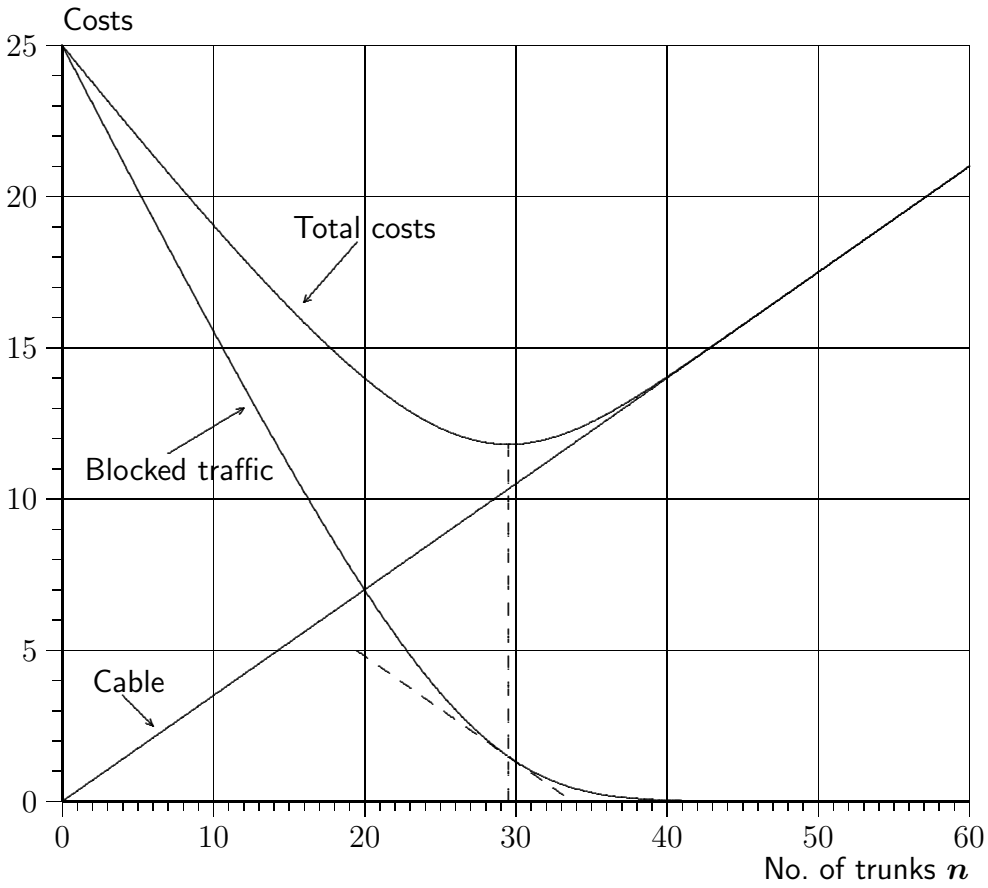


Figure 4.7: *The total costs are composed of costs for cable and lost income due to blocked traffic (4.51). Minimum of the total costs are obtained when (4.52) is fulfilled, i.e. when the two cost functions have the same slope with opposite signs (difference quotient). ($F_B = 0.35$, $A = 25$ erlang). Minimum is obtained for $n = 30$ trunks.*

In Denmark the following values have been used:

$$\begin{aligned} F_B &= 0.35 \quad \text{for primary trunk groups.} \\ F_B &= 0.20 \quad \text{for service protecting primary groups.} \\ F_B &= 0.05 \quad \text{for groups with no alternative route.} \end{aligned} \tag{4.56}$$

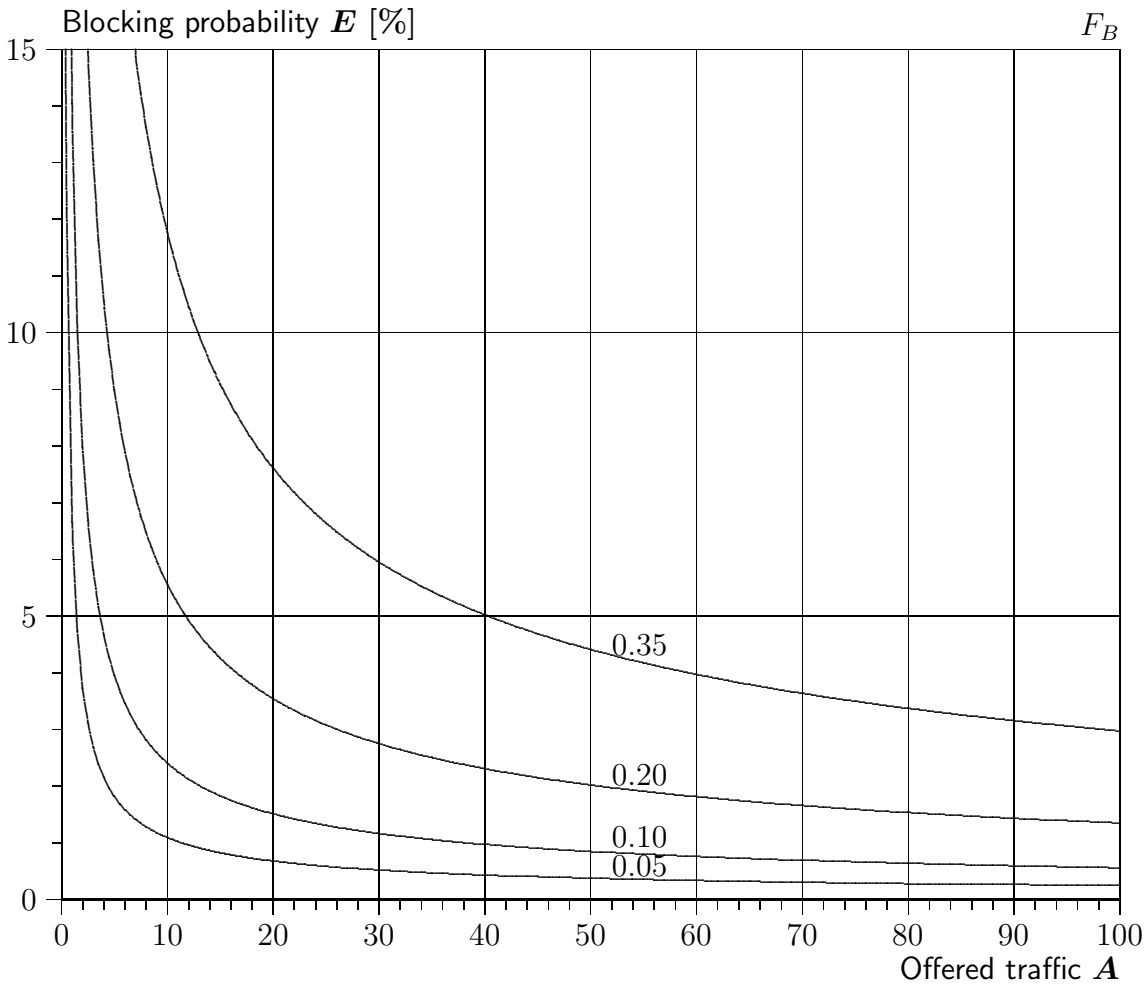


Figure 4.8: *When dimensioning with a fixed value of the improvement value F_B the blocking probabilities for small values of the offered traffic become large (cf. Tab. 4.2).*

Chapter 5

Loss systems with full accessibility

In this chapter we generalize 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) model: *Palm–Wallström's model.*

Erlang's model describes random traffic. Engset's model describes traffic which is more smooth than random traffic. The Negative Binomial model describes traffic which is more bursty than random traffic and includes models with Pareto-distributed inter-arrival times (heavy-tailed traffic) and traffic with batch arrivals. These models are all insensitive to the service time distribution. Engset and Pascal models are even insensitive to the distribution of the idle time of sources. It is important always to use *traffic congestion* as the most important performance metric.

After the introduction in Sec. 5.1 we go through the basic classical theory. In Sec. 5.2 we consider the Binomial case, where number of sources S (subscribers, customers, jobs) is limited and 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. 4.2). We consider the strategy *Blocked-Calls-Cleared (BCC)*. In Sec. 5.3 we restrict the number of channels so that it becomes less than the number of sources ($n < S$). We may then experience 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 probabilities 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. This is derived in a similar way as the

recursion formula for Erlang's B-formula. Also a formula recursive in number of sources S, and a formula simultaneously recursive in both n and S are derived.

In Sec. 5.6 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. 5.7). Finally, in Sec. 5.8 we consider a Batch Poisson arrival process and show it is similar to the Pascal case.

5.1 Introduction

We consider a system with same structure (full accessibility and homogeneous group) and strategy (Lost-Calls-Cleared) as in Chap. 4, 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. All processes considered in Chapter 4 and 5 are *insensitive* to the service time distribution, i.e. only the mean service time is of importance to the state probabilities. The service time distribution itself has no influence.

Definition of offered traffic: In Sec. 1.7 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 this definition is equivalent to the average number of calls attempts per mean service time. In Engset and Pascal cases the arrival processes are not renewal processes as the mean inter-arrival time depends on the actual state.

Carried traffic is by definition the mean value of the state probabilities (average number of busy channels).

Peakedness is defined as the ratio between variance and mean value of the state probabilities. For 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. 4:

1. *Erlang-case* (P – Poisson-case):
   The arrival process is a Poisson process with intensity λ. This type of traffic is called *Random Traffic* (RT) or *Pure Chance Traffic type One* (PCT–I). We consider two cases:

   a. $n = \infty$: Poisson distribution (Sec. 4.2).
      The peakedness is in this case equal to one: $Z = 1$.

b. $n < \infty$: Truncated Poisson distribution (Sec. 4.3).

2. *Engset-case* (B – Binomial-case):
There is a limited number of sources S. Each 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 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. 5.2).
   In this case the peakedness is less than one: $Z<1$.

   b. $n < S$: Truncated Binomial distribution (Sec. 5.3).

3. *Palm-Wallström–case* (P – Pascal-case):
There is a limited number of sources 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. 5.6).
   In this case peakedness is greater than one: $Z>1$.

   b. $n < \infty$: Truncated Pascal distribution (truncated negative Binomial distribution) (Sec. 5.7).

As the Poisson process may be obtained by an infinite number of sources with a limited total arrival intensity λ, the Erlang-case may be considered as a special case of the two other cases:

$$\lim_{S\to\infty,\ \gamma\to 0}(S\ \pm\ i)\,\gamma = \lim_{S\to\infty,\ \gamma\to 0} S\,\gamma = \lambda\,.$$

For any state $0 \leq i \leq n$ (n finite) we then have a constant arrival intensity λ. This is also seen from Palm-Khintchine theorem (Sec. 3.6.1).

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 modeling 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 performance parameters for loss systems are time congestion E, Call congestion B, traffic congestion C, and the utilization of the channels. Among these, traffic congestion C is the most important characteristic. These measures are derived for each of the above-mentioned models.

5.2 Binomial Distribution

We consider a system with a limited number of sources S. Sources is a generic term for subscribers, users, terminals, etc. The individual source alternates 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. 5.2). This kind of sources are called *sporadic sources* or *on/off* sources. This type of traffic is called *Pure Chance Traffic type Two* (*PCT–II*), or *pseudo-random traffic*.

Figure 5.1: *A full accessible loss system with S sources, which generates traffic to n channels. The system is shown by a so-called chicko-gram. The beak of a source symbolizes a selector which points upon the channels (servers) among which the source may choose.*

In this section the number of channels (trunks, servers) n is 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 [49]).

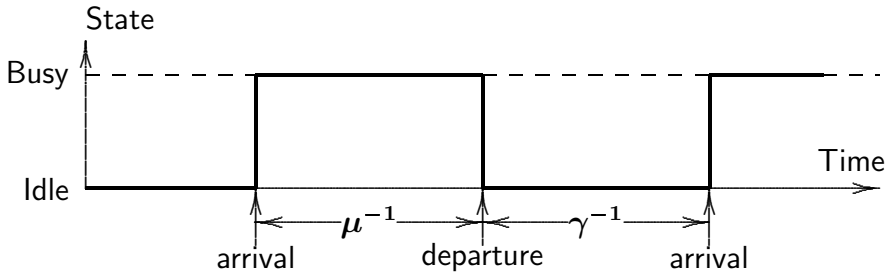


Figure 5.2: *Every individual source is either idle or busy, and behaves independent of all other sources.*

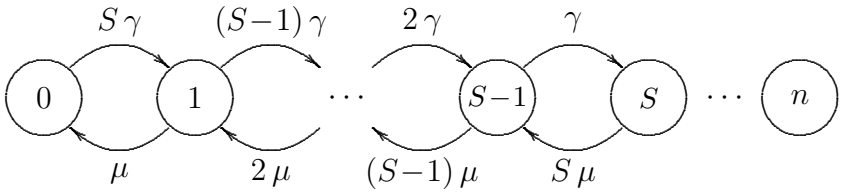


Figure 5.3: *State transition diagram for the Binomial case (Sec. 5.2). The number of sources S is less than or equal to the number of channels n ($S \leq n$).*

5.2.1 Equilibrium equations

We are interested in the steady state probabilities $p(i)$, which are the proportion of time the process spends in state $[\,i\,]$. Our calculations are based on the state transition diagram shown in Fig. 5.3. We consider cuts between neighboring states and find:

$$\left.\begin{aligned} S\,\gamma \cdot p(0) &= \mu \cdot p(1)\,, \\ (S-1)\,\gamma \cdot p(1) &= 2\mu \cdot p(2)\,, \\ \cdots &\quad \cdots \\ (S-(i-1))\,\gamma \cdot p(i-1) &= i\,\mu \cdot p(i)\,, \\ (S-i)\,\gamma \cdot p(i) &= (i+1)\mu \cdot p(i+1)\,, \\ \cdots &\quad \cdots \\ 1\,\gamma \cdot p(S-1) &= S\mu \cdot p(S)\,. \end{aligned}\right\} \tag{5.1}$$

All state probabilities are expressed by $p(0)$:

$$\begin{aligned} p(1) &= \frac{S\,\gamma}{\mu} \cdot p(0) &&= p(0) \cdot \binom{S}{1} \cdot \left(\frac{\gamma}{\mu}\right)^1\,, \\ p(2) &= \frac{(S-1)\,\gamma}{2\,\mu} \cdot p(1) &&= p(0) \cdot \binom{S}{2} \cdot \left(\frac{\gamma}{\mu}\right)^2\,, \\ \cdots &\quad \cdots &&\quad \cdots \\ p(i) &= \frac{(S-(i-1))\,\gamma}{i\,\mu} \cdot p(i-1) &&= p(0) \cdot \binom{S}{i} \cdot \left(\frac{\gamma}{\mu}\right)^i\,, \\ p(i+1) &= \frac{(S-i)\,\gamma}{(i+1)\,\mu} \cdot p(i) &&= p(0) \cdot \binom{S}{i+1} \cdot \left(\frac{\gamma}{\mu}\right)^{i+1}\,, \\ \cdots &\quad \cdots &&\quad \cdots \\ p(S) &= \frac{\gamma}{S\,\mu} \cdot p(S-1) &&= p(0) \cdot \binom{S}{S} \cdot \left(\frac{\gamma}{\mu}\right)^S\,. \end{aligned}$$

The total sum of all probabilities must be equal to one:

$$\begin{aligned} 1 &= p(0) \cdot \left\{ 1 + \binom{S}{1} \cdot \left(\frac{\gamma}{\mu}\right)^1 + \binom{S}{2} \cdot \left(\frac{\gamma}{\mu}\right)^2 + \cdots + \binom{S}{S} \cdot \left(\frac{\gamma}{\mu}\right)^S \right\} \\ &= p(0) \cdot \left\{ 1 + \frac{\gamma}{\mu} \right\}^S\,, \end{aligned}$$

where we have used Newton's Binomial expansion. By letting $\beta = \gamma/\mu$ we get:

$$p(0) = \frac{1}{(1+\beta)^S}\,. \tag{5.2}$$

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:

$$\begin{aligned} p(i) &= \binom{S}{i} \cdot \beta^i \cdot \frac{1}{(1+\beta)^S} \\ &= \binom{S}{i} \cdot \left(\frac{\beta}{1+\beta}\right)^i \cdot \left(\frac{1}{1+\beta}\right)^{S-i}, \quad i = 0, 1, \ldots, S, \quad 0 \leq S \leq n, \end{aligned}$$

which is the Binomial distribution (Tab. 3.1). Finally, we get by introducing the offered traffic per source a, defined as the traffic carried per source when there is no blocking:

$$a = \frac{\beta}{1+\beta} = \frac{\gamma}{\mu+\gamma} = \frac{1/\mu}{1/\gamma + 1/\mu}, \tag{5.3}$$

$$p(i) = \binom{S}{i} \cdot a^i \cdot (1-a)^{S-i}, \quad i = 0, 1, \ldots, S, \quad 0 \leq S \leq n. \tag{5.4}$$

In this case a call attempt from an idle source is never blocked, and the carried traffic α per source is equal to the offered traffic per source a, and this is 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. 5.2, 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.

The *Binomial distribution* obtained in (5.4) is in teletraffic theory sometimes called the Bernoulli distribution, but this should be avoided as we in statistics use this name for a two-point distribution.

Example 5.2.1: Binomial distribution and convolution
Formula (5.4) can be derived by elementary considerations. All subscribers can be split into two classes: idle subscribers and busy subscribers. The probability that an arbitrary subscriber is *busy* is $y = a$, which is independent of the state of all other subscribers as the system has no blocking and call attempts always are accepted. Then the state of a single source is given by the Bernoulli distribution:

$$p_1(i) = \begin{cases} 1-a, & i = 0, \\ a, & i = 1. \end{cases} \tag{5.5}$$

which has a finite mean value a. If we in total have S subscribers (sources), then the probability $p_S(i)$ that i sources are busy at an arbitrary instant is given by the Binomial distribution ((5.4) & Tab. 3.1):

$$p_S(i) = \binom{S}{i} a^i (1-a)^{S-i}, \qquad \sum_{i=0}^{S} p_S(i) = 1, \tag{5.6}$$

which has the mean value $S \cdot a$. If we add one source more to the system, then the distribution of the total number of busy sources is obtained by convolution of the Binomial distribution (5.6) and

the Bernoulli distribution (5.5):

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

□

5.2.2 Traffic characteristics of Binomial traffic

We summarize definitions of parameters given above:

$$\gamma = \text{call intensity per idle source,} \tag{5.7}$$

$$1/\mu = \text{mean service (holding) time,} \tag{5.8}$$

$$\beta = \gamma/\mu = \text{offered traffic per idle source.} \tag{5.9}$$

By definition, the offered traffic of a source is equal to the carried traffic in a system with no congestion, where the source freely alternates between states *idle* and *busy*. Therefore, we have the following definition of the offered traffic:

$$a = \frac{\beta}{1+\beta} = \text{offered traffic per source,} \tag{5.10}$$

$$A = S\cdot a = S\cdot\frac{\beta}{1+\beta} = \text{total offered traffic,} \tag{5.11}$$

$$\alpha = \text{carried traffic per source} \tag{5.12}$$

$$Y = S\cdot\alpha = \text{total carried traffic} \tag{5.13}$$

$$y = Y/n = \text{carried traffic per channel with random hunting} \tag{5.14}$$

Offered traffic per source is a difficult concept to deal with because the proportion of time a source is idle depends on the congestion. The number of calls offered by a source depends on the number of channels (feed-back): a high congestion results in more idle time for a source and thus in more call attempts.

Time congestion:

$$\begin{aligned} E &= 0\,, & S < n\,, \\ E &= p(n) = a^n\,, & S = n\,. \end{aligned}$$

Carried traffic:

$$\begin{aligned} Y &= S \cdot \alpha = \sum_{i=0}^{S} i \cdot p(i) \\ &= S \cdot a = A\,, \end{aligned} \tag{5.15}$$

which is the mean value of the Binomial distribution (5.4). In this case with no blocking we of course have $a = \alpha$ and:

Traffic congestion:

$$C = \frac{A - Y}{A} = 0\,. \tag{5.16}$$

Number of call attempts per time unit:

$$\begin{aligned} \Lambda &= \sum_{i=0}^{S} p(i) \cdot (S - i)\,\gamma \\ &= \gamma\,S - \gamma \cdot \sum_{i=0}^{S} i \cdot p(i) = \gamma\,S - \gamma\,S\,a \\ &= S\,(1 - \alpha) \cdot \gamma\,, \end{aligned}$$

where $S\,(1 - \alpha)$ is the average number of idle sources.

As all call attempts are accepted we get:
Call congestion:

$$B = 0\,. \tag{5.17}$$

Traffic carried by channel i:

Random hunting:

$$y = \frac{Y}{n} = \frac{S \cdot \alpha}{n}\,. \tag{5.18}$$

Sequential hunting: complex expression derived by L.A. Joys (1971 [65]).

Improvement function:

$$F_n(A) = Y_{n+1} - Y_n = 0\,. \tag{5.19}$$

Peakedness of the Binomial distribution is (Tab. 3.1):

$$Z = \frac{\sigma^2}{m_1} = \frac{S \cdot a \cdot (1-a)}{S \cdot a},$$

$$Z = 1 - a = \frac{1}{1+\beta} < 1. \tag{5.20}$$

We observe that the peakedness $Z = 1-a$ is independent of the number of sources and always less than one as $a > 0$ for the binomial case. Therefore it corresponds to *smooth traffic*.

Duration of state i: This is exponentially distributed with rate:

$$\gamma(i) = (S-i) \cdot \gamma + i \cdot \mu, \qquad 0 \le i \le S \le n. \tag{5.21}$$

Finite source traffic is characterized by number of sources S and offered traffic per idle source β. Alternatively, we often use offered traffic A and peakedness Z. From (5.11) and (5.20) we get the following relations between the two set of parameters (A, Z) and (S, β):

$$A = S \cdot \frac{\beta}{1+\beta}, \tag{5.22}$$

$$Z = \frac{1}{1+\beta}, \tag{5.23}$$

and by solving these equations with respect to β and S we get:

$$\beta = \frac{1-Z}{Z}, \tag{5.24}$$

$$S = \frac{A}{1-Z}. \tag{5.25}$$

5.3 Engset distribution

The only difference in comparison with Sec. 5.2 is that number of sources S now is greater than or equal to number of trunks (channels), $S \ge n$. Therefore, call attempts may experience congestion.

5.3.1 State probabilities

The cut equations are identical to (5.1), but they only exist for $0 \le i \le n$ (Fig. 5.4). The normalization equation becomes:

$$1 = p(0) \cdot \left\{ 1 + \binom{S}{1} \cdot \left(\frac{\gamma}{\mu}\right) + \cdots + \binom{S}{n} \cdot \left(\frac{\gamma}{\mu}\right)^n \right\}.$$

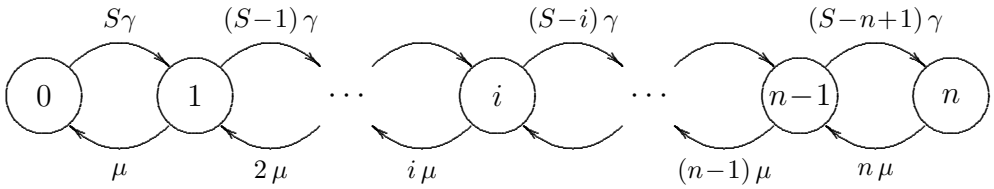


Figure 5.4: *State transition diagram for the Engset case with $S > n$, where S is the number of sources and n is the number of channels.*

From this we obtain $p(0)$, and by letting $\beta = \gamma/\mu$ the state probabilities become:

$$p(i) = \frac{\binom{S}{i} \cdot \beta^i}{\sum_{j=0}^{n} \binom{S}{j} \cdot \beta^j}\,, \qquad 0 \le i \le n\,. \tag{5.26}$$

In the same way as above we may by using (5.10) rewrite this expression to a form, which is analogue to (5.4):

$$p(i) = \frac{\binom{S}{i} \cdot a^i \cdot (1-a)^{S-i}}{\sum_{j=0}^{n} \binom{S}{j} \cdot a^j \cdot (1-a)^{S-j}}\,, \qquad 0 \le i \le n\,, \tag{5.27}$$

from which we directly observe why it is called a *truncated Binomial distribution* (cf. truncated Poisson distribution (4.10)). The distribution (5.26) & (5.27) is called the *Engset–distribution* after the Norwegian T. Engset (1865–1943) who first published the model with a finite number of sources (1918 [28]).

5.3.2 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 characterized by the parameters

1. $\beta = \gamma/\mu$ = offered traffic per idle source,
2. S = number of sources, and
3. n = number of channels.

Time congestion E: this is by definition equal to the proportion of time the system is blocking new call attempts, i.e. $p(n)$ (5.26):

$$E_{n,S}(\beta) = p(n) = \frac{\binom{S}{n} \cdot \beta^n}{\sum_{j=0}^{n} \binom{S}{j} \cdot \beta^j}, \qquad S \geq n\,. \tag{5.28}$$

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:

$$\begin{aligned} B_{n,S}(\beta) &= \frac{p(n) \cdot (S-n)\,\gamma}{\sum_{j=0}^{n} p(j) \cdot (S-j)\,\gamma} \\ &= \frac{\binom{S}{n} \cdot \beta^n \cdot (S-n)\,\gamma}{\sum_{j=0}^{n} \binom{S}{j} \cdot \beta^j \cdot (S-j)\,\gamma}\,. \end{aligned}$$

Using

$$\binom{S}{i} \cdot \frac{S-i}{S} = \binom{S-1}{i},$$

we get:

$$\begin{aligned} B_{n,S}(\beta) &= \frac{\binom{S-1}{n} \cdot \beta^n}{\sum_{j=0}^{n} \binom{S-1}{j} \cdot \beta^j}, \\ B_{n,S}(\beta) &= E_{n,S-1}(\beta)\,, \quad S \geq n\,. \end{aligned} \tag{5.29}$$

This result may be interpreted as follows. The probability that a call attempt from a random idle source (subscriber) is blocked 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 5.1 Arrival-theorem: *For full accessible systems with a limited number of sources, a random source upon arrival will observe the state of the system as if the source itself does not belong to the system.*

The *PASTA*–property is included in this case because an infinite number of sources less one is still an infinite number.

Carried traffic: By applying the cut equation between state $[\,i-1\,]$ and state $[\,i\,]$ we get:

$$\begin{aligned} Y &= \sum_{i=1}^{n} i \cdot p(i) && (5.30)\\ &= \sum_{i=1}^{n} \frac{\gamma}{\mu} \cdot (S-i+1) \cdot p(i-1) \\ &= \sum_{i=0}^{n-1} \beta \cdot (S-i) \cdot p(i) && (5.31)\\ &= \sum_{i=0}^{n} \beta \cdot (S-i) \cdot p(i) - \beta \cdot (S-n) \cdot p(n)\,, \\ Y &= \beta \cdot (S-Y) - \beta \cdot (S-n) \cdot E\,, && (5.32) \end{aligned}$$

as $E = E_{n,S}(\beta) = p(n)$. This is solved with respect to Y:

$$Y = \frac{\beta}{1+\beta} \cdot \{S - (S-n) \cdot E\}\ . \tag{5.33}$$

Traffic congestion $C = C_{n,S}(A)$. This is the most important congestion measure. The offered traffic is given by (5.22) and we get:

$$\begin{aligned} C &= \frac{A-Y}{A} \\ &= \frac{\dfrac{S\,\beta}{1+\beta} - \dfrac{\beta}{1+\beta} \cdot \{S - (S-n) \cdot E\}}{\dfrac{S\,\beta}{1+\beta}}\,, \\ C &= \frac{S-n}{S} \cdot E\,. && (5.34) \end{aligned}$$

We may also find the carried traffic if we know the call congestion B. The number of accepted call attempts from a source which on the average is idle $1/\gamma$ time unit before it generate one call attempt is $1 \cdot (1-B)$, and each accepted call has an average duration $1/\mu$. Thus the carried traffic per source, i.e. the proportion of time the source is busy, becomes:

$$\alpha = \frac{(1-B)/\mu}{1/\gamma + (1-B)/\mu}\,.$$

The total carried traffic becomes:

$$Y = S \cdot \alpha = S \cdot \frac{\beta\,(1-B)}{1+\beta\,(1-B)}\,. \tag{5.35}$$

Equalizing the two expressions for the carried traffic (5.33) & (5.35) we get the following relation between E and B:

$$E = \frac{S}{S-n} \cdot \frac{B}{1+\beta(1-B)}\,. \tag{5.36}$$

Number of call attempts per time unit:

$$\begin{aligned} \Lambda &= \sum_{i=0}^{n} p(i) \cdot (S-i)\,\gamma \\ \Lambda &= (S-Y) \cdot \gamma\,, \end{aligned} \tag{5.37}$$

where Y is the carried traffic (5.30). Thus $(S-Y)$ is the average number of idle sources, which is evident.

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). With few channels available many call attempts are blocked and the sources are idle a higher proportion of the time and thus generate more call attempts per time unit.

Lost traffic:

$$\begin{aligned} A_\ell &= A \cdot C \\ &= S\,\frac{\beta}{1+\beta} \cdot \frac{S-n}{S}\,E \\ &= \frac{(S-n)\,\beta}{1+\beta} \cdot E\,. \end{aligned} \tag{5.38}$$

Duration of state i: This is exponentially distributed with intensity:

$$\left.\begin{aligned} \gamma(i) &= (S-i)\cdot\gamma + i\cdot\mu\,, && 0 \le i < n\,, \\ \gamma(n) &= n\,\mu\,, && i = n\,. \end{aligned}\right\} \tag{5.39}$$

Improvement function:

$$F_{n,S}(A) = Y_{n+1} - Y_n\,. \tag{5.40}$$

Example 5.3.1: Call average and time average
Above we have under the assumption of statistical equilibrium defined the state probabilities $p(i)$

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:

$$\pi_{n,S,\beta}(i) = \frac{(S-i)\,\gamma \cdot p(i)}{\sum_{j=0}^{n}(S-j)\,\gamma \cdot p(j)}\,, \quad i = 0,1,\ldots n\,. \tag{5.41}$$

In a way analogue to the derivation of (5.29) we may show that in agreement with the arrival theorem (Theorem 5.1) we have as follows:

$$\pi_{n,S,\beta}(i) = p_{n,S-1,\beta}(i)\,, \quad i = 0,1,\ldots,n\,. \tag{5.42}$$

When a source leaves the system and looks back, it observes the system in state $[\,i-1\,]$ with probability:

$$\psi_{n,S,\beta}(i-1) = \frac{i\,\mu \cdot p(i)}{\sum_{j=1}^{n} j\,\mu \cdot p(j)}\,, \quad i = 1,2,\ldots,n\,. \tag{5.43}$$

By applying cut equations we immediately get that this is identical with (5.41), 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 time runs forward or backward. □

5.4 Relations between E, B, and C

From (5.36) we get the following relation between $E = E_{n,S}(\beta)$ and $B = B_{n,S}(\beta) = E_{n,S-1}(\beta)$:

$$E = \frac{S}{S-n} \cdot \frac{B}{1+\beta(1-B)} \quad \text{or} \quad \frac{1}{E} = \frac{S-n}{S}\left\{(1+\beta)\cdot\frac{1}{B} - \beta\right\}\,, \tag{5.44}$$

$$B = \frac{(S-n)\cdot E\cdot(1+\beta)}{S+(S-n)\cdot E\cdot\beta} \quad \text{or} \quad \frac{1}{B} = \frac{1}{1+\beta}\left\{\frac{S}{S-n}\cdot\frac{1}{E} + \beta\right\}\,. \tag{5.45}$$

The expressions to the right-hand side are linear in the reciprocal blocking probabilities. In (5.34) we obtained the following simple relation between C and E:

$$C = \frac{S-n}{S}\cdot E\,, \tag{5.46}$$

$$E = \frac{S}{S-n}\cdot C\,. \tag{5.47}$$

If we in (5.46) express E by B (5.44), then we get C expressed by B:

$$C = \frac{B}{1+\beta\cdot(1-B)}\,, \tag{5.48}$$

$$B = \frac{(1+\beta)\,C}{1+\beta\,C}\,. \tag{5.49}$$

This relation between B and C is general for any system and may be derived from carried traffic as follows. The carried traffic Y corresponds to $(Y\cdot\mu)$ accepted call attempts per time unit. The average number of idle sources is $(S-Y)$, so the average number of call attempts per time unit is $(S-Y)\cdot\gamma$ (5.37). The call congestion is the ratio between the number of rejected call attempts and the total number of call attempts, both per time unit:

$$B = \frac{(S-Y)\,\gamma - Y\cdot\mu}{(S-Y)\,\gamma}$$

$$= \frac{(S-Y)\,\beta - Y}{(S-Y)\,\beta}\,.$$

By definition, $Y = A\,(1-C)$ and from (5.22) we have $S = A\,(1+\beta)/\beta$. Inserting this we get:

$$B = \frac{A(1+\beta) - A(1-C)\,\beta - A\,(1-C)}{A(1+\beta) - A(1-C)\,\beta}$$

$$B = \frac{(1+\beta)\,C}{1+\beta\,C} \qquad \text{q.e.d.}$$

From the last equation we see that for small values of the traffic congestion C $(1+\beta\,C\approx 1)$ the traffic congestion is Z (peakedness value) times bigger than the call congestion:

$$C \approx \frac{B}{1+\beta} = Z\cdot B\,. \tag{5.50}$$

From (5.48) and (5.29) we get for Engset traffic:

$$C_{n,S}(\beta) < B_{n,S}(\beta) < E_{n,S}(\beta)\,. \tag{5.51}$$

5.5 Evaluation of Engset's formula

If we try to calculate numerical values of Engset's formula directly from (5.28) (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 (5.45) and (5.46). Numerically it is also simple to find any of the four parameters S, β, n, E when we know three of them. Mathematically we may assume that n and eventually S are non-integral.

5.5.1 Recursion formula on n

From the general formula (4.27) recursive in n we get using $\lambda_x = (S-x)\,\gamma$ and $\beta = \gamma/\mu$:

$$E_{x,S}(\beta) = \frac{\frac{\gamma_{x-1}}{x\,\mu} \cdot E_{x-1,S}(\beta)}{1 + \frac{\gamma_{x-1}}{x\,\mu} \cdot E_{x-1,S}(\beta)}\,,$$

$$E_{x,S}(\beta) = \frac{(S-x+1)\beta \cdot E_{x-1,S}(\beta)}{x + (S-x+1)\beta \cdot E_{x-1,S}(\beta)}\,, \quad E_{0,S}(\beta) = 1\,. \tag{5.52}$$

Introducing the reciprocal time congestion $I_{n,S}(\beta) = 1/E_{n,S}(\beta)$, we find the recursion formula:

$$I_{x,S}(\beta) = 1 + \frac{x}{(S-x+1)\,\beta}\, I_{x-1,S}(\beta)\,, \quad I_{0,S}(\beta) = 1\,. \tag{5.53}$$

The number of iterations is n. Both (5.52) and (5.53) are analytically exact, numerically stable and accurate recursions for increasing values of x. However, for decreasing values of x the numerical errors accumulate and the recursions are not reliable.

5.5.2 Recursion formula on S

Let us denote the normalized state probabilities of a system with n channels and $S-1$ sources by $p_{n,S-1}(i)$. We get the state probabilities of a system with n channels and S sources by convolving these state probabilities with the state probabilities of a single source which are given by $\{p_{1,1}(0) = 1-a,\ p_{1,1}(1) = a\}$. We then get states from zero to $n+1$, truncate the state space at n, and normalize the state probabilities (cf. Example 5.2.1) (assuming $p(x) = 0$ when $x < 0$):

$$q_{n,S}(i) = (1-a) \cdot p_{n,S-1}(i) + a \cdot p_{n,S-1}(i-1)\,, \quad i = 0, 1, \ldots, n\,. \tag{5.54}$$

The obtained state probabilities $q_{n,S}(i)$ are not normalized, because we truncate at state $[\,n\,]$ and exclude the last term for state $[\,n+1\,]$: $q_{n,S}(n+1) = a \cdot p_{n,S-1}(n)$. The normalized state probabilities $p_{n,S}(i)$ for a system with S sources and n channels are thus obtained from the normalized state probabilities $p_{n,S-1}(i)$ for a system with $S-1$ sources by:

$$p_{n,S}(i) = \frac{q_{n,S}(i)}{1 - a \cdot p_{n,S-1}(n)}\,, \quad i = 0, 1, \ldots, n\,. \tag{5.55}$$

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 inserting (5.54) in (5.55):

$$\begin{aligned} E_{n,S}(\beta) &= p_{n,S}(n) \\ &= \frac{(1-a)\cdot p_{n,S-1}(n) + a\cdot p_{n,S-1}(n-1)}{1 - a\cdot p_{n,S-1}(n)} \\ &= \frac{(1-a)\cdot E_{n,S-1}(\beta) + a\cdot \frac{n\,\mu}{(S-n)\,\gamma}\,E_{n,S-1}(\beta)}{1 - a\cdot E_{n,S-1}(\beta)}\,, \end{aligned}$$

where we have used the balance equation between state $[n-1, S-1]$ and state $[n, S-1]$. Replacing a by using (5.10) 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:

$$E_{n,S}(\beta) = \frac{S}{S-n}\cdot\frac{E_{n,S-1}(\beta)}{1+\beta\,\{1-E_{n,S-1}(\beta)\}}\,,\quad S>n\,,\quad E_{n,n}(\beta)=a^n\,. \tag{5.56}$$

The initial value is obtained from (5.15). Using the reciprocal blocking probability $I = 1/E$ we get:

$$I_{n,S}(\beta) = \frac{S-n}{S\,(1-a)}\cdot\{I_{n,S-1}(\beta)-a\}\,,\quad S>n\,,\quad I_{n,n}(\beta)=a^{-n}\,. \tag{5.57}$$

For increasing S the number of iterations is $S-n$. However, numerical errors accumulate due to the multiplication with $(S/(S-n)$ which is greater than one, and the applicability is limited. Therefore, it is recommended to use the recursion (5.59) 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.

5.5.3 Recursion formula on both n and S

If we insert (5.52) into (5.56), respectively (5.53) into (5.57), we find:

$$E_{n,S}(\beta) = \frac{S\,a\cdot E_{n-1,S-1}(\beta)}{n+(S-n)a\cdot E_{n-1,S-1}(\beta)}\,,\quad E_{0,S-n}(\beta)=1\,, \tag{5.58}$$

$$I_{n,S}(\beta) = \frac{n}{S\,a}\cdot I_{n-1,S-1}(\beta)+\frac{S-n}{S}\,,\qquad I_{0,S-n}(\beta)=1\,, \tag{5.59}$$

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 [63]).

From the above we have the following conclusions for recursion formulæ for the Engset formula. For increasing values of the parameter, recursion formulæ (5.52) & (5.53) are very accurate, and formulæ (5.58) & (5.59) are almost as good. Recursion formulæ (5.56) &(5.57) 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.

Example 5.5.1: 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:

$$\begin{aligned}
\beta &= \frac{\gamma}{\mu} = \frac{1}{3} && \text{erlang} \quad \text{(offered traffic per idle source)},\\
a &= \frac{\beta}{1+\beta} = \frac{1}{4} && \text{erlang} \quad \text{(offered traffic per source)},\\
A &= S \cdot a = 1 && \text{erlang} \quad \text{(offered traffic)},\\
Z &= 1 - \frac{A}{S} = \frac{3}{4} && \text{(peakedness)}.
\end{aligned}$$

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:

$$\begin{aligned}
\text{Time congestion:} \quad & E_{3,4}\left(\frac{1}{3}\right) = p(3) = 0.0471\,,\\
\text{Traffic congestion:} \quad & C_{3,4}\left(\frac{1}{3}\right) = \frac{A-Y}{A} = \frac{1-0.9882}{1} = 0.0118\,,\\
\text{Call congestion:} \quad & B_{3,4}\left(\frac{1}{3}\right) = \{\gamma(3)\cdot p(3)\} \Big/ \left\{\sum_{i=0}^{3} \gamma(i)\cdot p(i)\right\} = \frac{0.0157}{1.0039} = 0.0156\ .
\end{aligned}$$

We notice that $E > B > C$, which is a general result for the Engset case (5.51) & (Fig. 5.7). By

applying the recursion formula (5.52) we, of course, get the same results:

$$\begin{aligned}
E_{0,4}\left(\frac{1}{3}\right) &= 1\,, \\
E_{1,4}\left(\frac{1}{3}\right) &= \frac{(4-1+1)\cdot\frac{1}{3}\cdot 1}{1+(4-1+1)\cdot\frac{1}{3}\cdot 1} = \frac{4}{7}\,, \\
E_{2,4}\left(\frac{1}{3}\right) &= \frac{(4-2+1)\cdot\frac{1}{3}\cdot\frac{4}{7}}{2+(4-2+1)\cdot\frac{1}{3}\cdot\frac{4}{7}} = \frac{2}{9}\,, \\
E_{3,4}\left(\frac{1}{3}\right) &= \frac{(4-3+1)\cdot\frac{1}{3}\cdot\frac{2}{9}}{3+(4-3+1)\cdot\frac{1}{3}\cdot\frac{2}{9}} = \frac{4}{85} = 0.0471\,, \quad \text{q.e.d.}
\end{aligned}$$

□

Example 5.5.2: 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. 5.7 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 $S = A/(1-Z)$ (5.25). 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. 5.6 & 5.7 and Example 5.7.2). □

5.6 Pascal Distribution

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 (Wallström, 1964 [119]). The arrival intensity in state i is given by:

$$\lambda_i = \gamma \cdot (S + i), \quad 0 \le i \le n\,, \tag{5.60}$$

where γ and S are positive constants. The holding times are still assumed to be exponentially distributed with intensity μ. In this section we assume the number of channels is infinite. We

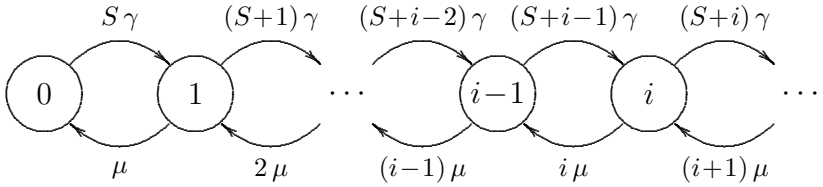


Figure 5.5: *State transition diagram for Negative Binomial case with infinite capacity.*

set up a state transition diagram (Fig. 5.6 with n infinite) and get the following cut equations:

$$\left.\begin{aligned} S\,\gamma \cdot p(0) &= \mu \cdot p(1)\,, \\ (S+1)\,\gamma \cdot p(1) &= 2\mu \cdot p(2)\,, \\ \dots &\quad \dots \\ (S+i-1)\,\gamma \cdot p(i-1) &= i\,\mu \cdot p(i)\,, \\ (S+i)\,\gamma \cdot p(i) &= (i+1)\mu \cdot p(i+1)\,, \\ \dots &\quad \dots \end{aligned}\right\} \tag{5.61}$$

To obtain statistical equilibrium it is obvious that for infinite number of channels we must require that $\gamma < \mu$ so that the arrival rate becomes smaller than the service rate from some state. All state probabilities can be expressed by $p(0)$. Assuming

$$\beta = \gamma/\mu < 1$$

and using:

$$\binom{-S}{i} = (-1)^i \cdot \binom{S+i-1}{i} = \frac{(-S)(-S-1)\dots(-S-i+1)}{i\,!} \tag{5.62}$$

we get:

$$\begin{aligned} p(1) &= \frac{S\,\gamma}{\mu} \cdot p(0) &&= p(0) \cdot \binom{-S}{1} \cdot (-\beta)^1\,, \\ p(2) &= \frac{(S+1)\,\gamma}{2\,\mu} \cdot p(1) &&= p(0) \cdot \binom{-S}{2} \cdot (-\beta)^2\,, \\ \dots &\quad \dots &&\quad \dots \\ p(i) &= \frac{(S+i-1)\,\gamma}{i\,\mu} \cdot p(i-1) &&= p(0) \cdot \binom{-S}{i} \cdot (-\beta)^i\,, \\ p(i+1) &= \frac{(S-i)\,\gamma}{(i+1)\,\mu} \cdot p(i) &&= p(0) \cdot \binom{-S}{i+1} \cdot (-\beta)^{i+1}\,, \\ \dots &\quad \dots &&\quad \dots \end{aligned}$$

The total sum of all probabilities must be equal to one:

$$\begin{aligned} 1 &= p(0)\left\{\cdot\binom{-S}{0}\cdot(-\beta)^0+\binom{-S}{1}\cdot(-\beta)^1+\binom{-S}{2}\cdot(-\beta)^2+\ldots\right\} \\ &= p(0)\cdot\{-\beta+1\}^{-S}\,, \end{aligned} \tag{5.63}$$

where we have used the generalized Newton's Binomial expansion:

$$(x+y)^r=\sum_{i=0}^{\infty}\binom{r}{i}x^i\,y^{r-i}\,, \tag{5.64}$$

which by using the definition (5.62) is valid also for complex numbers, in particular real numbers (need not be positive or integer). Thus we find the steady state probabilities:

$$p(i)=\binom{-S}{i}\cdot(-\beta)^i\,(1-\beta)^S\,,\quad 0\le i<\infty\,,\quad \beta<1\,. \tag{5.65}$$

By using (5.62) we get:

$$p(i)=\binom{S+i-1}{i}\cdot(-\beta)^i\,(1-\beta)^S\,,\quad 0\le i<\infty\,,\quad \beta<1\,, \tag{5.66}$$

which is the Pascal distribution (Tab. 3.1). The carried traffic is equal to the offered traffic as the capacity is unlimited, and it may be shown it has the following mean value and peakedness:

$$\begin{aligned} A &= S\cdot\frac{\beta}{1-\beta}\,, \\ Z &= \frac{1}{1-\beta}\,. \end{aligned}$$

These formulæ are similar to (5.22) and (5.23). The traffic characteristics of this model may be obtained by an appropriate substitution of the parameters of the Binomial distribution as explained in the following section.

5.7 Truncated Pascal distribution

We consider the same traffic process as in Sec. 5.6, but now we restrict the number of servers to a limited number n. The restriction $\gamma<\mu$ is no more necessary as we always will obtain statistical equilibrium with a finite number of states. The state transition diagram is shown in Fig. 5.6, and state probabilities are obtained by truncation of (5.65):

$$p(i)=\frac{\binom{-S}{i}(-\beta)^i}{\sum_{j=0}^{n}\binom{-S}{j}(-\beta)^j}\,,\qquad 0\le i\le n\,. \tag{5.67}$$

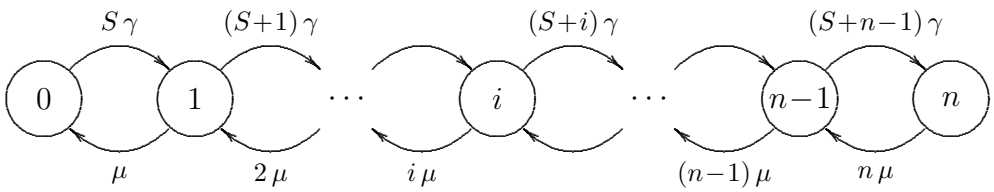


Figure 5.6: *State transition diagram for the Pascal (truncated Negative Binomial) case.*

This is the truncated Pascal distribution. Formally it can be obtained from the Engset case by the the following substitutions:

$$S \quad \text{is replaced by} \quad -S\,, \tag{5.68}$$

$$\gamma \quad \text{is replaced by} \quad -\gamma\,. \tag{5.69}$$

By these substitutions all formulæ of the Bernoulli/Engset cases are valid for the truncated Pascal distribution, and the same computer programs can be use for numerical evaluation.

It can be shown that the state probabilities (5.67) are valid for arbitrary holding time distribution (Iversen, 1980 [44]) like state probabilities for Erlang and Engset loss systems (insensitivity). 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 random intensity distributed as a gamma-distribution. Inter-arrival times are *Pareto distributed*, which is a heavy-tailed distribution. The model is used for modeling overflow traffic which has a peakedness greater than one. For the Pascal case we get (cf. (5.51)):

$$C_{n,S}(\beta) > B_{n,S}(\beta) > E_{n,S}(\beta)\,. \tag{5.70}$$

Example 5.7.1: Pascal loss system
We consider a Pascal loss system with $n = 4$ channels and $S = 2$ sources. The arrival rate is $\gamma = 1/3$ 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 let $S = -2$ (5.68) and $\gamma = -1/3$ (5.69):

$$\begin{aligned}
\beta &= \frac{\gamma}{\mu} = -\frac{1}{3}\,,\\
a &= \frac{\beta}{1+\beta} = -\frac{1}{2}\,,\\
A &= S \cdot a = -2 \cdot \left\{-\frac{1}{2}\right\} = 1 \quad \text{erlang},\\
Z &= 1 - a = \frac{3}{2}\,.
\end{aligned}$$

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:

$$\text{Time congestion:} \quad E_{4,-2}\left(-\frac{1}{3}\right) = p(4) = 0.0279\,.$$

$$\text{Traffic congestion:} \quad C_{4,-2}\left(-\frac{1}{3}\right) = \frac{A-Y}{A} = \frac{1-0.9162}{1} = 0.0838\,.$$

$$\text{Call congestion:} \quad B_{4,-2}\left(-\frac{1}{3}\right) = \frac{\gamma(4)\cdot p(4)}{\sum_{i=0}^{4}\gamma(i)\cdot p(i)} = \frac{0.0559}{0.9721} = 0.0575\,.$$

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 (5.52), we of course get the same results:

$$E_{0,-2}\left(-\frac{1}{3}\right) = 1.0000\,,$$

$$E_{1,-2}\left(-\frac{1}{3}\right) = \frac{\frac{2}{3}\cdot 1}{1+\frac{2}{3}\cdot 1} = \frac{2}{5}\,,$$

$$E_{2,-2}\left(-\frac{1}{3}\right) = \frac{\frac{3}{3}\cdot\frac{2}{5}}{2+\frac{3}{3}\cdot\frac{2}{5}} = \frac{1}{6}\,,$$

$$E_{3,-2}\left(-\frac{1}{3}\right) = \frac{\frac{4}{3}\cdot\frac{1}{6}}{3+\frac{4}{3}\cdot\frac{1}{6}} = \frac{2}{29}\,,$$

$$I_{4,-2}\left(-\frac{1}{3}\right) = \frac{\frac{5}{3}\cdot\frac{2}{29}}{4+\frac{5}{3}\cdot\frac{2}{29}} = \frac{5}{179} = 0.0279 \quad \text{q.e.d.}$$

□

Example 5.7.2: Peakedness: numerical example

In Fig. 5.7 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. □

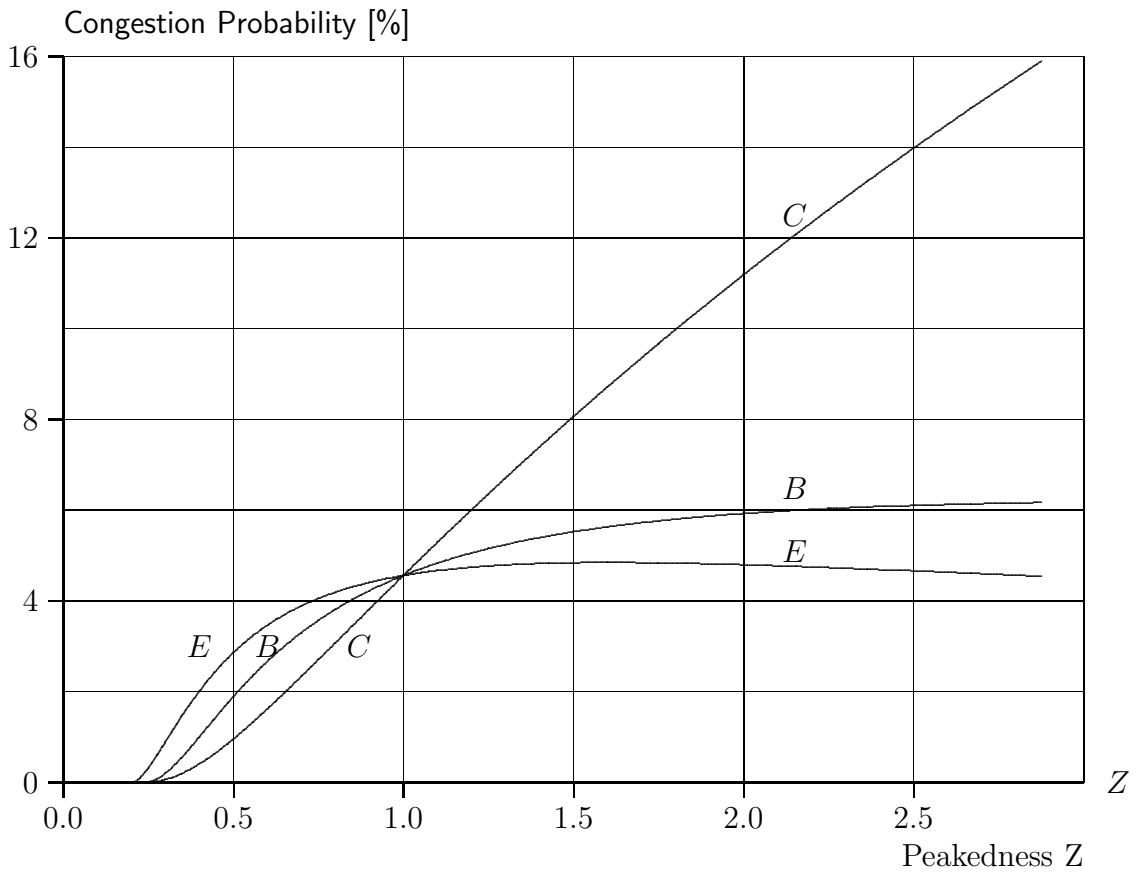


Figure 5.7: *Time congestion E, Call congestion B and Traffic congestion C as a function of peakedness Z for BPP–traffic i a system with $n = 20$ trunks and an offered traffic $A = 15$ erlang. More comments are given in Example 5.5.2 and Example 5.7.2. For applications the traffic congestion C is the most important, as it is almost a linear function of the peakedness.*

5.8 Batched Poisson arrival process

We consider an arrival process where events occur according to a Poisson process with rate λ. At each event a batch of calls (packets, jobs) arrive simultaneously. The distribution of the batch size is a discrete distribution $b(i)\,, (i = 1, 2, \ldots)$. The batch size is at least one. In the classical Erlang loss system the batch size is always one. We choose the simplest case

where the batch size distribution is a geometric distribution (Tab. 3.1, p. 92):

$$b(i) = p\,(1-p)^{i-1}\,, \quad i = 1, 2, \ldots\,. \tag{5.71}$$

$$m_1 = \frac{1}{p}\,, \tag{5.72}$$

$$\sigma^2 = \frac{1-p}{p^2}\,, \tag{5.73}$$

$$Z_{geo} = \frac{1-p}{p}\,. \tag{5.74}$$

The complementary distribution function is given by:

$$b(\geq i) = \sum_{j=i}^{\infty} p(j) = \frac{p\,(1-p)^{i-1}}{1-(1-p)} = (1-p)^{i-1}\,. \tag{5.75}$$

By the splitting theorem for the Poisson process the arrival process for batches of size i is a Poisson process with rate $\lambda \cdot b(i)$. If we assume service times are exponentially distributed with rate μ, and that each member of the batch is served independently, then the state probabilities of the number of busy channels in a system with infinite capacity has mean value and peakedness (Panken & van Doorn, 1993 [98]) as follows:

$$A = \frac{\lambda}{\mu} \cdot \frac{1}{p}\,, \tag{5.76}$$

$$Z = \frac{1}{p}\,. \tag{5.77}$$

The offered traffic A is defined as the average number of batches per mean service time multiplied by the average batch size. The peakedness is greater than one, and we may model bursty traffic by this batch Poisson model.

5.8.1 Infinite capacity

If we assume balance in a cut between state $[x-1]$ and state $[x]$ $(x = 1, 2, \ldots)$ we find:

$$x\,\mu \cdot p(x) = \sum_{i=1}^{x} p(x-i) \cdot \lambda \cdot b(\geq i) \tag{5.78}$$

$$= p(0)\,\lambda\,b(\geq x) + p(1)\,\lambda\,b(\geq x-1) + \ldots + p(x-2)\,\lambda\,b(\geq 2) + p(x-1)\,\lambda\,b(\geq 1)$$

For a cut between states state $[x-1]$ and $[x-2]$ we have in similar way:

$$\begin{aligned}(x-1)\,\mu\cdot p(x-1) &= \sum_{i=1}^{x-1} p(x-1-i)\cdot\lambda\, b(\geq i) \qquad (5.79)\\ &= p(0)\,\lambda\, b(\geq x-1)+p(1)\,\lambda\, b(\geq x-2)+\ldots+p(x-2)\,\lambda\, b(\geq 1)\,.\end{aligned}$$

As we from (3.49) have $b(\geq i+1)=(1-p)\cdot b(\geq i)$, then by multiplying (5.79) by $(1-p)$ the right hand side becomes identical with the right hand side of (5.78) except for the last term in (5.78). Observing that $b(\geq 1)=1$, we get:

$$\begin{aligned}x\,\mu\cdot p(x)-p(x-1)\,\lambda &= (1-p)\cdot(x-1)\,\mu\cdot p(x-1)\,, \qquad (5.80)\\ p(x) &= \left\{\frac{(1-p)(x-1)}{x}+\frac{\lambda}{x\,\mu}\right\}\cdot p(x-1) \qquad (5.81)\end{aligned}$$

We thus only need the previous state probability to calculate the next state probability. We may start by letting $p(0)=1$, calculate the states recursively, and normalize the state probabilities in each step of the recursion. If we rewrite (5.80) we get:

$$x\,\mu\cdot p(x)=\{(1-p)(x-1)\,\mu+\lambda\}\cdot p(x-1)\,.$$

For a Pascal traffic process we have:

$$x\,\mu\cdot p(x)=(S+x-1)\,\gamma\cdot p(x-1)\,.$$

Equalizing the right hand sides we get for the factors to $(x-1)$:

$$\begin{aligned}(1-p)\,\mu &= \gamma\,,\\ \frac{\gamma}{\mu}=\beta &= (1-p)=\frac{Z-1}{Z}\,, \qquad (5.82)\end{aligned}$$

where we have used (5.77). For the constant factors we get, exploiting (5.76) and (5.77):

$$\begin{aligned}\lambda &= S\,\gamma\,,\\ S &= \frac{\lambda}{\gamma}=\frac{A}{\beta}\cdot p=\frac{A}{Z-1}\,, \qquad (5.83)\end{aligned}$$

which is in agreement with the Pascal case ($Z>1$). So if we have a Batch geometric arrival process with mean value $A=\lambda/(\mu\cdot p)$ (5.76) and peakedness $Z=1/p$ (5.77), then we get an equivalent Pascal stream by choosing β as (5.82) and S as (5.83). Thus the Batch Poisson Process is identical with a Pascal traffic stream. This was first observed by (Kaufman & Rege [68]).

5.8.2 Finite capacity

If we have a finite number of channels and the batch size is bigger than the idle capacity, then we may either accept as many calls as possible and block the remaining calls (partial-blocking) or we may block the total batch (batch-blocking).

For partial-blocking we get the same relative state probabilities as above. This is similar to the classical loss systems where we may truncate the state space and re-normalize the state probabilities.

For batch-blocking the balance equations become:

$$x\,\mu \cdot p(x) = \sum_{i=0}^{x-1} p(i) \cdot \lambda\, b(u \mid u \leq n-i)\,, \qquad 0 < x \leq n\,, \tag{5.84}$$

where the batch size $b(u)$ to be accepted in state i now have an upper limit $n-i$. By using (5.75) we get the balance equation

$$x\,\mu \cdot p(x) = \sum_{i=0}^{x-1} p(i) \cdot \lambda\,\{1-(1-p)^{n+i}\}\,. \tag{5.85}$$

5.8.3 Performance measures

From the state probabilities we find the time, call, and traffic congestion in the usual way. The batch Poisson arrival process has the PASTA property, and therefore the time, call, and traffic congestion are equal. The traffic congestion is obtained from the state probabilities in the usual way:

$$Y = \sum_{i=0}^{n} i \cdot p(i)\,, \tag{5.86}$$

$$C = \frac{A-Y}{A}\,, \tag{5.87}$$

where the offered traffic is given by (5.76).

(The following will be elaborated further)
For partial-blocking (pb) in the Batch Poisson process we have $E_{pb} = B_{pb} = C_{pb}$, whereas the equivalent Pascal model get the same traffic congestion $C_{pas} = C_{pb}$, but smaller values of call congestion B_{pas} and time congestion E_{pas}.

For batch-blocking (bb) and a single traffic stream time congestion becomes:

$$E_{bb} = \frac{E_{bb}/p}{1+E_{bb}(1/p-1)} = \frac{E_{bb}\cdot Z}{1+E_{bb}\cdot(Z-1)} \approx E_{bb}\cdot Z\,. \tag{5.88}$$

This is close to the traffic congestion for the Pascal model, as the traffic congestion is approximately proportional to the time congestion times the peakedness.

Updated: 2010.03.04

Chapter 6

Overflow theory

In this chapter we consider systems with limited accessibility where traffic blocked from a primary group of channels overflows to secondary groups. In literature the term availability is often used for accessibility, but we will use the term availability in a more general meaning including reliability aspects and coverage in mobil communication. Both carried traffic and overflow traffic have properties different from pure chance traffic (PCT), and therefore we cannot use the classical traffic models for these streams. In Sec. 6.1 we describe a typical problem from telecommunication networks, where we use limited accessibility both for service protection and for equipment savings. The exact solution by state probabilities is dealt with in Se. 6.2. This approach is only possible for very small systems because of the state space explosion. Only for Erlang's ideal grading are we able obtain exact solutions for any parameter values.

For real systems we have to use approximate solutions or computer simulations. Approximations are either based on state space (Sec. 6.3 – Sec. 6.6) or time space (Sec. 6.7). In Sec. 6.3 we describe the carried traffic and the lost traffic by mean value and variance (or peakedness) of state probabilities. Then we assume that two traffic streams which have same mean and variance are equivalent, thereby looking away from moments of order higher than two. For a given mean and variance of overflow traffic we are able to find an Erlang loss system (defined by offered traffic and number of channels) which has the same mean and variance. This is exploited in the ERT-method (Sec. 6.4) which is the method most used in practice. Fredericks & Hayward's method (Sec. 6.5) is applicable for both smooth and bursty traffic, and easy to apply. It uses a simple transformation of the the parameters of Erlang's loss model, and is based on an optimal splitting of the traffic process. Other state-based methods are described in Sec. 6.6. In particular, the method based on the BPP (Binomial Poisson Pascal) modeling paradigm, using traffic congestion, is of interest.

Methods based on time space are in general more complex. State space based methods based on Erlang's loss model only allows for two parameters (mean and variance). Methods based on time space allows for any number of parameters. In Sec. 6.7 we describe the

application of interrupted Poisson processes and Cox-2 distributions. They both have three parameters. Using general Cox distributions or Markov modulated Poisson processes *MMPP* more parameters are available.

6.1 Limited accessibility

In this section we consider systems with limited (restricted) accessibility, i.e. systems where a subscriber or a traffic flow only has access to k specific channels out of 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. 6.1, 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 these channels all are busy, then the call is directed to the alternative (secondary) route via T (transit) 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. 6.1. 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 is routed via the group n_{12}, only when all n_1 primary channels are busy.

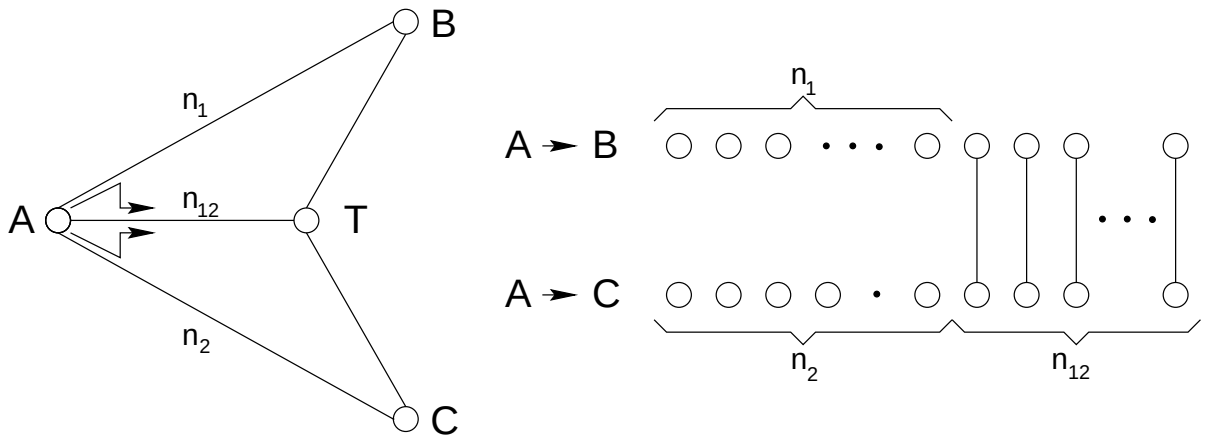


Figure 6.1: *Telecommunication network with alternate routing and the corresponding accessibility scheme, which is called an O'Dell–grading. We assume the links between the transit exchange T and the exchanges B and C are without blocking. The n_{12} channels are common for both traffic streams.*

It is typical for a hierarchical network that a certain *service protection* is inherent. 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 utilization will always be lower than for systems with full accessibility. However, the utilization

will be bigger than for two separate systems with the same total number of channels. The common channels allows for a certain blocking equalization between the 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 network planning and in guaranteeing a certain grade-of-service.

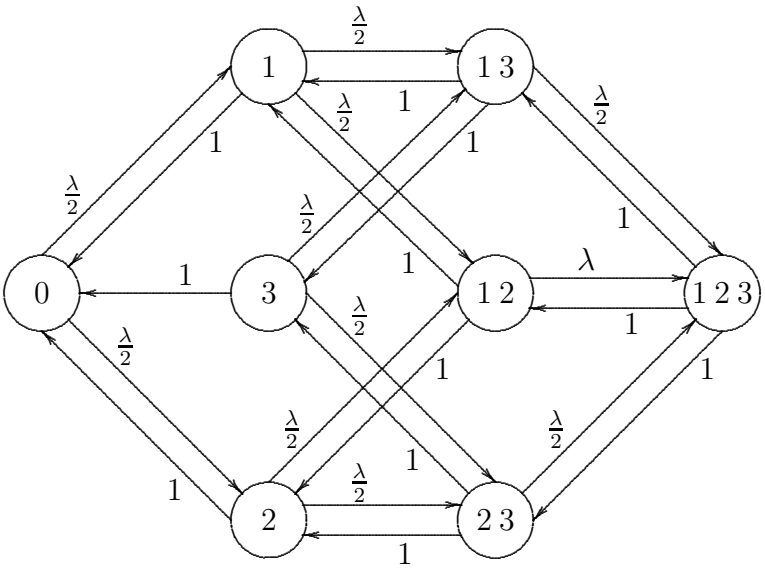


Figure 6.2: *State transition diagram for a small O'Dell grading (Fig. 6.1) with $n = 3$ channels, $n_1 = n_2 = n_{12} = 1$ (accessibility $k = 2$), ordered hunting, and offered traffic $A = \lambda$, equally distributed between the two groups (mean service time = time unit). The detailed state transition diagram has 8 states. We specify the state of each channel. The state probabilities can only be obtained by setting up all 8 balance equations (7 node equations and a normalization condition) and solve these linear equations.*

6.2 Exact calculation by state probabilities

The problem of evaluating systems with limited accessibility is due to the state space explosion. The number of states is in general so large that problems become intractable.

6.2.1 Balance equations

To have full information about the state of the system it is not sufficient to know how many channels are busy, we should also know which group a busy channel belongs to. Thus for the system in Fig. 6.1 the number of states will be $(n_1 + 1))(n_2 + 1)(n_{12} + 1)$. In worst case

we have to specify the state of each channel and thus get 2^n states. For a very small O'Dell grading with $n_1 = n_2 = n_{12} = 1$ we get 8 states as shown in Fig. 6.2.

For real systems the number of states becomes very large, and it is not convenient to find state probabilities from balance equations, or find performance measures from state probabilities. Only Erlang's ideal grading has a simple and general solution.

6.2.2 Erlang's ideal grading

Erlang's ideal grading (*EIG*) is the only system with limited accessibility, where the exact blocking probability can be calculated for any value of number of channels n, accessibility k, and offered traffic A. It is also named *Erlang's interconnection formula* (*EIF*).

The grading is optimal in the sense, that it can carry more traffic than any other gradings with *random* hunting and the same parameters. A small grading with sequential or intelligent hunting can sometimes carry a little more traffic. In this case (*EIG*) is very close to the optimal value, and the great importance of Erlang's ideal grading is, that it can be used as reference value for the maximum utilization, which can be obtained for any grading with the given parameters. Historically, there has been many misunderstandings about *EIG*, and numerically it has been difficult to evaluate the formula without computers. However, it is a model of basic theoretical interest. It can be shown that Erlangs' ideal grading is insensitive to the holding time distribution. The grading is constructed so that every channel carries the same traffic, and the number of unknown state probabilities becomes $n + 1$ as for a full accessible group.

In our terminology we consider *PCT–I* traffic offered to n identical channels. Each time a call attempt arrives, it chooses k channels at random among the n channels, and try to occupy an idle channels among these k channels. If all k channels chosen are busy, the call attempt is lost.

In order to implement this grading in an electromechanical system we divide the traffic into g inlet groups. By *r*andom *h*unting the number of inlet groups is:

$$g_{rt} = \binom{n}{k} \tag{6.1}$$

(or a whole multiple of this). This is the number of ways we can choose k channels among n channels. Each channels will appear in all possible different combinations with the other channels. By *o*rdered *h*unting the number of inlet groups becomes:

$$g_{oh} = \binom{n}{k} \cdot k! \tag{6.2}$$

(or a whole multiple of this). By ordered hunting a hunting position of a channel is important, and we ensure therefore that all possible permutations of the k hunting positions occur once.

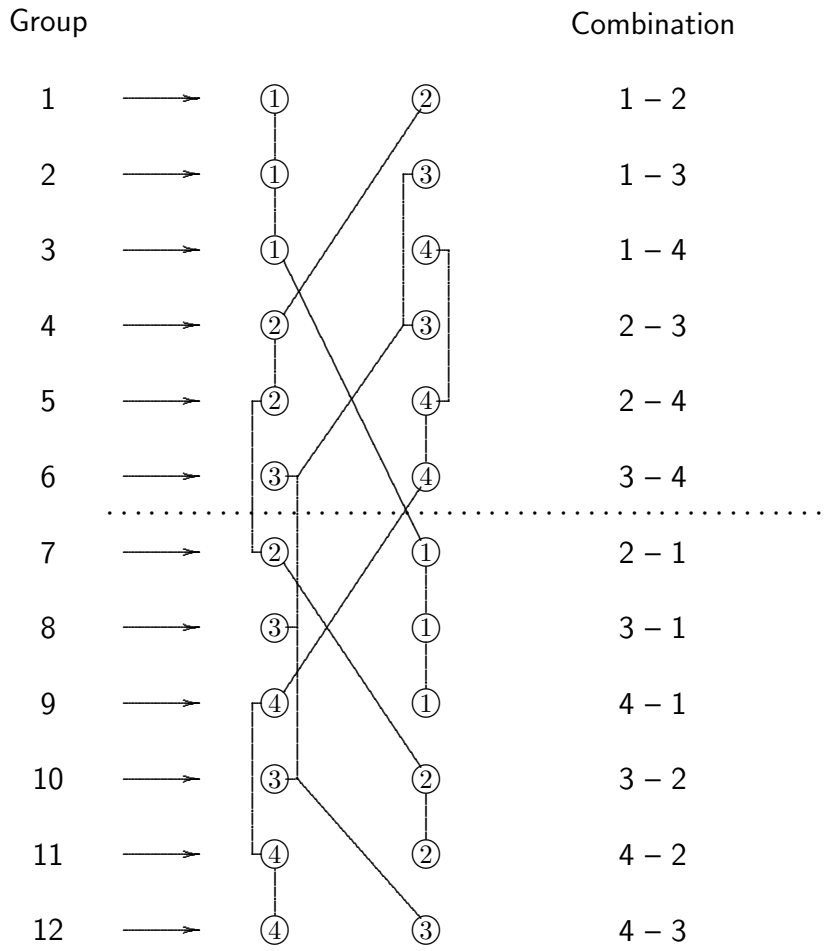


Figure 6.3: *An example of Erlang's ideal grading (Erlang's interconnection formula) (EIG) with $n = 4$ channels and $k = 2$. By random hunting the physical grading has $g_{ts} = 6$ groups (upper part), by sequential hunting the grading has $g_{os} = 12$ inlet groups. The offered traffic is distributed among the groups, so all groups receives PCT-I traffic with same intensity.*

In a digital stored-program-controlled (*SPC*) system, we do not construct these groups, but by pseudo-random numbers we may choose k channels at random, and thus construct a random group when needed. In Fig. 6.3 a realization of Erlang's ideal grading is shown.

State probabilities

Under the above mentioned assumptions we get a system where all channels are offered the same traffic, and therefore all have the same probability of being occupied at an arbitrary point of time. By exploiting this symmetry it is possible to set up the state equations under the assumption of statistical equilibrium in the such a way that we obtain the same advantages as in a full accessible group, where a state is uniquely defined by the total number of busy channels.

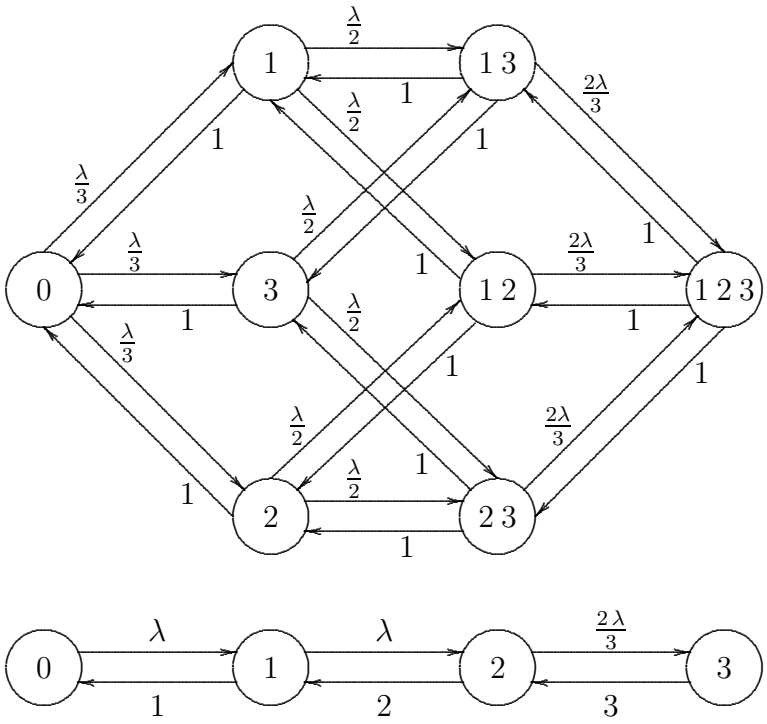


Figure 6.4: *State transition diagram for Erlang's Ideal grading with $n = 3$ channels, accessibility $k = 2$, and offered traffic $A = \lambda$ (mean service time = time unit). The detailed state transition diagram has 8 states, and there is local balance. The state is a list of individual busy channels. Due to symmetry, the detailed state transition diagram can be aggregated into a one-dimensional state transition (shown in the lower part of the figure) with the same number of states as a full accessible group.*

Fig. 6.4 shows the state transition diagram of an *EIG* with the same parameters as the O'Dell grading in Fig. 6.2. The state transition diagram is reversible and has local balance (Sec. 7.2). These are properties we consider further in connections with multi-dimensional loss systems and networks (Chap. 7).

For a call that arrives when i channels are busy, the blocking probability is equal to the probability that all k channels chosen at random are among the i busy channels. For $i < k$ no calls are lost. For $k \leq i \leq n$ the blocking probability for a call attempt becomes:

$$b_i = \frac{\binom{i}{k}}{\binom{n}{k}}, \qquad k \leq i \leq n\,. \tag{6.3}$$

For $i < k$ this is also valid, as we by definition have $\binom{i}{k} = 0$ for $i < k$. The denominator is the number of different ways we can choose k channels among n channels. The numerator is the number of times all k channels chosen are busy.

We look for the steady state probabilities $p(i)$ of the system. The cut flow balance equation

between state $i-1$ and i is:

$$\lambda\,(1-b_{i-1})\cdot p(i-1) = i\,\mu\cdot p(i)\,.$$

Thus we get:

$$\begin{aligned} p(i) &= \frac{\lambda(1-b_{i-1})}{i\,\mu}\cdot\frac{\lambda(1-b_{i-2})}{(i-1)\,\mu}\cdot\ldots\cdot\frac{\lambda(1-b_0)}{\mu}\cdot p(0) \\ &= Q_i\cdot\frac{A^{\,i}}{i\,!}\cdot p(0)\,, \end{aligned}$$

where

$$Q_i = \prod_{j=0}^{i-1}(1-b_j)\;, \qquad i=1,2,\ldots,n\,, \quad Q_0=1\,. \tag{6.4}$$

The steady state probabilities become (Brockmeyer, 1948 [12], pp. 113–119):

$$p(i) = \frac{Q_i\cdot\frac{A^{\,i}}{i\,!}}{\sum_{j=0}^{n}Q_j\cdot\frac{A^{\,j}}{j\,!}}\,, \qquad i=0,1,\ldots,n\,. \tag{6.5}$$

A call is blocked in state i with probability b_i, and the total blocking probability of Erlang's ideal grading becomes:

$$E = \sum_{i=0}^{n} b_i\cdot p(i)\,, \tag{6.6}$$

$$E = \frac{\sum_{i=0}^{n} b_i\cdot Q_i\cdot\frac{A^{\,i}}{i\,!}}{\sum_{i=0}^{n} Q_i\cdot\frac{A^{\,i}}{i\,!}}\,. \tag{6.7}$$

Due to the Poisson arrival process (*PASTA*-property) we have $E=B=C$. For $k=n$ we obtain Erlang's B-formula (4.10), since $b_i=0$ for $i<n$, and $b_n=1$.

Upper limit of channel utilization

In a trunk group (trunk = channel) there is correlation between the traffic carried by two different channels. On the average each channel carries the traffic y. The probability that a single channel is busy at a random point of time equals y. However, he probability that two channels chosen at random are busy at the same time is not y^2 due to correlation. Only when the channel group is very big, the correlation between the carried traffic on two channels becomes small. If n becomes very big and k is limited ($k \ll n$), then the congestion becomes:

$$E \approx y^k = \left\{\frac{A(1-E)}{n}\right\}^k\,, \qquad k \ll n\,. \tag{6.8}$$

as $y = A(1-E)/n$ is the carried traffic per channel (trunk).

It can be shown, that (6.8) is the theoretical lower bound for the blocking in a grading with random hunting and accessibility k. The utilization per trunk has therefore the *upper bound*:

$$\lim_{n\to\infty} y = \frac{A(1-E)}{n} = E^{1/k} < 1\,. \tag{6.9}$$

Notice that this bound is less than one and independent of n (Fig. 6.5). This formula gives a linear relation between the carried traffic $A(1-E)$ and the number of the channels n, i.e. a fixed carried traffic per channel.

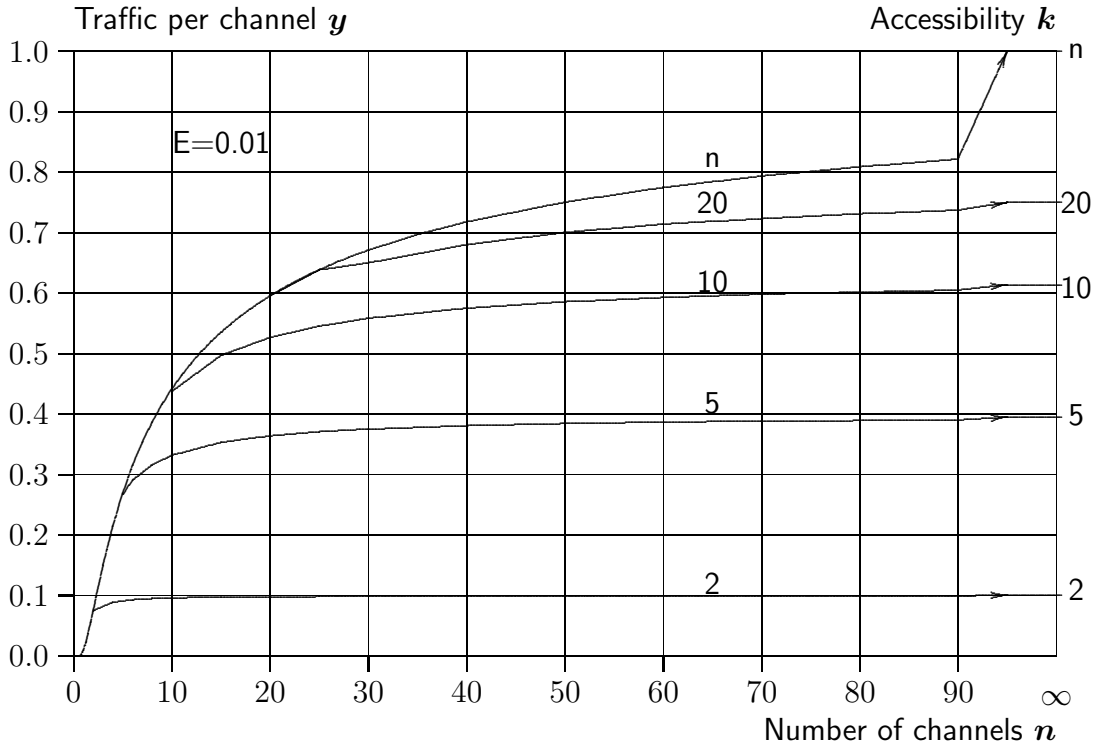


Figure 6.5: *Carried traffic y per channel as a function of the number of channels for Erlang's ideal grading with fixed blocking ($E = 0.01$). $k = n$ corresponds to a full accessible group. For fixed value of k, the carried traffic per channel y has an upper limit, which is obtained for $n \to \infty$ (6.9). This upper limit is indicated to the right.*

6.3 Overflow theory

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 alternate traffic routing, the traffic blocked from the primary group is offered to an overflow group, and this overflow traffic has properties different from *PCT* traffic as discussed in Sec. 3.7. Therefore, we cannot use the classical loss models for evaluating blocking probabilities of overflow traffic.

Example 6.3.1: 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 lost traffic:

$$A_\ell = A \cdot E_{16}(10) = 10 \cdot 0.02230 = 0.2230 \text{ [erlang]}.$$

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:

$$A_o = A \cdot E_8(A) = 10 \cdot 0.33832 = 3.3832 \text{ [erlang]}.$$

This traffic is offered to the overflow group.

Applying Erlang's B–formula for the overflow group we find the lost traffic from this group:

$$A_\ell = A_o \cdot E_8(A_o) = 3.3832 \cdot 0.01456 = 0.04927 \text{ [erlang]}.$$

The total blocking probability in this way becomes 0.4927%, 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 state space (vertical) studies we consider the state probabilities (Sec. 6.3–6.6). By time space (horizontal) studies we analyze the interval between call arrivals, i.e. the inter-arrival time distribution (Sec. 6.7).

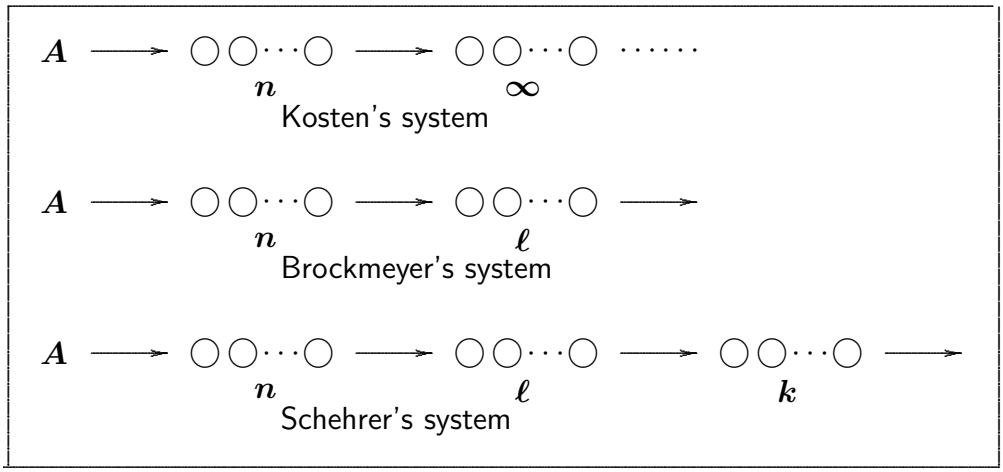


Figure 6.6: *Different overflow systems described in the literature.*

6.3.1 State probabilities of overflow systems

Let us consider a full accessible group with ordered hunting (sequential hunting with homing). The group is split into a 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. 6.6). The state of the system is described by a two-dimensional vector:

$$p(i,j), \qquad 0 \le i \le n, \quad 0 \le j \le \infty\,, \tag{6.10}$$

which is the probability that i channels are occupied in the primary group and j channels in the overflow group at a random point of time. The state transition diagram is shown in Fig. 6.7. Kosten (1937 [78]) analyzed this model and derived the marginal state probabilities:

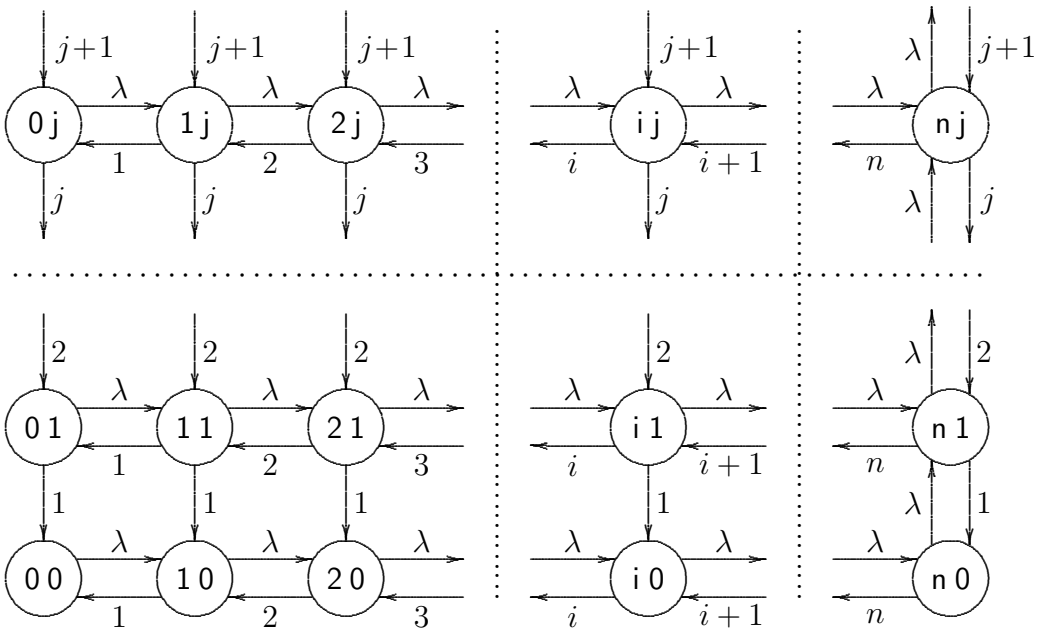


Figure 6.7: *State transition diagram for Kosten's system, which has a primary group with n channels and an unlimited overflow group. The states are denoted by $[i,j]$, where i is the number of busy channels in the primary group, and j is the number of busy channels in the overflow group. The mean holding time is chosen as time unit.*

$$p(i,\cdot) \;=\; \sum_{j=0}^{\infty} p(i,j), \qquad 0 \le i \le n\,, \tag{6.11}$$

$$p(\cdot,j) \;=\; \sum_{i=0}^{n} p(i,j), \qquad 0 \le j < \infty\,. \tag{6.12}$$

Riordan (1956 [104]) derived the moments of the marginal state probability distributions of the two groups. Mean value (carried traffic) and peakedness (= variance/mean ratio) become:

Primary group:

$$m_{1,p} = A \cdot \{1 - E_n(A)\} \,, \tag{6.13}$$

$$Z_p = \frac{v_p}{m_{1,p}} = 1 - A \cdot \{E_{n-1}(A) - E_n(A)\} \tag{6.14}$$

$$= 1 - F_{n-1}(A) \leq 1 \,,$$

where $F_{n-1}(A)$ is the improvement function of Erlang's B-formula.

Secondary group = Overflow group:

$$m_1 = A \cdot E_n(A) \,, \tag{6.15}$$

$$Z = \frac{v}{m_1} = 1 - m_1 + \frac{A}{n + 1 - A + m_1} \geq 1 \,. \tag{6.16}$$

For a fixed offered traffic, Fig. 6.8 shows that the peakedness of overflow traffic has a maximum for an increasing number of channels. Peakedness has the dimension [channels]. In practice we estimate the offered traffic by measuring the carried traffic. The peakedness is not measured, but used when dimensioning networks by the above theory.

For *PCT–I* traffic the peakedness is equal to one, and the blocking probability is calculated by using the Erlang-B formula. If the peakedness is less than one (6.14), the traffic is called *smooth*, and it will experience less congestion than *PCT–I* traffic. If the peakedness is larger than one, then the traffic is called *bursty*, and it experiences larger congestion than *PCT–I* traffic. Overflow traffic is usually bursty (6.16).

Brockmeyer (1954 [11]) derived state probabilities and moments of a system with a limited overflow group, which is called *Brockmeyer's system* (Fig. 6.6). Bech (1954 [6]) did the same by using matrix equations, and obtained more complicated and more general expressions. Brockmeyer's system is further generalized by Schehrer who also derived higher order moments for successive finite overflow groups (Fig. 6.6).

Wallström (1966 [120]) derived state probabilities and moments for overflow traffic of a generalized Kosten system, where the arrival intensity depends either upon the total number of calls in the system (Engset overflow model), or the number of calls in the primary group only (Engset loss model).

6.4 Equivalent Random Traffic Method

This equivalence method is called the *Equivalent Random Traffic Method* (*ERT*–method = *ERM*), *Wilkinson's method*, or *Wilkinson-Bretschneider's method*. It was published same

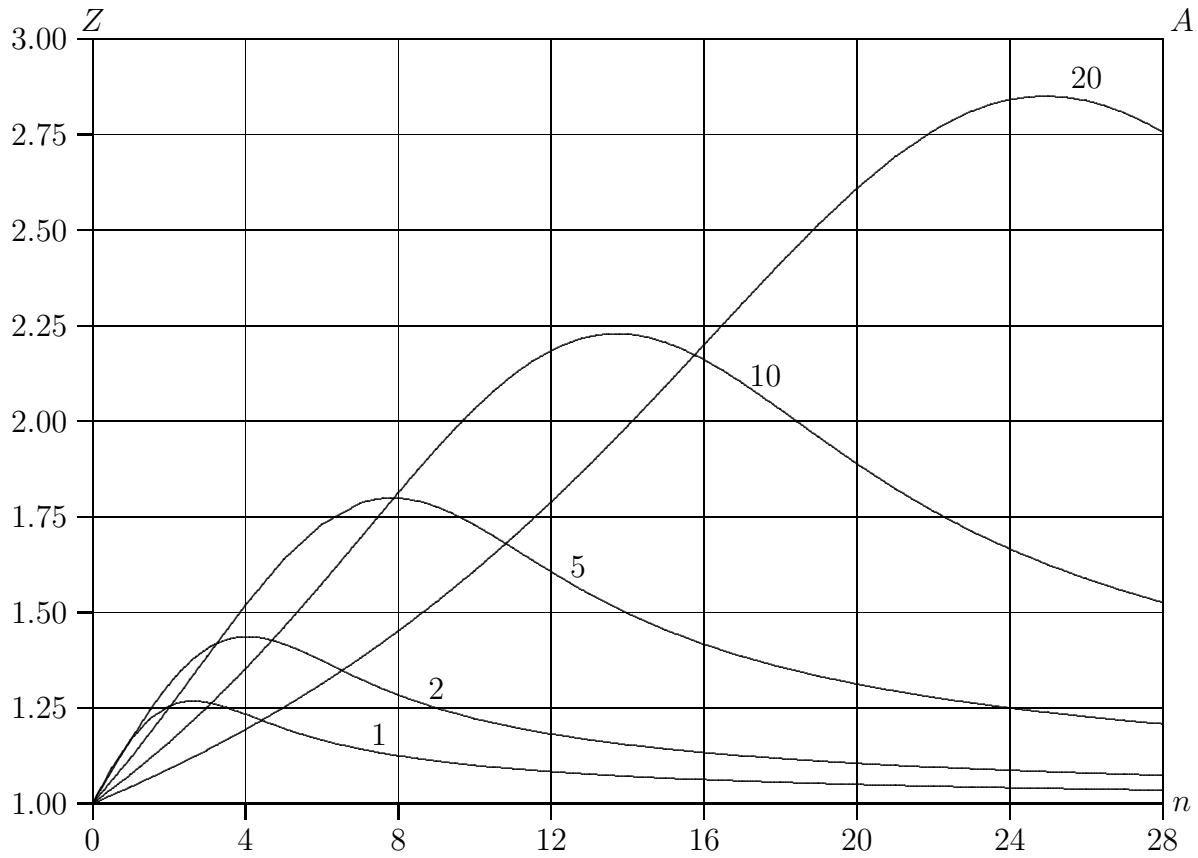


Figure 6.8: *Peakedness Z of overflow traffic as a function of number of channels for a fixed value of offered Poisson (PCT–I) traffic. Notice that Z has a maximum. When $n = 0$ all the offered traffic overflows and $Z = 1$. When n becomes very large call attempts are seldom blocked, and the blocked attempts will be mutually independent. Therefore, the process of overflowing calls converges to a Poisson process (Chap. 3).*

year in USA by Wilkinson (1956 [121]) and in Germany by Bretschneider (1956 [8]). It is a moment-matching method, approximating the first two moments of the state probabilities of an unknown traffic process with the first two moments of overflow traffic from Erlang's loss systems. It plays a key role when dimensioning telecommunication networks. (*EART* is an erroneous name for *ERT* in Cisco literature!).

6.4.1 Preliminary analysis

Let us consider a group with ℓ channels which is offered g traffic streams (Fig. 6.9). The traffic streams may be overflow 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 characterizing the i'th traffic stream by its mean value

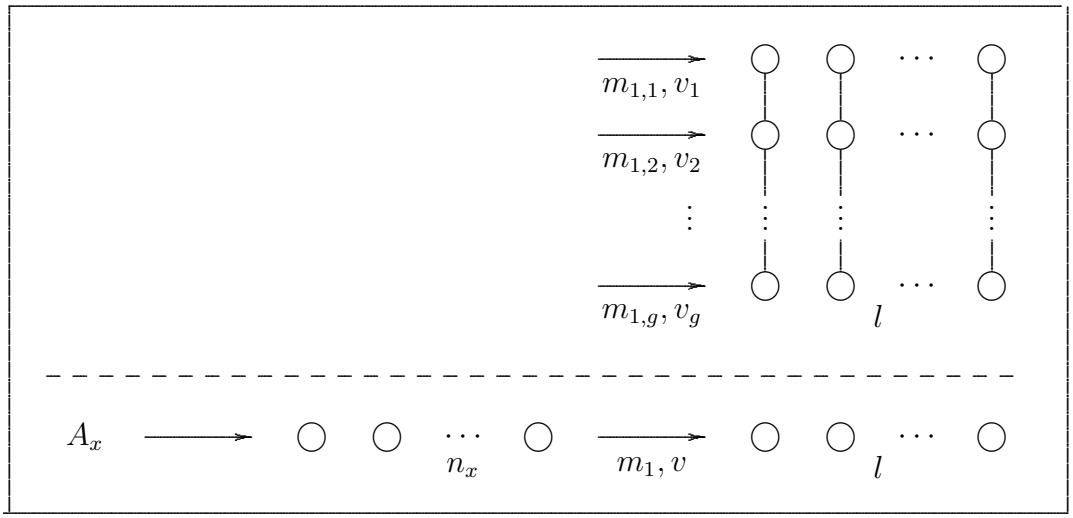


Figure 6.9: *Application of the ERT-method to a system having g independent input traffic streams offered to a common group of ℓ channels. The aggregated process of the g traffic streams is said to be equivalent to the traffic overflowing from an Erlang loss system, when the overflow traffic from the two systems have same mean value and variance. (6.17) & (6.18).*

$m_{1,i}$ and variance v_i. With this simplification *we will consider two traffic streams as being equivalent, if the state probability distributions have same mean value and variance*.

The total traffic offered to the group with ℓ channels has the mean value (2.45):

$$m_1 = \sum_{i=1}^{g} m_{1,i} \ . \tag{6.17}$$

We assume that the traffic streams are independent (non-correlated), and thus the variance of the total traffic stream becomes (2.46):

$$v = \sum_{i=1}^{g} v_i \ . \tag{6.18}$$

The total traffic is characterized by m_1 and v. So far we assume that $m_1 < v$. 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_1 and variance v. In Fig. 6.9 the upper system is replaced by the *equivalent random system* at the lower part, which is a full accessible Erlang loss system with $(n_x + \ell)$ channels and offered traffic A_x. For given values of m_1 and v we therefore solve equations (6.15) and (6.16) with respect to n and A. It can be shown there exists a unique solution which we denote by (n_x, A_x).

The traffic lost from the total system is obtained by Erlang's B-formula:

$$A_\ell = A_x \cdot E_{n_x+\ell}\,(A_x) \ . \tag{6.19}$$

As the offered traffic is m_1, the traffic congestion of the system becomes:

$$C = \frac{A_\ell}{m_1} \,. \tag{6.20}$$

Important note: the blocking probability is *not* $E_{n_x+\ell}(A_x)$. We should remember the last step (6.20), where we relate the lost traffic to the originally offered traffic, which in this case is given by m_1 (6.17). Thus we find the traffic congestion C.

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 blocking probability.

Example 6.4.1: Paradox
In Sec. 3.6 we derived the Palm-Khintchine theorem, which states that by superposition of many independent arrival processes, we *locally* get a Poisson process. This is *not* contradictory with (6.17) and (6.18), because these formulæ are valid *globally*. □

6.4.2 Numerical aspects

When applying the *ERT*–method we need to calculate (m_1, v) for given values of (A, n) and vice versa. It is easy to obtain (m_1, v) for given (A, n) by using (6.15) & (6.16). To obtain (A, n) for given (m_1, 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. 4.5). However, we can solve (6.16) with respect to n:

$$n = A \cdot \frac{m_1 + \frac{v}{m_1}}{m_1 + \frac{v}{m_1} - 1} - m_1 - 1 \,, \tag{6.21}$$

so that we can find n when A is know. Thus A is the only independent variable. We can use Newton-Raphson's iteration method to find the unknown A by introducing the function:

$$f(A) = m_1 - A \cdot E_n(A) = 0 \,.$$

For a proper starting value A_0 we iteratively improve this value until the resulting values of m_1 and v/m_1 become close enough to the known values.

Yngvé Rapp (1965 [103]) proposed a simple approximate solution for A, which can be used as initial value A_0 in the iteration:

$$A \approx v + 3 \cdot \frac{v}{m_1} \cdot \left\{ \frac{v}{m_1} - 1 \right\} \,. \tag{6.22}$$

From A obtained by iteration we then get n, using (6.21). *Rapp's approximation* is sufficient accurate for practical applications, except when A_x is very small. The peakedness $Z =$

v/m_1 has a maximum value, obtained when n is a little larger than A (Fig. 6.8). For some combinations of m_1 and v/m_1 the convergence is critical, but when using computers in a proper way we 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 assumes that v/m_1 is larger than one, and so it is only valid for bursty traffic. Individual traffic stream in Fig. 6.9 are allowed to have $v_i/m_i < 1$, provided the total aggregated traffic stream is bursty. Bretschneider ([9], 1973) 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*).

6.4.3 Individual stream blocking probabilities

The individual traffic streams (parcels) in Fig. 6.9 do not have the same mean value and variance, and therefore they will not experience the same blocking probabilities in the common overflow group with ℓ channels. From the above we calculate the mean blocking probability (6.20) for all traffic streams aggregated. Experiments show that the blocking probability is approximately proportional to the peakedness $Z = v/m_1$. We can split the total lost traffic into individual lost traffic parcels by assuming that the traffic lost by stream i is proportional to both the mean value $m_{1,i}$ and to the peakedness $Z_i = v_i/m_{1,i}$. Introducing a constant of proportionality c we get:

$$\begin{aligned} A_{\ell,i} &= A_\ell \cdot m_{1,i} \cdot Z_i \cdot c \\ &= A_\ell \cdot v_i \cdot c\,. \end{aligned}$$

We find the constant c from the total lost traffic:

$$\begin{aligned} A_\ell &= \sum_{i=1}^{g} A_{\ell,i} \\ &= \sum_{i=1}^{g} A_\ell \cdot v_i \cdot c \\ &= A_\ell \cdot v \cdot c\,. \end{aligned}$$

Thus we find $c = 1/v$. Inserting this in (6.23), the lost traffic of stream i becomes:

$$A_{\ell,i} = A_\ell \cdot \frac{v_i}{v}\,. \tag{6.23}$$

The total lost traffic is thus distributed among the individual streams according to the ratio of the individual variance of a stream to the total variance of all streams. The traffic congestion C_i for traffic stream i, which is called the *parcel blocking probability* for stream i, becomes:

$$C_i = \frac{A_{\ell,i}}{m_{1,i}} = \frac{A_\ell \cdot Z_i}{v} \,. \tag{6.24}$$

6.4.4 Individual group blocking probabilities

Furthermore, we can divide the blocking probability among the individual groups (primary, secondary, etc.). Consider the equivalent group at the bottom of Fig. 6.9 with n_x primary channels and ℓ secondary (overflow) channels. We may calculate both the blocking probability due to the n_x primary channels, and also the blocking probability due to the ℓ secondary channels. The probability that the traffic is lost from the ℓ channels is equal to the probability that the traffic is lost from the $n_x + \ell$ channels, under the condition that the traffic is offered to the ℓ channels:

$$H(l) = \frac{A \cdot E_{n_x+l}(A)}{A \cdot E_{n_x}(A)} = \frac{E_{n_x+l}(A)}{E_{n_x}(A)} \,. \tag{6.25}$$

The total loss probability can therefore be related to the two groups:

$$E_{n_x+l}(A) = E_{n_x}(A) \cdot \frac{E_{n_x+l}(A)}{E_{n_x}(A)} \,. \tag{6.26}$$

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. Formula (6.25) is called *Palm-Jacobæus* formula.

Example 6.4.2: Example 6.3.1 continued
In example 6.3.1 the blocking probability on the primary group of 8 channels is $E_8(10) = 0.3383$. The blocking of the overflow group is

$$H(8) = \frac{E_{16}(10)}{E_8(10)} = \frac{0.02230}{0.33832} = 0.06591 \,.$$

The total blocking of the system is:

$$E_{16}(10) = E_8(10) \cdot H(8) = 0.33832 \cdot 0.06591 = 0.02230 \,.$$

□

Example 6.4.3: Hierarchical cellular system (HCS)
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. We also introduce a common macro-cell covering all three areas with 8 channels. We allow

overflow from micro-cells to macro-cells, but do not rearrange (take back) the calls from macro-cells to micro-cells when a micro-cell channel becomes idle. Furthermore, we look away from hand-over traffic between micro-cells. Using (6.15) & (6.16) we find mean value and variance of the traffic overflowing from micro-cells and 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. The overflow traffic from an equivalent system with 10.78 erlang offered to 4.72 channels has the same mean and variance. Thus we end up with a system where 12.72 channels offered 10.78 erlang. Using Erlang-B formula, we find the total lost traffic 1.3049 erlang. Originally we offered 24 erlang, so the real traffic blocking probability becomes $B = 5.437\%$.
The three areas have individual blocking probabilities. Using (6.23) we estimate the traffic lost from the three traffic areas to be 0.2418 erlang, 0.5006 erlang, and 0.5625 erlang, respectively. Thus the traffic blocking probabilities become 2.02%, 6.26% and 14.06%, respectively.
A computer simulation with 100 million calls yields the individual blocking probabilities 1.77%, 5.72%, and 15.05%, respectively. The total lost traffic is 1.273 erlang, which corresponds to a blocking probability 5.30%. The accuracy of the method is thus sufficient for real applications. (The confidence intervals for the simulations are very small). □

6.5 Fredericks & Hayward's method

Fredericks (1980 [34]) has proposed an equivalence method which is simpler to use than Wilkinson-Bretschneider's *ERT*-method. The motivation for the method was first put forward by W.S. Hayward. Fredericks & Hayward's equivalence method also characterizes the traffic by mean value A and peakedness Z ($0 < Z < \infty$) ($Z = 0$ is a trivial case with constant traffic). The peakedness (4.7) is the ratio between the variance v and the mean value m_1 of the state probabilities, and the dimension is [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 which is offered the traffic A/Z. By this transformation the peakedness becomes equal to one. When $Z = 1$ the traffic is equivalent

to *PCT–I*, and we apply Erlang's B–formula for calculating the congestion:

$$E(n, A, Z) \sim E\left(\frac{n}{Z},\ \frac{A}{Z},\ 1\right) \sim E_{\frac{n}{Z}}\left(\frac{A}{Z}\right) . \tag{6.27}$$

When using this method we obtain *the traffic congestion* (Sec. 6.5.1). For fixed value of the blocking probability of the Erlang-B formula we know (Fig. 4.4) that the utilization increases, when the number of channels increases: the larger the system, the higher becomes the utilization. Fredericks & Hayward's method thus expresses that if the traffic has a peakedness Z larger than *PCT–I* traffic, then we get a lower utilization than the one obtained by using Erlang's B–formula. If peakedness $Z < 1$, then we get a higher utilization. The method can easily be applied for both peaked (bursty) and smooth traffic. By this method we avoid solving the equations (6.15) and (6.16) with respect to (A, n) for given values of (m_1, v). We only need to evaluate the Erlang-B formula. 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.

Example 6.5.1: Fredericks & Hayward's method
If we apply Fredericks & Hayward's method to example 6.4.3, 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 thus becomes $E = 1.2871/24 = 5.36\%$. This is very close to the result obtained (5.44%) by the *ERT*–method. and the result (5.30%) obtained by simulation. □

6.5.1 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, variance v/g^2, and thus peakedness Z/g because the mean value is reduced by a factor g and the variance by a factor g^2 (Example 2.4.2). If we choose the number g of identical sub-streams 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 identical sub-group, 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 each sub-stream, and we may (approximately) use Erlang's B–formula for calculating the blocking probability.

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 utilization 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. 3.6.2 we studied the splitting of the arrival process when the splitting is done in a random way (Raikov's theorem 3.2). By this splitting we do not reduce the variation of the process when the process is a Poisson process or more regular. The resulting sub-stream point processes converge to Poisson processes. In this section we have considered the splitting of the traffic load, 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. In a sub-process inter-arrival times and holding times become correlated.

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.

If we split a traffic stream into a sub-stream so that a busy channel belongs to the sub-stream with probability p, then it can be shown that the sub-stream has peakedness Z_p given by:

$$Z_p = 1 + p \cdot (Z - 1)\,, \tag{6.28}$$

where Z is the peakedness of the original stream. From this *random splitting* of the traffic process we see that the peakedness converges to one, when p becomes small. This corresponds to a Poisson process and this result is valid for any traffic process. It is similar to Raikov's theorem (3.47).

Example 6.5.2: 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 utilization as in a single system with the total capacity. This principle was first exploited in a Danish equipment (Johansen & Johansen & Rasmussen, 1991 [62]) 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 [115]), (Postigo–Boix & al. 2001 [99]). □

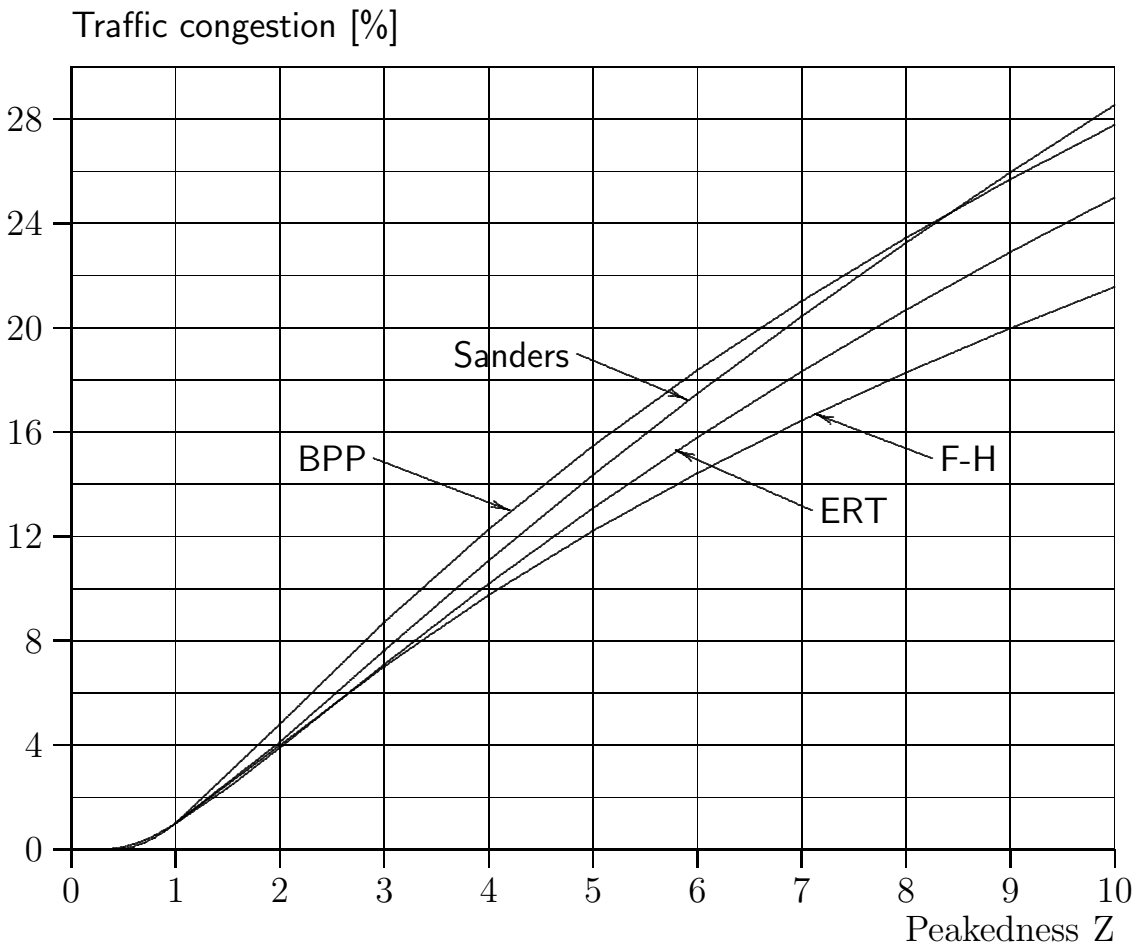


Figure 6.10: *Traffic congestion as a function of peakedness evaluated by different methods for a system with 30 channels offered 20.3373 erlang. When $Z = 1$ this corresponds to blocking probability 1 %. We notice that BPP-method is worst case method whereas Fredericks-Hayward's method yields the minimum blocking.*

6.6 Other methods based on state space

From a blocking point of the view, the mean value and variance do not necessarily characterize 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 (6.15 & 6.16). The Erlang loss system is uniquely defined by the number of channels

and the offered traffic A_x. Therefore, it is not possible to generalize the method to take account of more than two moments (mean & variance).

6.6.1 BPP traffic models

The *BPP* (Binomial–Poisson–Pascal) traffic models describe the traffic by two parameters, mean value and peakedness, and is thus natural candidates to modeling 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. 5.7 only the traffic congestion makes sense for overflow calculations. By proper application of the traffic congestion, the *BPP*–model is very applicable.

Example 6.6.1: BPP traffic model
If we apply the BPP–model to the overflow traffic in example 6.4.3 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. □

6.6.2 Sanders' method

Sanders & Haemers & Wilcke (1983 [110]) have proposed another simple and interesting equivalence method, also based on the state space. We will name it *Sanders' method*. Like Fredericks & Hayward's method, it is based on a transformation of state probabilities so that the peakedness becomes equal to one. The method transforms a non–Poisson traffic with (mean, variance) $= (m_1, v)$ into a traffic stream with peakedness one by adding a constant (zero–variance) traffic stream with mean $v - m_1$ so that the total traffic has mean equal to variance v. This constant traffic stream occupies $v - m_1$ channels permanently (with no loss) and we increase the number of channels by this amount. In this way we get a system with $n + (v - m_1)$ channels offered $m_1 + (v - m_1) = v$ erlang. The peakedness becomes one, and the blocking probability is obtained using Erlang's B-formula. We find the traffic lost from the equivalent system. To obtain the traffic congestion C of the original system, his lost traffic is divided by the originally offered traffic as the blocking probability relates to the originally offered traffic m_1.
The method is applicable for both both smooth ($m_1 > v$) and bursty traffic ($m_1 < v$), and it requires only the evaluation of the Erlang–B formula with a continuous number of channels.

Example 6.6.2: Sanders' method
If we apply Sanders' method to example 6.4.3, we increase both the number of channels and the offered traffic by $v - m_1 = 2.6691$ (channels/erlang). We 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%. □

6.6.3 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 now 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 only applicable for systems, where the primary groups all have the same number of channels.

Example 6.6.3: Group divided into primary and overflow group
If we apply Berkeley's method two example 6.3.1, then we get the exact solution. The idea of the method originates from this special case. □

Example 6.6.4: Berkeley's method
We consider example 6.4.3 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 values obtained by other methods. In general, Berkeley's method will be on the safe side. □

6.6.4 Comparison of state-based methods

In Fig. 6.10 we compare four different state-base methods. The *BPP*-method is on the safe side, whereas Frederick-Hayward's method is the most optimistic method, having lowest blocking probability. We cannot specify which method is the best one. This depends on the actual system generating the overflow traffic, which in general is a superposition of many traffic streams.

6.7 Methods based on arrival processes

The models in Chaps. 4 & 5 are all characterized 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 generalize the models by considering more general arrival processes. By using general arrival processes the insensitivity property will be 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.

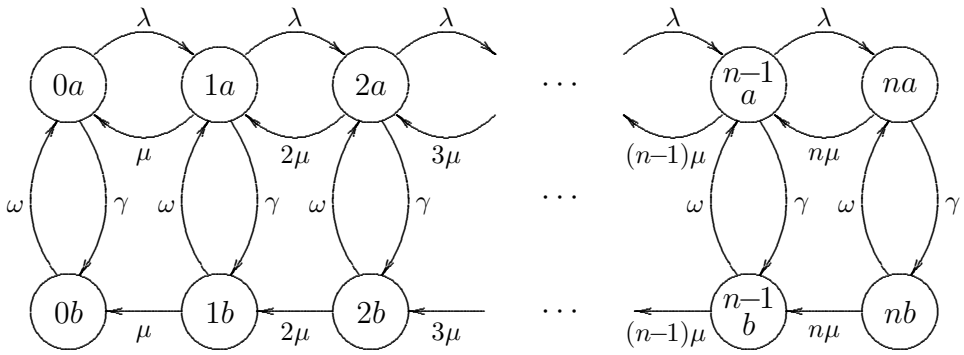


Figure 6.11: *State transition diagram for a full accessible loss system with n servers, IPP arrival process (cf. Fig. 3.9) and exponentially distributed service times (μ).*

6.7.1 Interrupted Poisson Process

In Sec. 3.7 we considered Kuczura's Interrupted Poisson Process (*IPP*) (Kuczura, 1977 [81]), which is characterized by three parameters and has been widely used for modeling overflow traffic. If we consider a full accessible group with n servers, which is offered calls arriving according to an *IPP* (cf. Fig. 3.9) with exponentially distributed service times, then we can construct a state transition diagram as shown in Fig. 6.11. 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 $p(i, j)$. Time congestion E becomes:

$$E = p(n, a) + p(n, b)\,. \tag{6.29}$$

Call congestion B becomes:

$$B = \frac{p(n, a)}{\sum_{i=0}^{n} p(i, a)} \geq E\,. \tag{6.30}$$

From the state transition diagram we have $\gamma \cdot p_{on} = \omega \cdot p_{off}$. Furthermore, $p_{on} + p_{off} = 1$. From this we get:

$$p_{on} = \sum_{i=n}^{n} p(i,a) = \frac{\omega}{\omega+\gamma},$$

$$p_{off} = \sum_{i=n}^{n} p(i,b) = \frac{\gamma}{\omega+\gamma}.$$

Traffic congestion C is defined as the proportion of the offered traffic which is lost. The offered traffic is equal to:

$$A = \frac{p_{on}}{p_{on}+p_{off}} \cdot \lambda \cdot \frac{1}{\mu} = \frac{\omega}{\omega+\gamma} \cdot \frac{\lambda}{\mu}.$$

The carried traffic is:

$$Y = \sum_{i=0}^{n} i \cdot \{p(i,a) + p(i,b)\}. \tag{6.31}$$

From this we obtain

$$C = \frac{A-Y}{A}. \tag{6.32}$$

The traffic congestion will be equal to the call congestion as the arrival process is a renewal process. But this is difficult to derive from the above. As shown in Sec. 3.7.1 the inter-arrival times are hyper-exponentially distributed with two phases (H_2). If we apply a Markov Modulated Poisson process ($MMPP$), then in principle we may get any number of parameters to model inter-arrival times.

Example 6.7.1: Calculating state probabilities for IPP models
The state probabilities of Fig. 6.11 can be obtained by solving the linear balance equations. Kuczura (1973, [80]) derived explicit expressions for the state probabilities, but they are complex and not fit for numerical evaluation of large systems. The way to calculate state probabilities in a very accurate way is to use the principles described in Sec. 4.4.1:

- let $p(n,b) = 1$,
- by using node equation for this state $[n,b]$ we obtain the value of $p(n,a)$ relative to $p(n,b)$, and normalize the two state probabilities so they add to one.
- by using node equation for state $[n,a]$ we obtain $p(n-1,a)$ relative to the previous states, and normalize the state probabilities obtained so far.
- by using node equation for state $[n-1,b]$, we obtain $p(n-1,b)$ and normalize all the obtained state probabilities.
- in this way we zig-zag down to state $[0,a]$ and obtain normalized probabilities for all states.

The relative values of for example $p(0,a)$ and $p(0,b)$ depend on the number of channels n. Thus we cannot truncate the state probabilities and re-normalize for a given number of channels, but we have to calculate all state probabilities from scratch for every number of channels. □

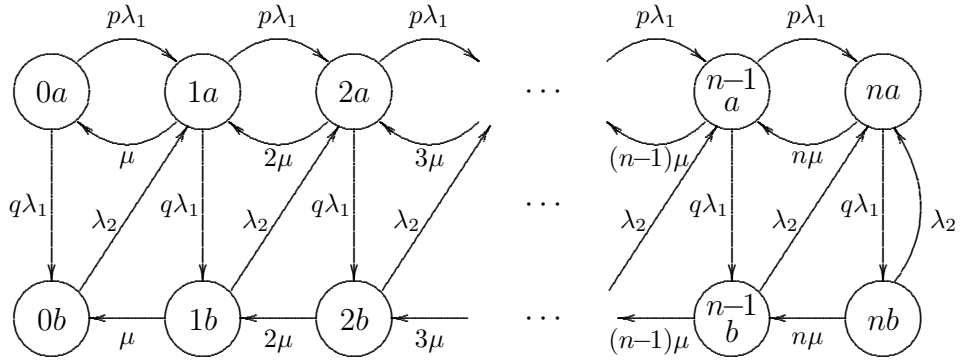


Figure 6.12: *State transition diagram for a full accessible loss system with n servers, Cox–2 arrival processes (cf. Fig. 2.13) and exponentially distributed service times* (μ).

6.7.2 Cox–2 arrival process

In Sec. 3.7 we noticed that a *Cox–2* arrival process is more general than an *IPP* (Kuczura, 1977 [81]). If we consider *Cox–2* arrival processes as shown in Fig. 2.13, then we get the state transition diagram shown in Fig. 6.12. From this we find under the assumption of statistical equilibrium the state probabilities and the following performance measures. There is no simple recursion to calculate the state probabilities.

Time congestion E:

$$E = p(na) + p(nb)\,. \tag{6.33}$$

Call congestion B:

$$B = \frac{p\,\lambda_1 \cdot p(na) + \lambda_2 \cdot p(nb)}{p\,\lambda_1 \cdot \sum_{i=0}^{n} p(ia) + \lambda_2 \cdot \sum_{i=0}^{n} p(ib)}\,. \tag{6.34}$$

Traffic congestion C:

The offered traffic is the average number of call attempts per mean service time. The mean inter-arrival time is (Fig. 2.13):

$$m_a = \frac{1}{\lambda_1} + (1-p) \cdot \frac{1}{\lambda_2} = \frac{\lambda_2 + (1-p)\lambda_1}{\lambda_1\,\lambda_2}\,.$$

The offered traffic then becomes $A = (m_a \cdot \mu)^{-1}$. The carried traffic Y is given by (6.31) applied to Fig. 6.12 and then we find the traffic congestion C by (6.32).

If we generalize 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 generalize 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 generalize the arrival process and assume exponentially distributed service times.

vbi-2010.03-16

Chapter 7

Multi-Dimensional Loss Systems

In this chapter we generalize the classical teletraffic theory to deal with service-integrated systems (e.g. *B-ISDN*). Every class of service corresponds to a traffic stream. Several traffic streams are offered to the same group of n channels.

In Sec. 7.1 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. 7.2. In Sec. 7.3 we look at more general loss models and strategies, including service-protection (maximum allocation) and multi-rate *BPP*–traffic. The models all have the so-called *product-form* property, and the numerical evaluation is very simple, using either the convolution algorithm for loss systems which aggregates traffic streams (Sec. 7.4), or state-based algorithms which aggregate the state space (Sec. 7.6).

All models considered are based on flexible channel/slot allocation, which means that if a call requests $d > 1$ channels, then these channels need not be adjacent. The models may be generalized to arbitrary circuit switched networks with direct routing, where we calculate end-to-end blocking probabilities (Chap. 8). All models considered are insensitive to the service time distribution, and thus they are very robust for applications.

7.1 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 $A_2 = \lambda_2/\mu_2$, and the total offered traffic is $A = A_1 + A_2$. In this section each connection requests one channel.

Let (x_1, x_2) denote the state of the system, i.e. x_1 is the number of channels used by stream

one and x_2 is the number of channels used by stream two. We have the following restrictions:

$$\begin{aligned} 0 \leq x_1 &\leq n\,, \\ 0 \leq x_2 &\leq n\,, \\ 0 \leq x_1 + x_2 &\leq n\,. \end{aligned} \tag{7.1}$$

The state transition diagram is shown in Fig. 7.1. Under the assumption of statistical equilibrium, the state probabilities are obtained by solving the global balance equations for each node (node equations). In total we have $(n+1)(n+2)/2$ equations. The system has a unique solution. So if we somehow find a solution, then we know that this is the correct solution. Many models can, however, be solved in a much simpler way.

As we shall see in 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 in product form:

$$\begin{aligned} p(x_1, x_2) &= Q \cdot p_1(x_1) \cdot p_2(x_2) \\ &= Q \cdot \frac{A_1^{x_1}}{x_1!} \cdot \frac{A_2^{x_2}}{x_2!}\,, \end{aligned} \tag{7.2}$$

where $p_1(x_1)$ and $p_2(x_2)$ are one-dimensional truncated Poisson distributions for traffic stream one, respectively two. Q is a normalization constant, and (x_1, x_2) must fulfil the above restrictions (7.1). As we have Poisson arrival processes, the *PASTA*-property (*P*oisson *A*rrivals *S*ee *T*ime *A*verages) is valid, and time, call, and traffic congestion are all equal to $p(x_1 + x_2 = n)$ for the two traffic streams.

By the Binomial expansion (2.38), or by convolving two Poisson distributions, we find the following aggregated state probabilities, where Q is obtained by normalization:

$$p(x_1 + x_2 = x) = Q \cdot \sum_{x_1=0}^{x} p_1(x_1) \cdot p_2(x - x_1) \tag{7.3}$$

$$= Q \cdot \sum_{x_1=0}^{x} \frac{A_1^{x_1}}{x_1!} \cdot \frac{A_2^{x-x_1}}{(x-x_1)!} \tag{7.4}$$

$$= Q \cdot \frac{1}{x!} \cdot \sum_{x_1=0}^{x} \binom{x}{x_1} A_1^{x_1} \cdot A_2^{x-x_1} \tag{7.5}$$

$$= Q \cdot \frac{1}{x!} \cdot (A_1 + A_2)^x = Q \cdot \frac{A^x}{x!}\,, \tag{7.6}$$

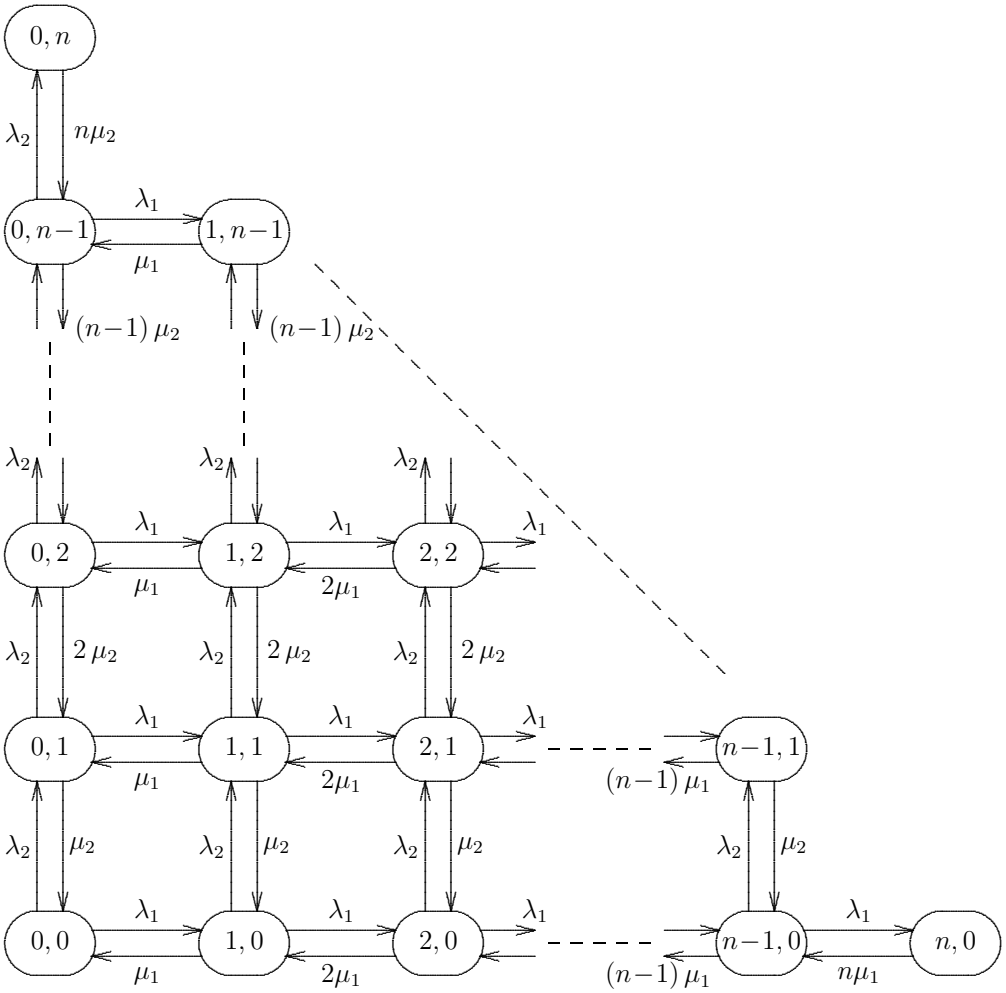


Figure 7.1: *Two-dimensional state transition diagram for a loss system with n channels which are offered two PCT–I traffic streams. This is equivalent to a state transition diagram for a loss system $M/H_2/n$, where the hyper-exponential distribution H_2 is given by (7.8).*

where $A = A_1 + A_2$, and the normalization constant is obtained by: $Q^{-1} = \sum_{i=0}^{n} \frac{A^i}{i!}$. This is the truncated Poisson distribution (4.9).

We may also interpret this model as an Erlang loss system with one Poisson arrival process and hyper-exponentially distributed holding times as follows. The total arrival process is a superposition of two Poisson processes and thus a Poisson process itself with arrival rate:

$$\lambda = \lambda_1 + \lambda_2 \,. \tag{7.7}$$

The holding time distribution is obtained by weighting the two exponential distributions according to the relative number of calls per time unit and becomes a hyper-exponential distribution (random variables in parallel, Sec. 2.3.2):

$$f(t) = \frac{\lambda_1}{\lambda_1 + \lambda_2} \cdot \mu_1 \cdot \mathrm{e}^{-\mu_1 t} + \frac{\lambda_2}{\lambda_1 + \lambda_2} \cdot \mu_2 \cdot \mathrm{e}^{-\mu_2 t} \,. \tag{7.8}$$

The mean service time is:

$$m_1 = \frac{\lambda_1}{\lambda_1+\lambda_2}\cdot\frac{1}{\mu_1}+\frac{\lambda_2}{\lambda_1+\lambda_2}\cdot\frac{1}{\mu_2}=\frac{A_1+A_2}{\lambda_1+\lambda_2}\,,$$

$$m_1 = \frac{A}{\lambda}\,, \tag{7.9}$$

which is in agreement with the definition of offered traffic (1.2).

Thus we have shown that Erlang's loss model is also valid for hyper-exponentially distributed holding times. This is a special case of the general insensitivity property of Erlang's B–formula.

We may generalize the above model to N traffic streams:

$$\begin{aligned} p(x_1,x_2,\cdots,x_N) &= Q\cdot p_1(x_1)\cdot p_2(x_2)\cdot\ldots\cdot p_N(x_N) \\ &= Q\cdot\frac{A_1^{x_1}}{x_1!}\cdot\frac{A_2^{x_2}}{x_2!}\cdot\ldots\cdot\frac{A_N^{x_N}}{x_N!}\,, \quad 0\le x_j\le n\,, \quad \sum_{j=1}^{N}x_j\le n\,, \end{aligned} \tag{7.10}$$

which is the general multi-dimensional Erlang-B formula. By the multinomial theorem (2.99) this can be reduced to:

$$\begin{aligned} p(x_1+x_2+\ldots+x_N=x) &= Q\cdot\frac{(A_1+A_2+\ldots+A_N)^x}{x!} \\ &= Q\cdot\frac{A^x}{x!}\,, \qquad \text{where} \qquad A=\sum_{j=1}^{N}A_j\,, \end{aligned}$$

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. From the cut equations of Erlang's loss system (Sec. 4.2.2) we have:

$$q(x) = \frac{1}{x}\cdot A\cdot q(x-1)=\frac{1}{x}\cdot\left(\sum_{j=1}^{N}A_j\right)\cdot q(x-1)\,, \qquad q(0)=1\,, \tag{7.11}$$

$$p(x) = \frac{q(x)}{Q(n)}\,, \quad 0\le x\le n\,, \qquad \text{where} \qquad Q(n)=\sum_{i=0}^{n}q(i)\,. \tag{7.12}$$

If we use this recursion with normalization in each step (Sec. 4.4), then we get the recursion formula for Erlang–B. For all services the time congestion is $E=p(n)$, and as the *PASTA*-property is valid, this is also equal to the call and traffic congestion. Multi-dimensional systems were first mentioned by Erlang and more thoroughly dealt with by Jensen in the *Erlangbook* (Jensen, 1948 [58]).

Example 7.1.1: Infinite server (IS) system
If the number of channels is infinite, then we get:

$$\begin{aligned} p(x_1, x_2, \ldots, x_N) &= p_1(x_1) \cdot p_2(x_2) \cdot \ldots \cdot p_N(x_N) \\ &= \left(\frac{A_1^{x_1}}{x_1!} \cdot e^{-A_1}\right) \cdot \left(\frac{A_2^{x_2}}{x_2!} \cdot e^{-A_2}\right) \cdot \ldots \cdot \left(\frac{A_N^{x_N}}{x_N!} \cdot e^{-A_N}\right) \end{aligned} \tag{7.13}$$

By using the multi-nominal expansion (2.97), the global state probabilities obtained by aggregating the detailed states probabilities become Poisson distributed (4.6):

$$\begin{aligned} p(x_1 + x_2 + \ldots + x_N = x) &= \frac{(A_1 + A_2 + \ldots + A_N)^{(x_1+x_2+\ldots+x_N)}}{(x_1 + x_2 + \ldots + x_N)!} \cdot e^{-(A_1+A_2+\ldots+A_N)} \\ &= \frac{A}{x\,!} \cdot e^{-A}\,, \end{aligned}$$

which has the mean $A = A_1 + A_2 + \ldots + A_N$. The product of individual Poisson distributions is already normalized because we don't truncate the state space. □

7.2 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 and properties of the state transition diagram. Let us consider the two-dimensional state transition diagram shown in Fig. 7.2 with state-dependent arrival and service rates. The process is reversible if there is no circulation flow in the diagram. Thus, if we consider four neighboring states, then flow in clockwise direction must equal flow in opposite direction (Kingman, 1969 [74]), (Sutton, 1980 [113]). From Fig. 7.2 we have the following average number of jumps per time unit:

Clockwise:

$$\begin{aligned} [x_1, x_2] &\rightarrow [x_1, x_2 + 1]: && p(x_1, x_2) \cdot \lambda_2(x_1, x_2) \\ [x_1, x_2 + 1] &\rightarrow [x_1 + 1, x_2 + 1]: && p(x_1, x_2{+}1) \cdot \lambda_1(x_1, x_2{+}1) \\ [x_1 + 1, x_2 + 1] &\rightarrow [x_1 + 1, x_2]: && p(x_1{+}1, x_2{+}1) \cdot \mu_2(x_1{+}1, x_2{+}1) \\ [x_1 + 1, x_2] &\rightarrow [x_1, x_2]: && p(x_1{+}1, x_2) \cdot \mu_1(x_1{+}1, x_2)\,, \end{aligned}$$

Counter clockwise:

$$\begin{aligned} [x_1, x_2] &\rightarrow [x_1 + 1, x_2]: && p(x_1, x_2) \cdot \lambda_1(x_1, x_2) \\ [x_1 + 1, x_2] &\rightarrow [x_1 + 1, x_2 + 1]: && p(x_1{+}1, x_2) \cdot \lambda_2(x_1{+}1, x_2) \\ [x_1 + 1, x_2 + 1] &\rightarrow [x_1, x_2 + 1]: && p(x_1{+}1, x_2{+}1) \cdot \mu_1(x_1{+}1, x_2{+}1) \\ [x_1, x_2 + 1] &\rightarrow [x_1, x_2]: && p(x_1, x_2{+}1) \cdot \mu_2(x_1, x_2{+}1)\,. \end{aligned}$$

We can reduce both expressions by the state probabilities and then obtain the conditions given by the following theorem.

Theorem 7.1 (Kolmogorov cycle criteria) *A necessary and sufficient condition for reversibility is that the following two flows are equal:*

Clockwise: $\lambda_2(x_1, x_2) \cdot \lambda_1(x_1, x_2+1) \cdot \mu_2(x_1+1, x_2+1) \cdot \mu_1(x_1+1, x_2)$,

Counter clockwise: $\lambda_1(x_1, x_2) \cdot \lambda_2(x_1+1, x_2) \cdot \mu_1(x_1+1, x_2+1) \cdot \mu_2(x_1, x_2+1)$.

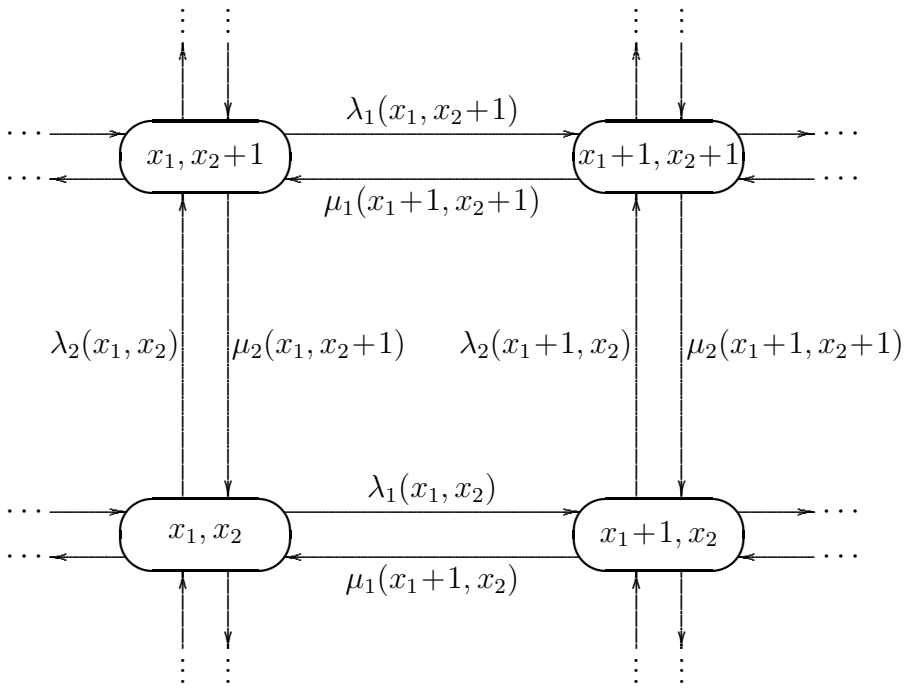


Figure 7.2: *Kolmogorov cycle criteria: a necessary and sufficient condition for reversibility of a two-dimensional Markov process is that the circulation flow among four neighbouring states in a square equals zero: Flow clockwise = flow counter-clockwise (Theorem 7.1).*

If these two expressions are equal, then there is *local balance or detailed balance*. A necessary condition for reversibility is thus that if there is a flow (an arrow) from state x_1 to state x_2, then there must also be a flow (an arrow) from state x_2 to state x_1, and the flows must be equal. It can be shown that this is also a sufficient condition. We may then apply cut equations locally between any two connected states. For example, we get from Fig. 7.2:

$$p(x_1, x_2) \cdot \lambda_1(x_1, x_2) = p(x_1 + 1, x_2) \cdot \mu_1(x_1 + 1, x_2) \,. \tag{7.14}$$

We can express any state probability $p(x_1, x_2)$ by state probability $p(0, 0)$ by choosing any path between the two states *(Kolmogorov cycle criteria)*. If we for example choose the path:

$$(0, 0), (1, 0), \ldots, (x_1, 0), (x_1, 1), \ldots, (x_1, x_2) \,,$$

then we obtain the following balance equation:

$$p(x_1, x_2) = \frac{\lambda_1(0,0)}{\mu_1(1,0)} \cdot \frac{\lambda_1(1,0)}{\mu_1(2,0)} \cdots \frac{\lambda_1(x_1-1,0)}{\mu_1(x_1,0)} \cdot \frac{\lambda_2(x_1,0)}{\mu_2(x_1,1)} \cdot \frac{\lambda_2(x_1,1)}{\mu_2(x_1,2)} \cdots \frac{\lambda_2(x_1,x_2-1)}{\mu_2(x_1,x_2)} \cdot p(0,0) \tag{7.15}$$

State probability $p(0,0)$ is obtained by normalization of the total probability mass.

The condition for reversibility will be fulfilled in many cases, for example when:

$$\lambda_1(x_1, x_2) = \lambda_1(x_1)\,, \qquad \mu_1(x_1, x_2) = x_1 \cdot \mu_1\,, \tag{7.16}$$

$$\lambda_2(x_1, x_2) = \lambda_2(x_2)\,, \qquad \mu_2(x_1, x_2) = x_2 \cdot \mu_2\,, \tag{7.17}$$

which includes Engset and Pascal traffic.

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 (Theorem 7.1). Kolmogorov cycle criteria must still be fulfilled for all possible cycles. 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 general advanced multi-service traffic model which are robust and easy to deal with. The models are all insensitive to the holding time distribution which means that the state probabilities depend only upon the mean service time,

7.3 Multi-Dimensional Loss Systems

In this section we consider generalizations of the classical teletraffic theory to cover several traffic streams (classes, services) offered to a link with a fixed bandwidth, which is expressed in channels of basic bandwidth units (BBU). Each traffic stream may have individual parameters and may be state-dependent Poisson arrival processes with multi-rate 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 generalizations one at a time and present a small case-study to illustrate the basic ideas.

7.3.1 Class limitation

In comparison with the case considered in Sec. 7.1 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 a limited number of specific channels, then

we now have access to all channels, but at any instant we may at most occupy a maximum number of channels. 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:

$$0 \leq x_j \leq n_j \leq n\,, \qquad j = 1, 2, \ldots, N\,, \tag{7.18}$$

where

$$\sum_{j=1}^{N} x_j \leq n \qquad \text{and} \qquad \sum_{j=1}^{N} n_j > n\,.$$

If the latter restriction is not fulfilled, then we get a system with separate groups, corresponding to N independent one-dimensional loss systems. Due to these restrictions the state transition diagram is truncated. This is shown for two traffic streams in Fig. 7.3.

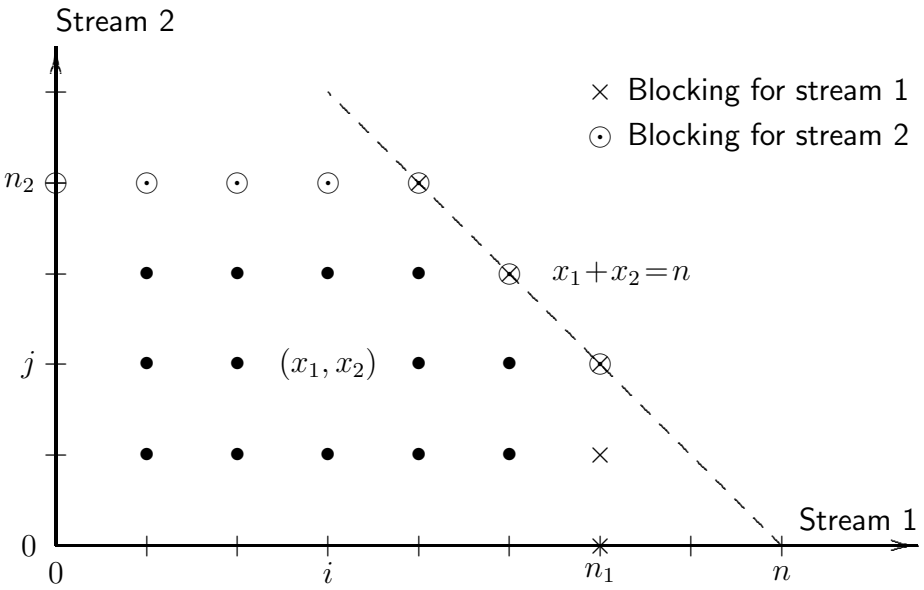


Figure 7.3: *Structure of the state transition diagram for two-dimensional traffic processes with class limitations (cf. (7.18)). When calculating the equilibrium probabilities, state* (x_1, x_2) *can be expressed by state* $(x_1, x_2 - 1)$ *and recursively by state* $(x_1, 0)$, $(x_1 - 1, 0)$, *and finally by* $(0, 0)$ *(cf. (7.15)).*

We notice that the truncated state transition diagram still is reversible, and that the values of $p(x_1, x_2)$ relative to the value $p(0, 0)$ are unchanged by the truncation. Only the normalization 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 subsets of traffic streams so that any traffic stream has a minimum (guaranteed) number of allocated channels.

7.3.2 Generalized traffic processes

We are not restricted to *PCT–I* traffic only as in Sec. 7.1. Every traffic stream may be a state-dependent Poisson arrival process with a linear state-dependent death (service) rate (cf. (7.16) and (7.17)). The system still fulfils the reversibility conditions given by Theorem 7.1. The product form is valid 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 [58]). As mentioned above, the system is insensitive to the holding time distributions with individual mean values. Every traffic stream may have its own individual holding time distribution.

7.3.3 Multi-rate traffic

In service-integrated systems the bandwidth requested depend on the type of service. We choose a *Basic Bandwidth Unit* (BBU) and split the available bandwidth into n BBUs. The BBU corresponds to a channel, a slot, a server, etc. The smaller the basic bandwidth unit is, the more accurate we may model different services, but with a finer granularity the state space increases.

Thus a voice telephone call may only require one channel (slot), whereas for example a video connection may require d channels simultaneously. Therefore, we get the capacity restrictions:

$$0 \leq x_j = i_j \cdot d_j \ \leq \ n_j \ \leq \ n \, , \quad j = 1, 2, \ldots, N \, , \tag{7.19}$$

and

$$0 \leq \sum_{j=1}^{N} i_j \cdot d_j \ \leq n \, , \tag{7.20}$$

where i_j is the actual number of type j calls (connections) and x_j is the number of channels (BBU) occupied by type j. 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. 7.5.

Offered traffic A_j is usually defined as the traffic carried when the capacity in unlimited. If we measure the carried traffic Y_j as the average number of busy channels, then the lost traffic measured in channels becomes:

$$A_\ell = \sum_{j=1}^{N} A_j \, d_j - \sum_{j=1}^{N} Y_j \, , \tag{7.21}$$

where we as usual define $A_j = \lambda_j / \mu_j$.

Example 7.3.1: Basic bandwidth units
For a 640 Mbps link we may choose BBU = 64 Kbps, corresponding to one voice channel. Then the

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
$n_1 = 6 = n$	$n_2 = 6 = n$

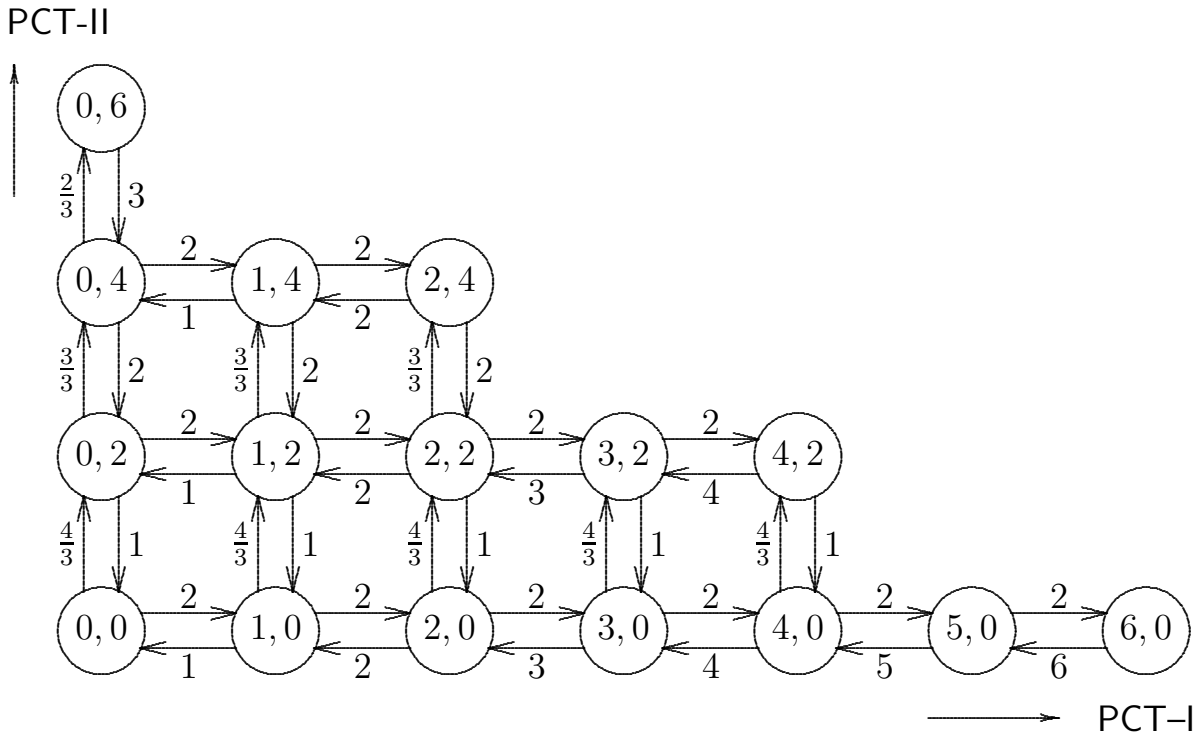
Table 7.1: *Two traffic streams: a Poisson traffic process (Example 4.5.1) and a Binomial traffic process (Example 5.5.1) are offered to the same trunk group.*

total capacity becomes $n = 10{,}000$ channels.
For a UMTS CDMA system with chip rate 3.84 Mcps, one chip is one bit from the direct sequence spread spectrum code. We can choose the BBU as a multiple of 1 cps. In practice the BBU depends on the code length. A 10–bit code allows for a granularity of 1024 channels, and the BBU becomes 3.75 Kcps (we consider gross rates).
For variable bit rate (*VBR*) services we may statistically define an effective bandwidth which is the capacity we need to reserve on a link with a given total capacity to fulfill a certain grade-of-service. □

Example 7.3.2: Rönnblom's model
The first example of a multi-rate traffic model was published by Rönnblom (1958 [109]). The paper considers a *PABX* telephone exchange with both-way channels with both external (outgoing and incoming) traffic and internal traffic. The external calls occupies only one channel per call. The internal calls occupies both an outgoing channel and an incoming channel and thus requires two channels simultaneously. It was shown by Rönnblom that this model has product form. □

Example 7.3.3: Two traffic streams
We now illustrate the above models by a small instructive case-study. The principles and procedures are the same as for the general case considered later by the convolution algorithm (Sec. 7.4.1). We consider a trunk group of 6 channels which is offered two traffic streams, specified in Tab. 7.1.
We notice that the second traffic stream is a multi-rate traffic stream. We may at most have three type-2 calls in our system. For state probabilities we need only specify offered traffic, not individual values of arrival rates and service rates. The offered traffic is as usually defined as the traffic carried by an infinite trunk group. For multi-rate traffic we have to consider traffic measured either in connections or channels.

We get a two-dimensional state transition diagram shown in Fig. 7.4. The total sum of all relative state probabilities equals 20.1704. So by normalization we find $p(0,0) = 0.0496$ and we get state probabilities and marginal state probabilities $p(x_1, \cdot)$ and $p(\cdot, x_2)$ (Table 7.2). The global state probabilities are shown in Table 7.3.



$\frac{4}{27}$

$\frac{2}{3}$ $\frac{4}{3}$ $\frac{4}{3}$

$\frac{4}{3}$ $\frac{8}{3}$ $\frac{8}{3}$ $\frac{16}{9}$ $\frac{8}{9}$

1 2 2 $\frac{4}{3}$ $\frac{2}{3}$ $\frac{4}{15}$ $\frac{4}{45}$

Figure 7.4: *Example 7.3.3: 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. 7.1. If we allocate state* $(0,0)$ *the relative probability one, then we find by exploiting local balance the relative state probabilities* $q(x_1, x_2)$ *shown below the state transition diagram.*

$\boldsymbol{p(x_1, x_2)}$	$x_1 = 0$	$x_1 = 1$	$x_1 = 2$	$x_1 = 3$	$x_1 = 4$	$x_1 = 5$	$x_1 = 6$	$\boldsymbol{p(\cdot\,, j)}$
$x_2 = 6$	0.0073							0.0073
$x_2 = 4$	0.0331	0.0661	0.0661					0.1653
$x_2 = 2$	0.0661	0.1322	0.1322	0.0881	0.0441			0.4627
$x_2 = 0$	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

Table 7.2: Detailed state probabilities for the system specified in Table 7.1.

$p(0) = p(0,0)$	$=$	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

Table 7.3: Global state probabilities for the system specified in Table 7.1.

Performance measures for traffic stream 1 (PCT-I traffic):

Due to the *PASTA*-property time congestion (E_1), call congestion (B_1), and 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 (PCT-II traffic):

Time congestion E_2 (proportion of time the system is blocked for stream 2) becomes:

$$\begin{aligned} E_2 &= p(0,6) + p(1,4) + p(2,4) + p(3,2) + p(4,2) + p(5,0) + p(6,0) \\ &= p(5) + p(6)\,, \\ E_2 &= 0.2894\,. \end{aligned}$$

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 in Table 7.2:

$$\begin{aligned} x_t &= \sum_{i=0}^{6} \lambda_2(i) \cdot p(\cdot, i) \\ &= \frac{4}{3} \cdot 0.3647 + \frac{3}{3} \cdot 0.4627 + \frac{2}{3} \cdot 0.1653 + \frac{1}{3} \cdot 0.0073 \\ &= 1.0616\,. \end{aligned}$$

The number of blocked call attempts per time unit becomes (Fig. 7.4):

$$\begin{aligned} x_\ell &= \frac{4}{3} \cdot \{p(5,0) + p(6,0)\} + \frac{3}{3} \cdot \{p(3,2) + p(4,2)\} + \frac{2}{3} \cdot \{p(1,4) + p(2,4)\} + \frac{1}{3} \cdot p(0,6) \\ &= 0.2462\,. \end{aligned}$$

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 in Table 7.2:

$$\begin{aligned} Y_2 &= \sum_{i=0}^{6} i \cdot p(\cdot, i)\,, \\ Y_2 &= 2 \cdot 0.4627 + 4 \cdot 0.1653 + 6 \cdot 0.0073\,, \\ Y_2 &= 1.6306 \quad \text{erlang}\,. \end{aligned}$$

The offered traffic, measured in the unit [*channel*], is $d_2 \cdot A_2 = 2$ erlang (Tab. 7.1). 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 16 (Fig. 7.4). When the number of traffic streams and channels increase, then the number of states increases very fast and we become unable to evaluate the system by calculating the individual state probabilities. In the following section we introduce two classes of algorithms for loss systems which eliminates this problem by aggregation of states.

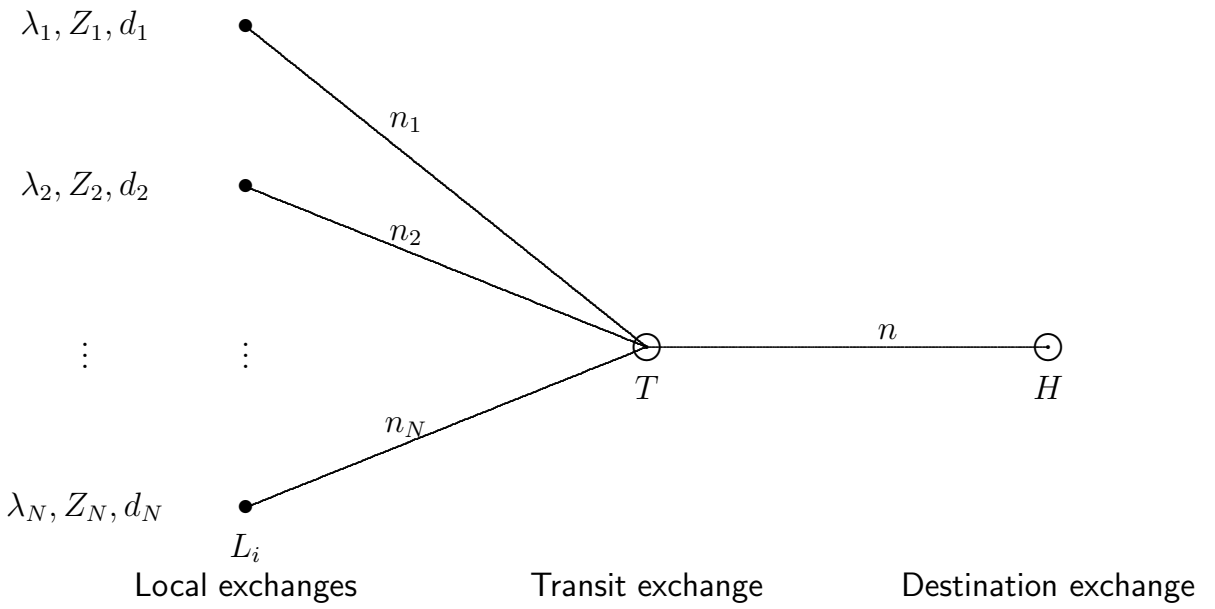


Figure 7.5: *Generalization of the classical teletraffic model to BPP–traffic and multi-rate traffic. The parameters λ_j and Z_j describe the BPP–traffic, and d_j denotes the number of slots required per connection.*

7.4 Convolution Algorithm for loss systems

We now consider a trunk group with a total of n homogeneous channels. Being homogeneous means that they have the same service rate. The channel group is offered N different services, also called streams, or classes. A call (connection) of type i requires d_i channels (slots) during the whole service time, i.e. all d_i channels are occupied and released simultaneously. If less than d_i channels are idle, then the call attempt is blocked (BCC = blocked calls cleared). We define the state of the system $\{x_1, x_2, \ldots, x_N\}$ where x_j is the number of channels occupied by type j which must fulfill the restrictions (7.19) and (7.20).

The arrival processes are general state-dependent Poisson processes. For the j'th arrival process the arrival intensity in state $x_j = i_j \cdot d_j$, when i_j calls (connections) of type j are being served, is $\lambda_j(i_j)$. We may restrict the number i_j of simultaneous calls of type j so that:

$$0 \;\leq\; x_j = i_j \cdot d_j \;\leq\; n_j \;\leq\; n\,.$$

It will be natural to require that x_j is an integral multiple of d_i, i.e. $x_j/d_j = i_j$. This model describes for example the system shown in Fig. 7.5.

The above system fulfills the conditions for reversibility and product form:

$$p(x_1, x_2, \cdots, x_N) = p_1(x_1) \cdot p_2(x_2) \cdot \ldots \cdot p_N(x_N)\,,$$

where the restrictions (7.19) and (7.20) must be fulfilled. Product-form is equivalent to independence between the state probabilities of the traffic streams and therefore we may

convolve the traffic streams to get the global state probability. To aggregate the traffic streams we should express the states in the same bandwidth unit which we call the Basic Bandwidth Unit (*BBU*). Here a BBU is one channel.

We thus express the state probability of stream j as:

$$\underline{p_j} = \{p_j(0), p_j(1), p_j(2), \ldots, p_j(n_j)\} \ ,$$

where $p_j(i) = 0$ when $i \neq k \cdot d_j \, , \quad k = 0, 1, \ldots, \lfloor n_j/d_j \rfloor \, .$

The system mentioned above can be evaluated in an efficient way by the convolution algorithm first introduced in (Iversen, 1987 [46]).

7.4.1 The convolution algorithm

The algorithm is described by the following three steps:

- **Step 1: One–dimensional state probabilities:**
  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. 4 & 5. For traffic stream j we find:

  $$\underline{p_j} = \{p_j(0), \ p_j(1), \ \ldots \ , \ p_j(n_j)\} \, , \quad j = 1, 2, \ldots, N \, . \tag{7.22}$$

  Only the relative values of $p_j(x)$ are of importance, so we may choose $q_j(0) = 1$ and calculate the values of $q_j(x_j)$ relative to $q_j(0)$. If during the recursion a term $q_j(x_j)$ becomes greater than K (e.g. 10^{10}), then we may divide all values $q_j(x_j), \ 0 \leq x_j \leq x$, by K and calculate the following values relatively to these re-scaled values. To avoid any numerical problems in the following it is advisable to normalize the relative state probabilities so that:

  $$p_j(x) = \frac{q_j(x)}{Q_j} \, , \quad x = 0, 1 \ldots, n_j \, , \quad Q_j = \sum_{i=0}^{n_j} q_j(i) \, .$$

  As described in Sec. 4.4 we may normalize at each step to avoid any numerical problems.

- **Step 2: Aggregation of traffic streams:**
  By successive convolutions (convolution operator $*$) we calculate the aggregated state probabilities for the total system excepting traffic stream number j:

  $$\begin{aligned} \underline{q_{N/j}} &= \left\{q_{N/j}(0), \ q_{N/j}(1), \ \ldots \ , q_{N/j}(n)\right\} \, . \\ &= \underline{p_1} * \underline{p_2} * \cdots * \underline{p_{j-1}} * \underline{p_{j+1}} * \cdots * \underline{p_N} \end{aligned} \tag{7.23}$$

We first convolve $\underline{p_1}$ and $\underline{p_2}$ and obtain $\underline{p_{12}}$ which is convolved with $\underline{p_3}$ to obtain $\underline{p_{123}}$, and so on. Both the commutative and the associative laws are valid for the convolution operator, defined in the usual way (Sec. 2.3):

$$\underline{p_i} * \underline{p_j} = \left\{ p_i(0) \cdot p_j(0),\ \sum_{x=0}^{1} p_i(x) \cdot p_j(1-x),\ \cdots,\ \sum_{x=0}^{u} p_i(x) \cdot p_j(u-x) \right\}, \tag{7.24}$$

where we stop at

$$u = \min\{n_i + n_j, n\}\,. \tag{7.25}$$

Notice, that we truncate the state space at state u. Even if $\underline{p_i}$ and $\underline{p_j}$ are normalized, then the result of a convolution is in general not normalized due to the truncation. It is recommended to normalize after every convolution to avoid any numerical problems both during this step and the following.

- **Step 3: Performance measures:**
  Above we have reduced the state space to two traffic streams: $\underline{p_{N/j}}$ and $\underline{p_j}$, and we have product form between these. Thus the problem is reduced to a two-dimensional state transition diagram as e.g shown in Fig. 7.3.

  For stream j we know the state probabilities, arrival rate, and departure rate in every state. For the aggregated stream $p_{N/j}$ we only know the state probabilities; the transition rates between the states are complex and we don't need them in the following. We calculate time congestion E_j, call congestion B_j, and traffic congestion C_j of stream j from the reduced two-dimensional state-transition diagram. This is done during the convolution:

$$\underline{p_N} = \underline{q_{N/j}} * \underline{p_j}\,.$$

  This convolution results in:

$$\underline{q_N(x)} = \sum_{x_j=0}^{x} q_{N/j}(x - x_j) \cdot p_j(x_j) = \sum_{x_j=0}^{x} p_j(x_j \mid x)\,, \tag{7.26}$$

  where for $p_j(x_j \mid x)$, x is the total number of busy channels, and x_j is the number of channels occupied by stream j. Steps 2 – 3 are repeated for every traffic stream. In the following we derive formulæ for E_j, B_j, and C_j.

Time congestion E_j for traffic stream j becomes:

$$E_j = \frac{1}{Q} \cdot \sum_{x \in S_{Ej}} p_j(x_j \mid x)\,. \tag{7.27}$$

where

$$S_{Ej} = \{(x_j, x) \mid x_j \le x \le n \wedge (x_j > n_j - d_j) \vee (x > n - d_i)\}\ ,$$

The summation over S_{Ej} is extended to all states (x_j, x) where calls belonging to class j are blocked. The set $\{x_j > n_j - d_j\}$ corresponds to the states where traffic stream j has utilized

$\boldsymbol{Q_{N/j}(n)}$	$p(0 \mid n)$	0	0	...	0
$\boldsymbol{Q_{N/j}(n-1)}$	$p(0 \mid n-1)$	$p(1 \mid n)$	0	...	0
$\boldsymbol{Q_{N/j}(n-2)}$	$p(0 \mid n-2)$	$p(1 \mid n-1)$	$p(2 \mid n)$	...	0
...	...	...	...	...	...
...	$p(0 \mid n-n_j+1)$	$p(1 \mid n-n_j+2)$	$p(2 \mid n-n_j+3)$	...	0
$\boldsymbol{Q_{N/j}(n-n_j)}$	$p(0 \mid n-n_j)$	$p(1 \mid n-n_j+1)$	$p(2 \mid n-n_j+2)$	...	$p(n_j \mid n)$
...	$p(0 \mid n-n_j-1)$	$p(1 \mid n-n_j)$	$p(2 \mid n-n_j+1)$	...	$p(n_j \mid n-1$
...	...	...	...	...	...
$\boldsymbol{Q_{N/j}(2)}$	$p(0 \mid 2)$	$p(1 \mid 3)$	$p(2 \mid 4)$	...	$p(n_j \mid n_j+2)$
$\boldsymbol{Q_{N/j}(1)}$	$p(0 \mid 1)$	$p(1 \mid 2)$	$p(2 \mid 3)$	...	$p(n_j \mid n_j+1)$
$\boldsymbol{Q_{N/j}(0)}$	$p(0 \mid 0)$	$p(1 \mid 1)$	$p(2 \mid 2)$	...	$p(n_j \mid n_j)$
	$\boldsymbol{p_j(0)}$	$\boldsymbol{p_j(1)}$	$\boldsymbol{p_j(2)}$	**...**	$\boldsymbol{p_j(n_j)}$

Table 7.4: *Convolution algorithm. Exploiting product form we convolve $Q_{N\mathrm{j}(x)}$ and $p_j(x)$ to obtain the global distribution adding contributions in the diagonals and normalize. During this convolution we obtain the detailed performance measures for stream j. Rows have a fixed number of channels occupied by by the aggregated streams N/j and columns have a fixed number x_j of channels occupied by stream j.*

its quota, and $(x > n - d_j)$ corresponds to states with less than d_j idle channels. Q is the normalization constant:

$$Q = \sum_{i=0}^{n} q_N(i)\,.$$

At this stage we usually have normalized the state probabilities so that $Q = 1$.
The truncated state space is shown in Table 7.4, and the global state probability

$$q_N(i) = \sum_{k=0}^{i} q_{N/j}(k) \cdot q_j(i-k)$$

is the total probability mass on diagonal i.

Call congestion $\boldsymbol{B_j}$ for traffic stream j is the ratio between the number of blocked call attempts for traffic stream j and the total number of call attempts for traffic stream j, both for example per time unit. We find:

$$B_j = \frac{\sum_{S_{Ej}} \lambda_j(x_j) \cdot p_j(x_j \mid x)}{\sum_{x=0}^{n} \sum_{x_j=0}^{x} \lambda_j(x_j) \cdot p_j(x_j \mid x)}\,. \tag{7.28}$$

Traffic congestion C_j for traffic stream j: We define as usual the offered traffic as the traffic carried by an infinite trunk group. The carried traffic for traffic stream j is:

$$Y_j = \sum_{x=0}^{n} \sum_{x_j=0}^{x} x_j \cdot p_j(x_j \mid x)\,. \tag{7.29}$$

Thus we find:

$$C_j = \frac{A_j - Y_j}{A_j}\,.$$

Above we have included states which are outside the state space and takes the value zero.

Thus we can find the detailed performance measures for stream j because we know arrival rate and service rate of stream j for every state in the reduced state transition diagram in Table 7.4. For the aggregated stream we are able to calculate the total carried traffic and thus the aggregated traffic congestion. But we are not able to calculate time congestion or call congestion, because we don't know state transitions for the aggregated stream N/j. We only know the state probabilities and that the product form is valid.

The algorithm was first implemented in the PC-tool *ATMOS* (Listov–Saabye & Iversen, 1989 [85]). The storage requirements are 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 [48]) 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 7.4.1: De-convolution
In principle we may obtain $\underline{q_{N/j}}$ from $\underline{q_N}$ by de-convolving $\underline{p_j}$ and then calculate the performance measures during the re-convolution of $\underline{p_j}$ and $\underline{q_{N/j}}$. In this way we need not repeat all the convolutions (7.23) 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 7.4.2: 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 7.3.3 we add a Pascal stream with class limitation as shown in Tab. 7.5 (cf. Example 5.7.1). We want to calculate the performance measures of traffic stream 3.

- **Step 1**: We calculate the state probabilities $p_j(x)$, $(x = 1, 2, \ldots, n_j)$ of each traffic stream j $(j = 1, 2, 3)$ as if it were alone. The results are given in Tab. 7.6.

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 7.5: *A Pascal traffic stream (Example 5.7.1) is offered to the same trunk as the two traffic streams of Tab. 7.1.*

- **Step 2:** We evaluate the convolution of $p_1(x_1)$ with $p_2(x_2)$, $p_1 * p_2(x_{12})$, truncate the state space at $n = 6$, and normalize the probabilities so that we obtain $\underline{p}_{12}$ shown in the Tab. 7.6. Notice that this is the result obtained in Example 7.3.3.

- **Step 3:** We convolve $p_{12}(x_{12})$ with $p_3(x_3)$, truncate at n, and obtain $q_{123}(x_{123})$ as shown in Tab. 7.6.

State	Probabilities		$\boldsymbol{q_{12}(x)}$	Normal.	Prob.	$\boldsymbol{q_{123}(x)}$	Normal.
x	$p_1(x)$	$p_2(x)$	$p_1 * p_2$	$p_{12}(x)$	$p_3(x)$	$p_{12} * p_3$	$p_{123}(x)$
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 7.6: *Convolution algorithm applied to Example 7.4.2. The state probabilities for the individual traffic streams have been calculated in the examples 4.5.1, 5.5.1 and 5.7.1.*

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:

$$\begin{aligned} E_3 &= \frac{q_{123}(6) + p_3(4) \cdot \{p_{12}(0) + p_{12}(1)\}}{0.8678} \\ &= \frac{0.1535 + 0.0279 \cdot \{0.0496 + 0.0992\}}{0.8678}\,, \\ E_3 &= 0.1817\,. \end{aligned}$$

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:

$$\begin{aligned} Y_3 &= \frac{1}{0.8678} \left\{ \sum_{x_3=1}^{4} x_3 \cdot p_3(x_3) \sum_{x_{12}=0}^{6-x_{12}} p_{12}(j) \right\}, \\ Y_3 &= \frac{0.6174}{0.8678} = 0.7115\,. \end{aligned}$$

As the offered traffic is $A_3 = 1$, we get:

Traffic congestion:

$$\begin{aligned} C_3 &= \frac{1 - 0.7115}{1}\,, \\ C_3 &= 0.2885\,. \end{aligned}$$

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 normalized probabilities from Tab. 7.6 we get $\{\lambda_3(i) = (S_3 - i)\,\gamma_3\}$:

$$\begin{aligned} x_\ell &= \lambda_3(0) \cdot \{p_3(0) \cdot p_{12}(6)\} \\ &\quad + \lambda_3(1) \cdot \{p_3(1) \cdot p_{12}(5)\} \\ &\quad + \lambda_3(2) \cdot \{p_3(2) \cdot p_{12}(4)\} \\ &\quad + \lambda_3(3) \cdot \{p_3(3) \cdot p_{12}(3)\} \\ &\quad + \lambda_3(4) \cdot p_3(4) \cdot \{p_{12}(2) + p_{12}(1) + p_{12}(0)\}\,, \end{aligned}$$

$$
\begin{aligned}
x_\ell &= 0.2503\,. \\
x_t &= \lambda_3(0)\cdot p_3(0)\cdot \sum_{j=0}^{6} p_{12}(j) \\
&\quad + \lambda_3(1)\cdot p_3(1)\cdot \sum_{j=0}^{5} p_{12}(j) \\
&\quad + \lambda_3(2)\cdot p_3(2)\cdot \sum_{j=0}^{4} p_{12}(j) \\
&\quad + \lambda_3(3)\cdot p_3(3)\cdot \sum_{j=0}^{3} p_{12}(j) \\
&\quad + \lambda_3(4)\cdot p_3(4)\cdot \sum_{j=0}^{2} p_{12}(j)\,, \\
x_t &= 1.1763\,.
\end{aligned}
$$

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. 7.7 and Tab. 7.8. 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 $\boldsymbol{n = 6}$						
	Offered traffic	Peaked ness	Maximum allocation	Slot size	Mean holding time	Sources	beta
$\boldsymbol{j}$	$\boldsymbol{A_j}$	$\boldsymbol{Z_j}$	$\boldsymbol{n_j}$	$\boldsymbol{d_j}$	$\boldsymbol{\mu_j^{-1}}$	$\boldsymbol{S_j}$	$\boldsymbol{\beta_j}$
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	-0.3333

Table 7.7: *Input data to ATMOS for Example 7.4.2 with three traffic streams.*

Output	Call congestion	Traffic congestion	Time congestion	Carried traffic
j	B_j	C_j	E_j	Y_j
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 7.8: *Output data from ATMOS for the input data in Tab. 7.7.*

Example 7.4.3: Large-scale example
To illustrate the tool *"ATMOS"* we consider in Tab. 7.9 and Tab. 7.10 an example with 1536 trunks and 24 traffic streams. We notice that the time congestion is independent of peakedness Z_j and proportional to the slot-size d_j, because we often have:

$$p(n) \approx p(n-1) \approx \ldots \approx p(n-d_j) \quad \text{for} \quad d_j \ll n\,. \tag{7.30}$$

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 $(d_j \cdot Z_j)$, using the single-slot Poisson traffic as reference value $(d_j = Z_j = 1)$. We notice that the traffic congestion is proportional to $d_j \cdot Z_j$, which is the usual assumption when using the Equivalent Random Traffic (*ERT*) method (Sec. 6.4.3). 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-rate 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 characterizing the performance of the system. Below in Example 7.5.1 we calculate the total traffic congestion using *Fredericks & Hayward's* method for multi-rate traffic (Sec. 7.5). □

7.5 Fredericks-Haywards's method

Basharin & Kurenkov has extended Fredericks-Hayward's method (Sec. 6.5) to include multi-slot (multi-rate) traffic. Let every connection require d channels during the whole holding time from start to termination. Then by splitting this traffic into d identical sub-streams (Sec. 6.4) 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.

If a call uses 1 channels instead of d channels, then the mean value becomes d times smaller and the variance d^2 times smaller (change of scale, Example 2.4.2). Therefore, the peakedness becomes d times smaller. If furthermore the arrival process has a peakedness Z, then by

Input	Total # of channels $n = 1536$						
	Offered traf.	Peakedness	Max. sim. #	Channels/call	mht	Sources	
j	A_j	Z_j	n_j	d_j	μ_j	S_j	β_j
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 7.9: *Input data for Example 7.4.3 with 24 traffic streams and 1536 channels. The maximum number of simultaneous calls of type j (n_j) 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
j	B_j	C_j	E_j	Y_j	$C_j/(d_j Z_j)$
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 7.10: *Output for Example 7.4.3 with input data given in Tab. 7.9. As mentioned earlier in Example 7.5.1, 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.*

splitting into $d \cdot Z$ traffic streams the traffic process becomes a single-slot traffic process with peakedness one, which we evaluate by Erlang's B-formula.

$$\begin{aligned}(n,\ A,\ Z,\ d) \ &\sim \ \left(\frac{n}{d\,Z},\ \frac{A}{d\,Z},\ 1,\ 1\right) \sim \left(\frac{n}{d},\ \frac{A}{d},\ Z,\ 1\right) \\ &\sim \ \left(\frac{n}{Z},\ \frac{A}{Z},\ 1,\ d\right) \quad \sim \left(n,\ \frac{A}{Z},\ 1,\ d \cdot Z\right).\end{aligned} \tag{7.31}$$

The last equivalence show that by increasing number the bandwidth d by the factor Z, we may keep the number of channels n constant and get an arrival process with $Z = 1$. If we have more traffic streams offered to the same group, then we may keep the number of channels fixed. The bandwidth $d \cdot Z$ is in general not integral. Then we should choose a basic bandwidth unit (BBU)so that both n and $d{\cdot}Z$ approximately become integral multiples of this unit. The smaller the bandwidth unit (granularity) is chosen, the better the approximation becomes. However, it is recommended to aggregate all traffic streams into one single-slot Poisson traffic stream and calculate the total traffic congestion. Then this may be split up into traffic congestion for each stream as shown in Example 7.4.3

Example 7.5.1: Multi-slot traffic
In example 7.4.3 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) of the theoretical value 5.950 %. □

7.6 State space based algorithms

The convolution algorithm is based on aggregation of traffic streams, where we end up with a traffic stream which is the aggregation of all traffic streams except the one which we are interested in. Another approach is to aggregate the state space into global state probabilities.

7.6.1 Fortet & Grandjean (Kaufman & Robert) algorithm

In case of Poisson arrival processes the algorithm becomes very simple by generalizing (7.11). Let $p_j(x)$ denote the contribution of stream j to the global state probability $p(x)$:

$$p(x) = \sum_{j=1}^{N} p_j(x)\,. \tag{7.32}$$

Thus the average number of channels occupied by stream j when the system is in global state x is $x \cdot p_j(x)$. Let traffic stream j have the slot-size d_j. Due to reversibility we will have local balance for every traffic type. The local balance equation becomes:

$$\lambda_j \cdot p(x - d_j) = \frac{x}{d_j} \cdot p_j(x) \cdot \mu_j \,, \qquad x = d_j, d_j + 1, \ldots n \,. \tag{7.33}$$

The left-hand is the flow from global state $[\, x - d_j \,]$ to state $[\, x \,]$ due to arrivals of type j. The right-hand side is the flow from state $[\, x \,]$ to state $[\, x - d_j \,]$ due to departures of type j calls. The average number of channels occupied by stream j in global state x is not an integer because it is a weighted sum over more state probabilities. From (7.33) we get:

$$p_j(x) = \frac{1}{x}\, d_j\, A_j \cdot p(x - d_j) \,. \tag{7.34}$$

The total state probability $p(x)$ is obtained by summing up over all traffic streams (7.32):

$$p(x) = \frac{1}{x} \sum_{j=1}^{N} d_j\, A_j \cdot p(x - d_j) \,, \qquad p(x) = 0 \quad \text{for} \quad x < 0 \,. \tag{7.35}$$

This is *Fortet & Grandjean's algorithm* (Fortet & Grandjean, 1964 [33]) The algorithm is usually called *Kaufman & Roberts' algorithm*, as it was re-discovered by these authors in 1981 (Kaufman, 1981 [67]) (Roberts, 1981 [105]).

7.6.2 Generalized algorithm

The above model can easily be generalized to *BPP*-traffic (Iversen, 2005 [50])

$$\frac{x\, p_j(x)}{d_j} \cdot \mu_j = p(x - d_j) \cdot S_j\, \gamma_j - p_j(x - d_j) \cdot \frac{x - d_j}{d_j} \cdot \gamma_j \,. \tag{7.36}$$

On the right-hand side the first term assumes that all type j sources are idle during one time unit. As we know

$$\frac{x - d_j}{d_j} \cdot p_j(x - d_j)$$

type j sources on the average are busy in global state $x - d_j$ we reduce the first term with the second term to get the right value. Thus we get:

$$p(x) \;=\; \begin{cases} 0 & x < 0 \\ p(0) & x = 0 \\ \displaystyle\sum_{j=1}^{N} p_j(x) & x = 1, 2, \ldots, n \end{cases} \tag{7.37}$$

$$\text{where} \qquad p_j(x) \;=\; \frac{d_j}{x} \cdot \frac{S_j\, \gamma_j}{\mu_j} \cdot p(x - d_j) - \frac{x - d_j}{x} \cdot \frac{\gamma_j}{\mu_j} \cdot p_j(x - d_j) \tag{7.38}$$

$$p_j(x) \;=\; 0 \qquad x < d_j \,. \tag{7.39}$$

The state probability $p(0)$ is obtained by the normalization condition:

$$\sum_{i=0}^{n} p(i) = p(0) + \sum_{i=1}^{n} \sum_{i=1}^{N} p_j(i) = 1\,, \tag{7.40}$$

as $p_j(0) = 0$, whereas $p(0) \neq 0$. Above we have used the parameters (S_j, β_j) to characterize the traffic streams. Alternatively we may also use (A_j, Z_j) related to (S_j, β_j) by the formulæ (5.22) – (5.25). Then (7.38) becomes:

$$p_j(x) = \frac{d_j}{x} \cdot \frac{A_j}{Z_j} \cdot p(x - d_j) - \frac{x - d_j}{x} \cdot \frac{1 - Z_j}{Z_j} \cdot p_j(x - d_j) \tag{7.41}$$

For Poisson arrivals we of course get (7.35). In practical evaluation of the formula we will use normalization in each step as described in Sec. 4.4.1. This results in a very accurate and effective algorithm. In this way also the number of operations and the memory requirements become very small, as we only need to store the d_i previous state probabilities of traffic stream i, and the $\max\{d_i\}$ previous values of the global state probabilities. The number of operations is linear in number of channels and number of traffic streams and thus extremely effective.

Performance measures

By this algorithm we are able to obtain performance measures for each individual traffic stream.

Time congestion:
Call attempts of stream j require d_j idle channel and will be blocked with probability:

$$E_j = \sum_{x=n-d_j+1}^{n} p(x)\,. \tag{7.42}$$

Traffic congestion:
From the state probabilities $p_j(x)$ we get the total carried traffic of stream j:

$$Y_j = \sum_{x=1}^{n} x \cdot p_j(x)\,. \tag{7.43}$$

Thus the traffic congestion of stream j becomes:

$$C_j = \frac{A_j \cdot d_j - Y_j}{A_j \cdot d_j}\,. \tag{7.44}$$

The total carried traffic is

$$Y = \sum_{j=1}^{N} Y_j\,, \tag{7.45}$$

so the total traffic congestion becomes:

$$C = \frac{A - Y}{A}\,, \tag{7.46}$$

where A is the total offered traffic measured in channels:

$$A = \sum_{j=1}^{N} d_j\, A_j\,.$$

Call congestion:
This is obtained from the traffic congestion by using (5.49):

$$B_j = \frac{(1+\beta_j)\, C_j}{1+\beta_j\, C_j}\,. \tag{7.47}$$

The total call congestion cannot be obtained by this formula as we do not have a global value of β. But from individual carried traffic and individual call congestion we may find the total number of offered calls and accepted calls for each stream, and from this we get the total call congestion.

Example 7.6.1: Generalized algorithm
We evaluate Example 7.3.3 by the general algorithm. For the Poisson traffic (stream 1) we have $d = 1$, $A = 2$, and $Z = 1$. We thus get:

$$q_1(x) = \frac{2}{x} \cdot q_1(x-1)\,, \qquad q_1(0) = 0\,, \quad q(0) = 1\,.$$

The total relative state probability is $q(x) = q_1(x) + q_2(x)$. For the Engset traffic (stream 2) we have $d = 2$, $A = 1$, and $Z = 0.75$. We then get:

$$q_2(x) = \frac{2}{x} \cdot \frac{1}{0.75} \cdot q(x-2) - \frac{x-2}{x} \cdot \frac{1}{3} \cdot q_2(x-2)\,, \quad q_2(0) = q_2(1) = 0\,.$$

Table 7.11 shows the non-normalized relative state probabilities when we let state zero equal to one. Table 7.12 shows the normalized state probabilities and the carried traffic of each stream in each state. In a computer program we would normalize state probabilities after each iteration (increasing number of channels by one) and calculate the aggregated carried traffic for each stream. This traffic value should of course also be normalized in each step. In this way we only need to store the previous d_i values and the carried traffic of each traffic stream. We get the following performance measures, which of course are the same as obtained by convolution algorithm.

$$\begin{aligned}
E_1 &= p(6) && = 0.1219\\
E_2 &= p(5) + p(6) && = 0.2894\\
C_1 &= \tfrac{2\cdot 1 - 1.7562}{2\cdot 1} && = 0.1219\\
C_2 &= \tfrac{1\cdot 2 - 1.6306}{1\cdot 2} && = 0.1847\\
B_1 &= \tfrac{(1+0)\cdot 0.1219}{1+0\cdot 0.1219} && = 0.1219\\
B_1 &= \tfrac{(1+1/3)\cdot 0.1847}{1+(1/3)\cdot 0.1847} && = 0.2320
\end{aligned}$$

□

State	Poisson				Engset							Total
x	$q_1(x)$				$q_2(x)$							$q(x)$
	$\frac{2}{x} \cdot q(x-1) = q_1(x)$				$\frac{2}{x} \cdot \frac{4}{3} \cdot q(x-2) - \frac{x-2}{x} \cdot \frac{1}{3} \cdot q_2(x-2) = q_2(x)$							
0				0							0	1
1	$\frac{2}{1}$	$\cdot$	1	$= 2$							0	2
2	$\frac{2}{2}$	$\cdot$	2	$= 2$	$\frac{2}{2}$	$\cdot \frac{4}{3} \cdot$	1	$- \frac{0}{2}$	$\cdot \frac{1}{3} \cdot$	0	$= \frac{4}{3}$	$\frac{10}{3}$
3	$\frac{2}{3}$	$\cdot$	$\frac{10}{3}$	$= \frac{20}{9}$	$\frac{2}{3}$	$\cdot \frac{4}{3} \cdot$	2	$- \frac{1}{3}$	$\cdot \frac{1}{3} \cdot$	0	$= \frac{16}{9}$	4
4	$\frac{2}{4}$	$\cdot$	4	$= 2$	$\frac{2}{4}$	$\cdot \frac{4}{3} \cdot$	$\frac{10}{3}$	$- \frac{2}{4}$	$\cdot \frac{1}{3} \cdot$	$\frac{4}{3}$	$= 2$	4
5	$\frac{2}{5}$	$\cdot$	4	$= \frac{8}{5}$	$\frac{2}{5}$	$\cdot \frac{4}{3} \cdot$	4	$- \frac{3}{5}$	$\cdot \frac{1}{3} \cdot$	$\frac{16}{9}$	$= \frac{16}{9}$	$\frac{152}{45}$
6	$\frac{2}{6}$	$\cdot$	$\frac{152}{45}$	$= \frac{152}{135}$	$\frac{2}{6}$	$\cdot \frac{4}{3} \cdot$	4	$- \frac{4}{6}$	$\cdot \frac{1}{3} \cdot$	2	$= \frac{180}{135}$	$\frac{332}{135}$
Total												$\frac{2723}{135}$

Table 7.11: *Example 7.6.1: relative state probabilities for Example 7.3.3 evaluated by the generalized algorithm.*

State	Poisson		Engset		Total	
x	$p_1(x)$	$x \cdot p_1(x)$	$p_2(x)$	$x \cdot p_2(x)$	$p(x)$	$x \cdot p(x)$
0	0.0000	0.0000	0.0000	0.0000	0.0496	0.0000
1	0.0992	0.0992	0.0000	0.0000	0.0992	0.0992
2	0.0992	0.1983	0.0661	0.1322	0.1653	0.3305
3	0.1102	0.3305	0.0881	0.2644	0.1983	0.5949
4	0.0992	0.3966	0.0992	0.3966	0.1983	0.7932
5	0.0793	0.3966	0.0881	0.4407	0.1675	0.8373
6	0.0558	0.3349	0.0661	0.3966	0.1219	0.7315
Total		1.7562		1.6306	1.0000	3.3867

Table 7.12: *Example 7.6.1: absolute state probabilities and carried traffic $y_i(x) = x \cdot p_i(x)$ for Example 7.3.3 evaluated by the generalized algorithm.*

7.6.3 Batch Poisson arrival process

When we have more traffic streams the state-based algorithm is modified by exploiting the analogy with the Pascal distribution. Inserting A (5.76) and Z (5.77) we get:

$$p_j(x) = \frac{d_j}{x} \cdot \frac{\lambda_j}{\mu_j} \cdot p(x-d_j) + (1-p_j) \cdot \frac{x-d_j}{x} \cdot p_j(x-d_j)\,, \quad 0 \le x \le n\,, \tag{7.48}$$

where p_j, λ_j and μ_j are parameters of the Batched Poisson process. Thus the state-based algorithm for *BPP* (Binomial, Poisson, Pascal) is generalized to include Batch Poisson process in a simple way.
This section is to be elaborated in further details, in particular the performance measures.

7.7 Final remarks

The convolution algorithm for loss systems was first published in (Iversen, 1987 [46]). A similar approach to a less general model was published in two papers by Ross & Tsang (1990 [107]), (1990 [108]) without reference to this original paper from 1987 even though it was known by the authors.

The generalized algorithm in Sec. 7.6.2 is new (Iversen, 2007 [51]) and includes Delbrouck's algorithm (Delbrouck, 1983 [23]) which is more complex to evaluate. Compared with all other algorithms the generalized algorithm requires much less memory and operations to evaluate. By normalizing the state probabilities in each iteration we get a very accurate and simple algorithm. In principle, we may apply the generalized algorithm for *BPP*–traffic to calculate the global state probabilities for $(N-1)$ traffic streams and then use the convolution algorithm to calculate the performance measures for the remaining traffic stream we want to evaluate.

The convolution algorithm allows for minimum and maximum allocation of channels to each traffic stream, but it does not allow for restrictions based on global states. It also allows for arbitrary state-dependent arrival processes.

The generalized algorithm does not keep account of the number of calls of the individual traffic stream, but allows for restrictions based on global states, e.g. trunk reservation.

Updated 2010-03-23

Chapter 8

Dimensioning of telecom networks

Network planning includes designing, optimizing, and operating telecommunication networks. In this chapter we will consider traffic engineering aspects of network planning. In Sec. 8.1 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. 8.2) and traffic routing (Sec. 8.3).

In Sec. 8.4 we consider approximate calculation of end-to-end blocking probabilities, and describe the Erlang fix-point method (reduced load method). Sec. 8.5 generalizes the convolution algorithm introduced in Chap. 7 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. 8.6 we consider service-protection mechanisms. Finally, in Sec. 8.7 we consider optimizing of telecommunication networks by applying *Moe's principle*.

8.1 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. 8.1. 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 ?)

FROM	TO 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 8.1: *A traffic matrix. The total incoming traffic is equal to the total outgoing traffic.*

8.1.1 Kruithof's double factor method

Let us assume we know the actual traffic matrix and that we have a forecast for future row sums $O(i)$ (Originating) and column sums $T(i)$ (Terminating), i.e. the total outgoing and incoming 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 [79]) 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:

$$A_{ij} \Leftarrow A_{ij} \cdot \frac{S_1}{S_0}\,, \tag{8.1}$$

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 8.1.1: 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.
- *Fractionizing*. The single exchanges can be split into sub-exchanges or be aggregated into larger exchanges without influencing the result. This property is not exactly fulfilled for Kruithof's double factor method, but the deviations are small.

8.2 Topologies

In Chap. 1 we have described the basic topologies as star net, mesh net, ring net, hierarchical net and non-hierarchical net.

8.3 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.

8.4 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:

$$E_j = 1 - \prod_{i \in \mathcal{R}} (1 - E_i)\,, \tag{8.2}$$

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:

$$E_j \approx \sum_{i \in \mathcal{R}} E_i\,. \tag{8.3}$$

8.4.1 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.

8.5 Exact end-to-end calculation methods

Circuit switched telecommunication networks with direct routing have the same complexity as queueing networks with more chains. (Sec. 12.8) and Tab. 12.3). It is necessary to keep account of the number of busy channels on each link. Therefore, the maximum number of states becomes:

$$\prod_{i=1}^{K}(n_i + 1)\,. \tag{8.4}$$

Link	Route				Number of channels
	1	2	···	N	
1	d_{11}	d_{21}	···	d_{N1}	n_1
2	d_{12}	d_{22}	···	d_{N2}	n_2
·	·	·		·	·
···	···	···		···	···
·	·	·		·	·
K	d_{1K}	d_{2K}	···	d_{NK}	n_K

Table 8.2: *In a circuit switched telecommunication network with direct routing d_{ij} denotes the slot-size (bandwidth demand) of route j on link i (cf. Tab. 12.3).*

8.5.1 Convolution algorithm

The convolution algorithm described in Chap. 7 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.

8.6 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 decentralized 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. 3.6). 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. 10). In telecommunication networks two strategies are common: trunk reservation and virtual channels protection.

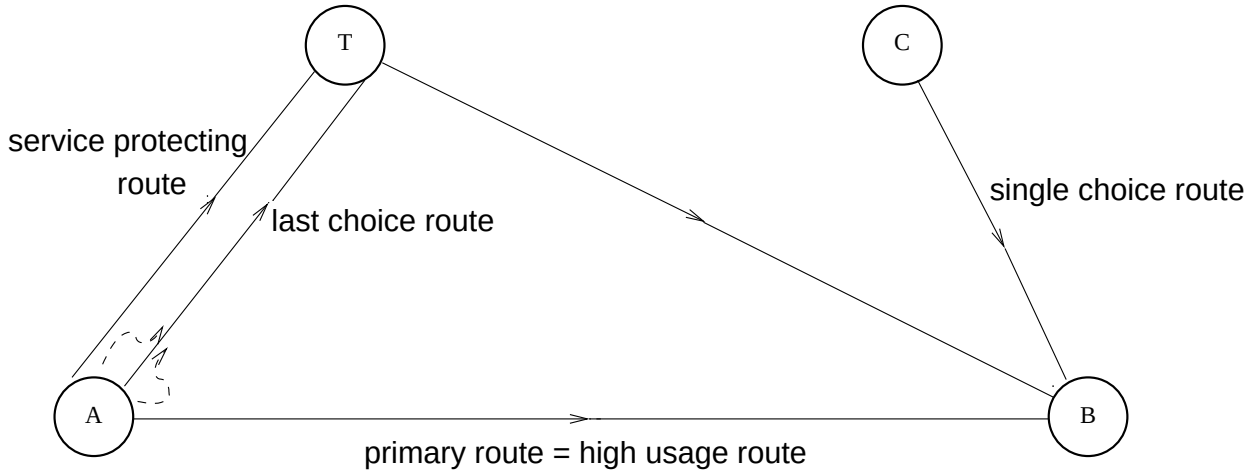


Figure 8.1: *Alternative traffic routing (cf. example 8.6.2). Traffic from A to B is partly carried on the direct route (primary route = high usage route), partly on the secondary route via the transit exchange T.*

8.6.1 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. 8.1), 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. In some states calls of a type having been accepted earlier in lower states may depart but not be accepted, and thus the process is non-reversible. We cannot apply the convolution algorithm developed in Sec. 7.4 for this case, but the generalized algorithm in Sec. 7.6.2 can easily be modified by letting $p_i(x) = 0$ when $x \geq n - r_i$.

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 8.6.1: 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. □

8.6.2 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.

8.7 Moe's principle

Theorem 8.1 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. 8.7.1 and question *b* in Sec. 8.7.2. 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 [59]).

8.7.1 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:

$$C_i = c_{0i} + c_i \cdot n_i, \qquad i = 1, 2, \ldots, k\,. \tag{8.5}$$

The total cost of cables then becomes:

$$C\left(n_1, n_2, \ldots, n_k\right) = C_0 + \sum_{i=1}^{k} c_i \cdot n_i\,, \tag{8.6}$$

where C_0 is a constant.

The total carried traffic is a function of the number of channels:

$$Y = f\left(n_1, n_2, \ldots, n_k\right)\,. \tag{8.7}$$

As we always operate with limited resources we will have:

$$\frac{\partial f}{\partial n_i} = \mathrm{D}_i f > 0\,. \tag{8.8}$$

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 minimize C for a given total carried traffic Y:

$$\min\{C\} \quad \textit{given} \quad Y = f(n_1, n_2, \ldots, n_k) \,. \tag{8.9}$$

By applying the Lagrange multiplier (shadow prices) ϑ, where we introduce $G = C - \vartheta \cdot f$, this is equivalent to:

$$\min\{G(n_1, n_2, \ldots, n_k)\} = \min\{C(n_1, n_2, \ldots, n_k) - \vartheta[f(n_1, n_2, \ldots, n_k) - Y]\} \tag{8.10}$$

A necessary condition for the minimum solution is:

$$\frac{\partial G}{\partial n_i} = c_i - \vartheta \frac{\partial f}{\partial n_i} = c_i - \vartheta \mathrm{D}_i f = 0, \qquad i = 1, 2, \ldots, k \,, \tag{8.11}$$

or

$$\frac{1}{\vartheta} = \frac{\mathrm{D}_1 f}{c_1} = \frac{\mathrm{D}_2 f}{c_2} = \cdots = \frac{\mathrm{D}_k f}{c_k} \,. \tag{8.12}$$

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 (4.52).

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 [59]). 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 (8.12) we shall replace c_i by c_i/g_i.

8.7.2 Optimum carried traffic

Let us consider the case where the carried traffic, which is a function of the number of channels (8.7) is Y. If we denote the revenue with $R(Y)$ and the costs with $C(Y)$ (8.6), then the profit becomes:

$$P(Y) = R(Y) - C(Y) \,. \tag{8.13}$$

A necessary condition for optimal profit is:

$$\frac{dP(Y)}{dY} = 0 \qquad \Rightarrow \qquad \frac{dR}{dY} = \frac{dC}{dY} \,, \tag{8.14}$$

i.e. the marginal income should be equal to the marginal cost.

Using:

$$P\left(n_1, n_2, \ldots, n_k\right) = R\left(f\left(n_1, n_2, \ldots, n_k\right)\right) - \left\{C_0 + \sum_{i=1}^{k} c_i \cdot n_i\right\}, \tag{8.15}$$

the optimal solution is obtained for:

$$\frac{\partial P}{\partial n_i} = \frac{dR}{dY} \cdot \mathrm{D}_i f - c_i = 0, \qquad i = 1, 2, \ldots, k\,, \tag{8.16}$$

which by using (8.12) gives:

$$\frac{dR}{dY} = \vartheta\,. \tag{8.17}$$

The factor ϑ given by (8.12) 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 (4.54).

Example 8.7.1: 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 (8.12) 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 8.7.2: Triangle optimization
This is a classical optimization of a triangle network using alternative traffic routing (Fig. 8.1). 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) $A \to T \to B$, 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 $A \to T \to B$ already carries traffic to and from other destinations, and we denote the marginal utilization for a channel

on this route by a. We assume it is independent of the additional traffic, which is blocked from A $\rightarrow$ B.

According to (8.12), 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. 13 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 emphasized 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:

$$\begin{aligned} E_n\left(\bar{A}_{30}\right) &\leq 0.01\,, \\ E_n\left(\bar{A}_{5}\right) &\leq 0.07\,. \end{aligned} \tag{8.18}$$

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 9

Markovian queueing systems

In this chapter we consider traffic to a system with n identical servers, full accessibility, and an a queue with an infinite number of waiting positions. When all n servers are busy, an arriving customer joins the 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 models as in Chaps. 4 & 5.

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*. In this system the carried traffic will be equal to the offered traffic as no customers are blocked. The probability of delay, mean queue length, mean waiting time, carried traffic per channel, and improvement functions will be dealt with in Sec. 9.2. In Sec. 9.3 *Moe's principle* is applied for optimizing the system. The waiting time distribution is derived for the basic queueing discipline, First–Come First–Served (*FCFS*) in Sec. 9.4. In Sec. 9.5 we summarize the results for the important single-server system *M/M/1*.

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. 9.6. This model is widely applied for dimensioning of computer systems, terminal systems, flexible manufacturing system (*FMS*), etc. Palm's machine repair model is optimized in Sec. 9.7. The waiting time distribution for Palm's model with *FCFS* queueing discipline is derived in Sec. 9.8.

9.1 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

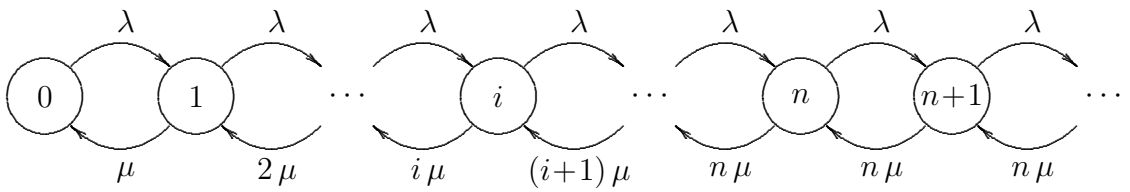


Figure 9.1: *State transition diagram of the M/M/n delay system having n servers and an unlimited number of waiting positions.*

waiting in the queue). We are interested in the steady state probabilities of the system. By the procedure described in Sec. 4.4 we set up the state transition diagram shown in Fig. 9.1. Assuming statistical equilibrium, the cut equations become:

$$\begin{aligned}
\lambda \cdot p(0) &= \mu \cdot p(1)\,, \\
\lambda \cdot p(1) &= 2\,\mu \cdot p(2)\,, \\
&\vdots \quad \vdots \quad \vdots \\
\lambda \cdot p(i) &= (i{+}1)\,\mu \cdot p(i{+}1)\,, \\
&\vdots \quad \vdots \quad \vdots \\
\lambda \cdot p(n{-}1) &= n\,\mu \cdot p(n)\,, \\
\lambda \cdot p(n) &= n\,\mu \cdot p(n{+}1)\,, \\
&\vdots \quad \vdots \quad \vdots \\
\lambda \cdot p(n+j) &= n\,\mu \cdot p(n{+}j{+}1)\,. \\
&\vdots \quad \vdots \quad \vdots
\end{aligned} \tag{9.1}$$

As $A = \lambda/\mu$ is the offered traffic, we get:

$$p(i) = \begin{cases} p(0) \cdot \dfrac{A^i}{i!}\,, & 0 \le i \le n\,, \\ p(n) \cdot \left(\dfrac{A}{n}\right)^{i-n} = p(0) \cdot \dfrac{A^i}{n! \cdot n^{i-n}}\,, & i \ge n\,. \end{cases} \tag{9.2}$$

By normalization 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. Statistical equilibrium is only obtained for:

$$A < n \, . \tag{9.3}$$

Otherwise, the queue will continue to increase towards infinity. We obtain:

$$p(0) = \frac{1}{\sum_{i=0}^{n-1} \frac{A^i}{i!} + \frac{A^n}{n!} \frac{n}{n-A}} \, , \qquad A < n \, , \tag{9.4}$$

and equations (9.2) and (9.4) yield the steady state probabilities $p(i), \; i > 0$.

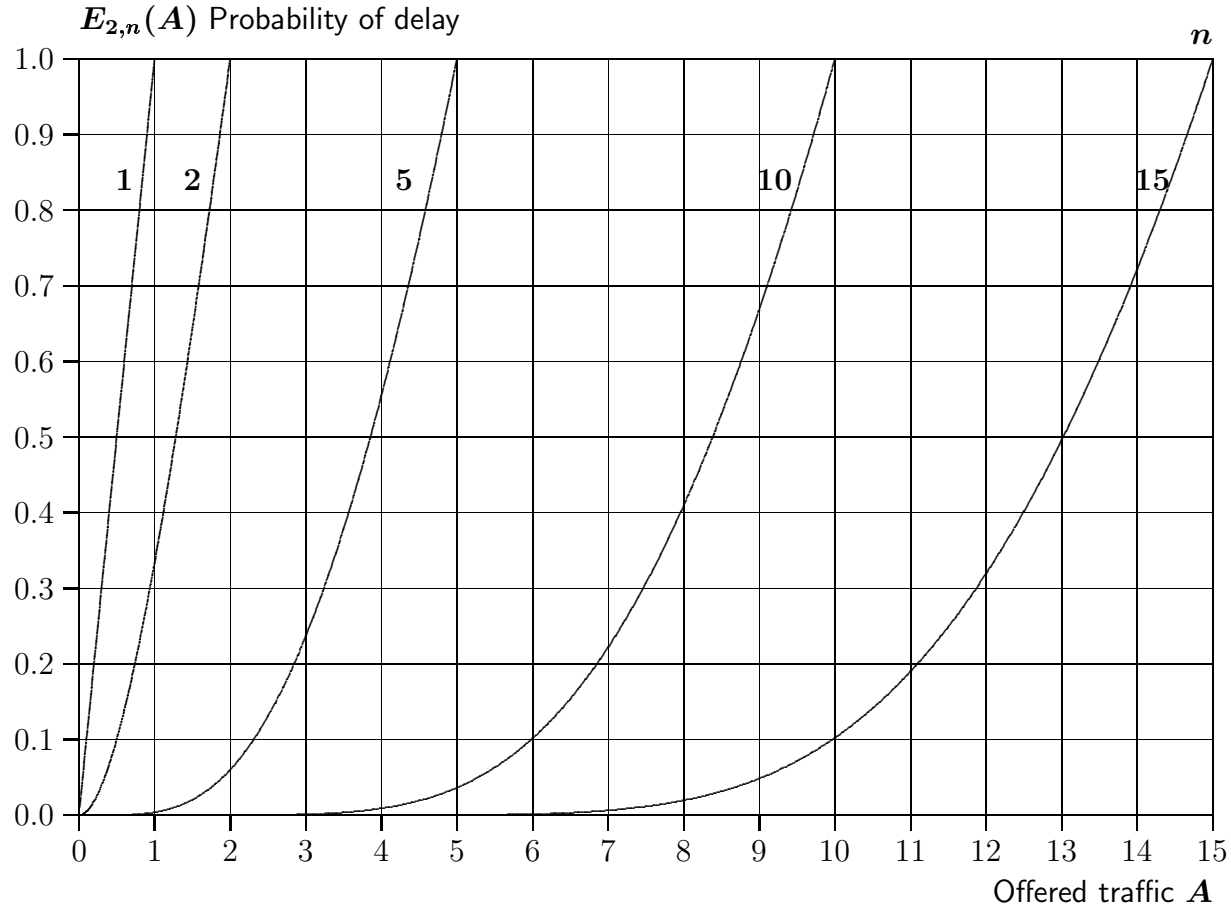


Figure 9.2: *Erlang's C–formula for the delay system M/M/n. The probability $E_{2,n}(A)$ for a positive waiting time is shown as a function of the offered traffic A for different values of the number of servers n.*

9.2 Traffic characteristics of delay systems

For evaluation of the performance of the system, several characteristics have to be considered. They are expressed by the steady-state probabilities.

9.2.1 Erlang's C-formula

The stationary Poisson arrival process is independent of the state of the system, and therefore 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 Averages). The waiting time is a random variable denoted by $\mathcal{W}$. For an arbitrary arriving customer we have:

$$\begin{aligned} E_{2,n}(A) &= p\{\mathcal{W} > 0\} \\ &= \frac{\sum_{i=n}^{\infty} \lambda\, p(i)}{\sum_{i=0}^{\infty} \lambda\, p(i)} = \sum_{i=n}^{\infty} p(i) \\ &= p(n) \cdot \frac{n}{n-A}\,. \end{aligned} \tag{9.5}$$

Erlang's C–formula (1917):

$$E_{2,n}(A) = \frac{\dfrac{A^n}{n!}\,\dfrac{n}{n-A}}{1 + \dfrac{A}{1} + \dfrac{A^2}{2!} + \cdots + \dfrac{A^{n-1}}{(n-1)!} + \dfrac{A^n}{n!}\,\dfrac{n}{n-A}}\,, \quad A < n\,. \tag{9.6}$$

This probability of delay depends only upon $A = \lambda/\mu$, not upon the parameters λ and μ 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:

$$S_n = 1 - E_{2,n}(A) = p(0) + p(1) + \ldots + p(n-1)\,.$$

The carried traffic Y equals the offered traffic A, as no customers are rejected and the arrival process is a Poisson process:

$$\begin{aligned} Y &= \sum_{i=1}^{n} i \cdot p(i) + \sum_{i=n+1}^{\infty} n \cdot p(i) \qquad (9.7) \\ &= \sum_{i=1}^{n} \frac{\lambda}{\mu} \cdot p(i-1) + \sum_{i=n+1}^{\infty} \frac{\lambda}{\mu} \cdot p(i-1) = \frac{\lambda}{\mu} \cdot \sum_{i=0}^{\infty} p(i)\,, \\ Y &= \frac{\lambda}{\mu} = A\,. \end{aligned}$$

Here we have exploited the cut balance equation between state $[i-1]$ and state $[i]$.

The queue length is a random variable $\mathcal{L}$. The probability of having customers in queue at a random point of time is:

$$\begin{aligned} p\{\mathcal{L} > 0\} &= \sum_{i=n+1}^{\infty} p(i) = p(n) \cdot \frac{\frac{A}{n}}{1 - \frac{A}{n}}\,, \\ p\{\mathcal{L} > 0\} &= \frac{A}{n - A} \cdot p(n) = \frac{A}{n} \cdot E_{2,n}(A)\,, \qquad (9.8) \end{aligned}$$

where we have used (9.5).

9.2.2 Numerical evaluation

Erlang's C-formula (9.6) is similar to Erlang's B-formula (4.10) except for the factor $n/(n-A)$ in the last term. As we have very accurate recursive algorithm for numerical evaluation of Erlang's B-formula (4.29) we use the following relationship for obtaining numerical values of the C-formula:

$$\begin{aligned} E_{2,n}(A) &= \frac{n \cdot E_{1,n}(A)}{n - A\,(1 - E_{1,n}(A))}\,, \qquad A < n \qquad (9.9) \\ &= \frac{E_{1,n}(A)}{1 - y}\,, \end{aligned}$$

where y is the carried traffic per channel in the corresponding loss system (4.13):

$$y = \frac{A\,\{1 - E_n(A)\}}{n} > 0\,.$$

We notice that:

$$E_{1,n}(A) < E_{2,n}(A)\,.$$

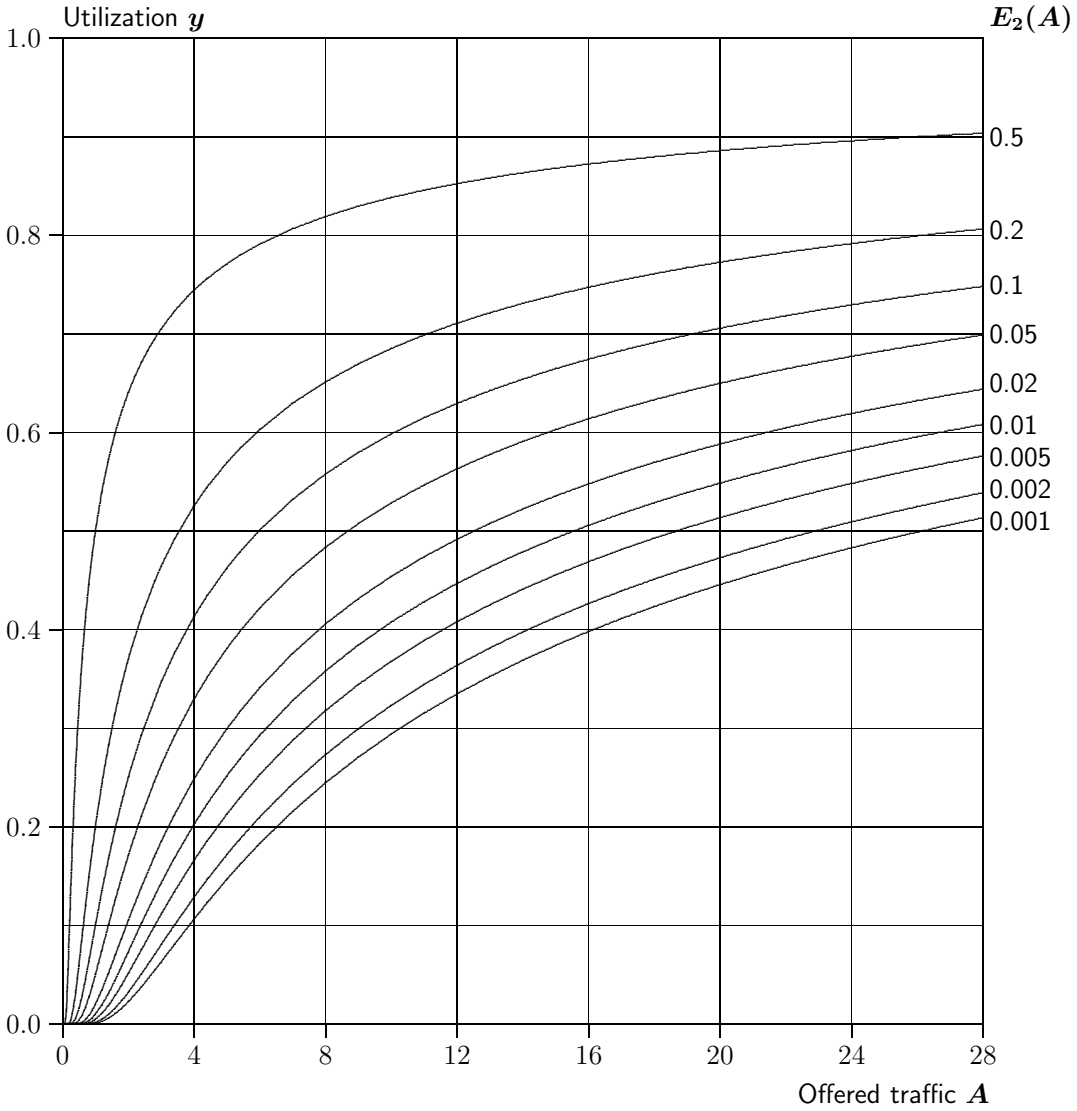


Figure 9.3: *The average utilization per channel y for a fixed probability of delay $E_{2,n}(A)$ as a function of the number of channels n.*

For $A \geq n$, we have $E_{2,n}(A) = 1$ as all customers are delayed.

By using the general approach described in Sec. 4.4.1 we observe from the denominator of (9.6) that the first terms for state [0] to state $[n-1]$ are the same as for Erlang's loss system. The last term which includes all classes from state $[n]$ to ∞ is obtained from state [n-1] by multiplying by

$$\frac{A}{n} \cdot \frac{n}{n-a} = \frac{A}{n-A} \,.$$

So a direct recurrence is obtained by using the recursion for Erlang-B up to state $[n-1]$ and

then find $E_{2,n}(A)$ by the final step:

$$E_{2,n}(A) = \frac{\frac{A}{n-A} \cdot E_{1,n-1}(A)}{1 + \frac{A}{n-A} \cdot E_{1,n-1}(A)},$$

$$E_{2,n}(A) = \frac{A \cdot E_{1,n-1}(A)}{n - A(1 - E_{1,n-1}(A))}. \tag{9.10}$$

Thus we use the same recursion as for Erlang-B formula except for the last step. The two formuæ (9.9) and (9.10) are of course equivalent, but the last one requires one iteration less.

Erlang's C-formula may in an elegant way be expressed by the B-formula as noticed by B. Sanders:

$$\frac{1}{E_{2,n}(A)} = \frac{1}{E_{1,n}(A)} - \frac{1}{E_{1,n-1}(A)}. \tag{9.11}$$

Erlang's C-formula has been tabulated in many books and tables, i.a. in *Moe's Principle* (Jensen, 1950 [59]) and is shown in Fig. 9.2, Fig. 9.3, and Fig. 9.4. We notice that for a given value of $E_{2,n}(A)$, the utilization of each channel increases as number of channels n increases (economy of scale).

9.2.3 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 a random 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 obtain the mean queue length $L_n = E\{\mathcal{L}\}$ at an arbitrary point of time from the state probabilities:

$$L_n = 0 \cdot \sum_{i=0}^{n} p(i) + \sum_{i=n+1}^{\infty} (i-n) \cdot p(i)$$

$$= \sum_{i=n+1}^{\infty} (i-n) \cdot p(n) \left(\frac{A}{n}\right)^{i-n}$$

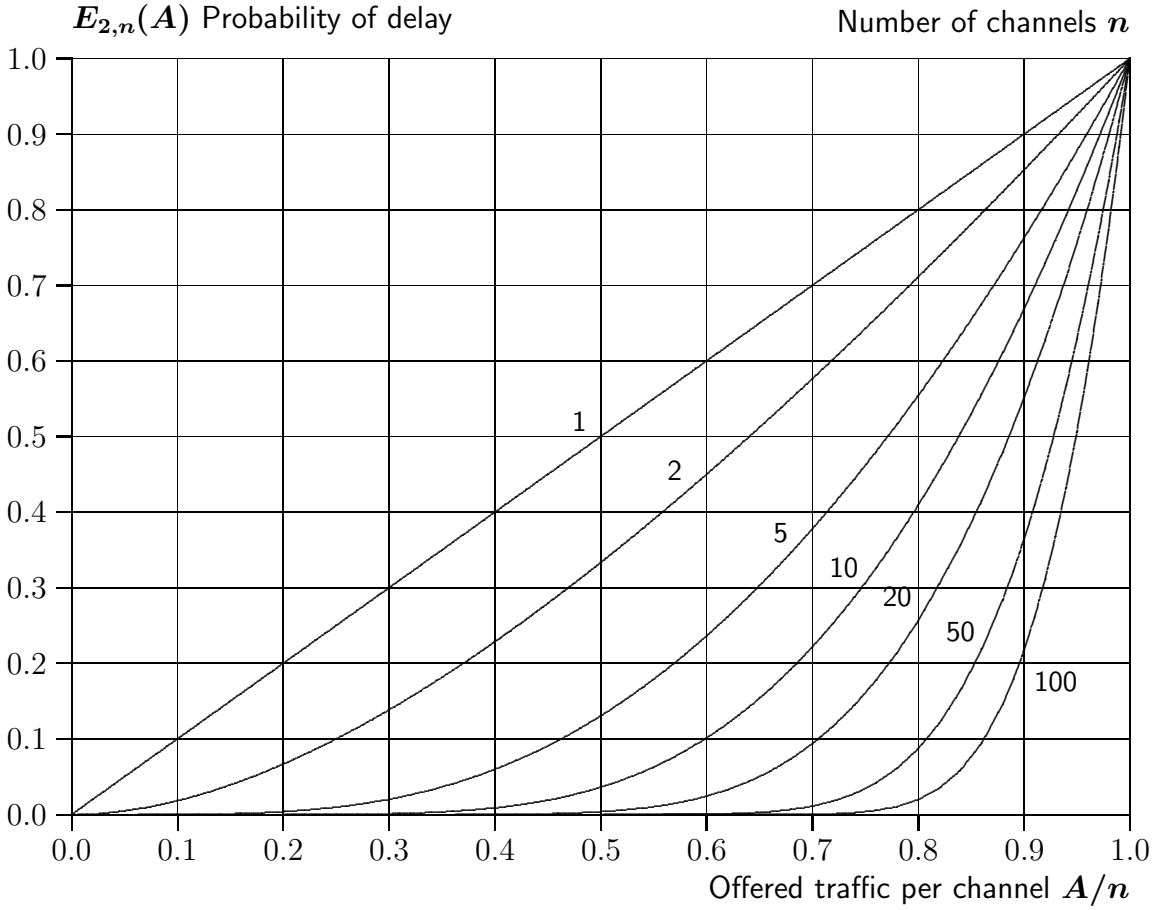


Figure 9.4: *Erlang's C–formula for the delay system M/M/n. The probability $E_{2,n}(A)$ for a positive waiting time is shown as a function of the offered traffic A/n per channel for different values of the number of servers n. This figure is a re-scaling of Fig. 9.2.*

$$L_n = p(n) \cdot \sum_{i=1}^{\infty} i \cdot \left(\frac{A}{n}\right)^i$$

$$= p(n) \cdot \frac{A}{n} \sum_{i=1}^{\infty} \frac{\partial}{\partial(A/n)} \left\{ \left(\frac{A}{n}\right)^i \right\}.$$

As we have $A/n \leq c < 1$, the series is uniformly convergent, and the differentiation operator may be put outside the summation:

$$L_n = p(n) \cdot \frac{A}{n} \cdot \frac{\partial}{\partial(A/n)} \left\{ \frac{A/n}{1-(A/n)} \right\} = p(n) \cdot \frac{A/n}{\{1-(A/n)\}^2}$$

$$= p(n) \cdot \frac{n}{n-A} \cdot \frac{A}{n-A},$$

$$L_n = E_{2,n}(A) \cdot \frac{A}{n-A}. \tag{9.12}$$

The average queue length is 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:

$$\begin{aligned} L_{nq} &= \frac{\sum_{i=n+1}^{\infty}(i-n)\,p(i)}{\sum_{i=n+1}^{\infty}p(i)} \\ &= \frac{p(n)\cdot\frac{n}{n-A}\cdot\frac{A}{n-A}}{p(n)\cdot\frac{A}{n-A}} \\ &= \frac{n}{n-A} \end{aligned} \tag{9.13}$$

By applying (9.8) and (9.12), 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.

9.2.4 Mean waiting times

Also here 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:

$$L_n = \lambda\cdot W_n\,, \tag{9.14}$$

where $L_n = L_n(A)$, and $W_n = W_n(A)$. Inserting L_n from (9.12) we get:

$$W_n = \frac{L_n}{\lambda} = \frac{1}{\lambda}\cdot E_{2,n}(A)\cdot\frac{A}{n-A}\,.$$

As $A = \lambda\, s$, where $s = 1/\mu$ is the mean service time, we get:

$$W_n = E_{2,n}(A) \cdot \frac{s}{n - A}\,. \tag{9.15}$$

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 a positive waiting time w_n (2.30):

$$W_n = w_n \cdot E_{2,n}(A)\,, \tag{9.16}$$

$$w_n = \frac{s}{n - A}\,. \tag{9.17}$$

Example 9.2.1: Mean waiting time w when $A \to 0$
Notice, that as $A \to 0$, we get $w_n = s/n$ (9.17). If a customer experiences waiting time (which seldom happens when $A \to 0$), 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. □

9.2.5 Improvement functions for $M/M/n$

The marginal improvement 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:

$$F_{2,n}(A) = A\,\{E_{2,n}(A) - E_{2,n+1}(A)\}\,. \tag{9.18}$$

- The decrease in mean queue length (traffic carried by the waiting positions) becomes:

$$F_{L,n}(A) = L_n(A) - L_{n+1}(A)\,. \tag{9.19}$$

- The decrease in mean waiting time $W_n(A)$ for all customers:

$$\begin{aligned} F_{W,n}(A) &= W_n(A) - W_{n+1}(A) \\ &= \frac{1}{\lambda} \cdot F_{L,n}(A)\,, \end{aligned} \tag{9.20}$$

  where we have used Little's law (9.14). If we choose the mean service time as time unit, then $\lambda = A$. We consider $W_n(A)$ below.

Both (9.18) and (9.19) are tabulated in *Moe's Principle* (Jensen, 1950 [59]) and are simple to evaluate by a calculator or computer.

9.3 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 by all k systems has the total average waiting time $W = \sum_i W_i$, 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:

$$C = C_0 + \sum_{i=1}^{k} n_i \, c_i \, .$$

If the waiting time also is considered as a cost, then the total costs to be minimized becomes $f = f(n_1, n_2, \ldots, n_k)$. This is to be minimized as a function of number of channels n_i in the individual systems. The allocation of channels to the individual systems is determined by:

$$\min \left\{ f\big(n_1, n_2, \ldots, n_k\big) \right\} = \min \left\{ C_0 + \sum_i n_i \, c_i + \vartheta \cdot \Big(\sum_i W_i - W \Big) \right\} . \tag{9.21}$$

where ϑ (theta) is Lagrange's multiplier (shadow prices).

As n_i are integers, a necessary condition for minimum, which in this case can be shown also to be a sufficient condition, becomes:

$$\begin{aligned} 0 \;&<\; f(n_1, n_2, \ldots, n_i - 1, \ldots, n_k) - f(n_1, n_2, \ldots, n_i, \ldots, n_k) \; , \\ 0 \;&\geq\; f(n_1, n_2, \ldots, n_i, \ldots, n_k) - f(n_1, n_2, \ldots, n_i + 1, \ldots, n_k) \; , \end{aligned} \tag{9.22}$$

which corresponds to:

$$\begin{aligned} W_{n_i - 1}\,(A_i) - W_{n_i}\,(A_i) \;&>\; \frac{c_i}{\vartheta} \, , \\ W_{n_i}\,(A_i) - W_{n_i + 1}\,(A_i) \;&\leq\; \frac{c_i}{\vartheta} \, , \end{aligned} \tag{9.23}$$

where $W_{n_i}(A_i)$ is given by (9.15).

Expressed by the improvement function for the waiting time $F_{W,n}(A)$ (9.20) the optimal solution becomes:

$$F_{W,n_i - 1}(A) > \frac{c_i}{\vartheta} \geq F_{W,n_i}(A) \, , \qquad i = 1, 2, \ldots k \, . \tag{9.24}$$

The function $F_{W,n}(A)$ is tabulated in *Moe's Principle* (Jensen, 1950 [59]). Similar optimizations can be carried out for other improvement functions.

Example 9.3.1: 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 ratio equals $c_2/\vartheta = 0.1$. A table of the improvement function $F_{W,n}(A)$ gives:

$$n_1 = 32 \text{ channels and}$$

$$n_2 = 5 \text{ channels.}$$

The mean waiting times are:

$$W_1 = 0.075 \text{ s.}$$

$$W_2 = 0.199 \text{ s.}$$

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 (capacity). We should go on investing more 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.

9.4 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 among 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 (9.6).

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 (3.4): The probability $p\{\mathcal{W} \le t\} = F(t)$ 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 depart during the interval t (3.21):

$$F(t \,|\, k) = \sum_{i=k+1}^{\infty} \frac{(n\mu t)^i}{i!} \cdot \mathrm{e}^{-n\mu t}\,. \tag{9.25}$$

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:

$$\begin{aligned} p_w(k) &= \frac{\lambda \cdot p(n+k)}{\lambda \cdot \sum_{i=0}^{\infty} p(n+i)} = \frac{p(n) \cdot \left(\frac{A}{n}\right)^k}{p(n) \cdot \sum_{i=0}^{\infty} \left(\frac{A}{n}\right)^i} \\ &= \left(1 - \frac{A}{n}\right) \left(\frac{A}{n}\right)^k, \qquad k = 0, 1, \ldots\,. \end{aligned} \tag{9.26}$$

This is a geometric distribution including the zero class (Tab. 3.1). The unconditional waiting time distribution then becomes:

$$F(t) = \sum_{k=0}^{\infty} p_w(k) \cdot F(t \,|\, k)\,, \tag{9.27}$$

$$\begin{aligned} F(t) &= \sum_{k=0}^{\infty} \left\{ \left(1 - \frac{A}{n}\right) \left(\frac{A}{n}\right)^k \cdot \sum_{i=k+1}^{\infty} \frac{(n\mu t)^i}{i!}\, \mathrm{e}^{-n\mu t} \right\} \\ &= \mathrm{e}^{-n\mu t} \sum_{i=1}^{\infty} \left\{ \frac{(n\mu t)^i}{i!} \cdot \sum_{k=0}^{i-1} \left(1 - \frac{A}{n}\right) \left(\frac{A}{n}\right)^k \right\}, \end{aligned}$$

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:

$$\begin{aligned}F(t) &= \mathrm{e}^{-n\mu t}\cdot\sum_{i=1}^{\infty}\frac{(n\mu t)^i}{i!}\left\{1-\left(\frac{A}{n}\right)^i\right\} \\ &= \mathrm{e}^{-n\mu t}\left\{\sum_{i=0}^{\infty}\frac{(n\mu t)^i}{i!}-\sum_{i=0}^{\infty}\frac{(n\mu t)^i}{i!}\left(\frac{A}{n}\right)^i\right\} \\ &= \mathrm{e}^{-n\mu t}\left\{\mathrm{e}^{n\mu t}-\mathrm{e}^{n\mu t\cdot A/n}\right\}, \\ F(t) &= 1-\mathrm{e}^{-(n-A)\mu t}, \\ F(t) &= 1-\mathrm{e}^{-(n\mu-\lambda)t}, \qquad n>A, \quad t>0. \end{aligned} \tag{9.28}$$

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. 9.6 the phase-diagram for (9.27) is shown, and we notice immediately that it can be reduced to a single exponential distribution (Sec. 2.4.4 & Fig. 2.12). Formula (9.28) confirms that the mean waiting time w_n for customers who have to wait in the queue becomes as shown in (9.17).

The waiting time distribution for all (an arbitrary customer) becomes (2.29):

$$F_s(t) = 1 - E_{2,n}(A)\cdot\mathrm{e}^{-(n-A)\mu t}, \quad A<n, \quad t\geq 0, \tag{9.29}$$

and the mean value of this distribution is W_n in agreement with (9.15). The results may be derived in an easier way by means of generation functions.

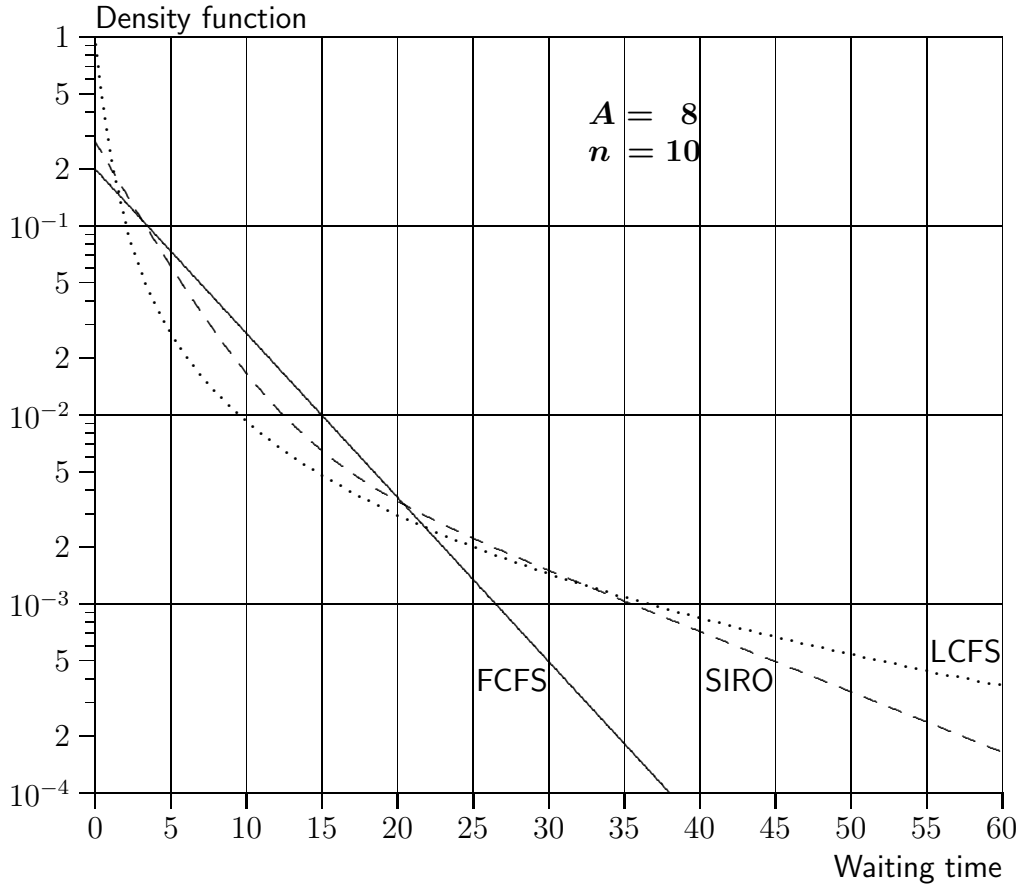


Figure 9.5: *Density function for the waiting time distribution for the queueing discipline FCFS, LCFS, and SIRO (RANDOM). For all three cases the mean waiting time for delayed calls is 5 time-units. The form factor is 2 for FCFS, 3.33 for LCFS, and 10 for SIRO. The number of servers is 10 and the offered traffic is 8 erlang. The mean service time is* $s = 10$ *time-units.*

9.5 Single server queueing system M/M/1

This is the system appearing most often in the literature. The state probabilities (9.2) are given by a geometric series:

$$p(i) = (1 - A) \cdot A^i, \qquad i = 0, 1, 2, \ldots, \tag{9.30}$$

as $p(0) = 1 - A$. The mean value of state probabilities is $m_1 = A/(1 - A)$.

The probability of delay become:

$$E_{2,1}(A) = A\,.$$

The mean queue length L_n (9.12) and the mean waiting time for all customers W_n (9.15)

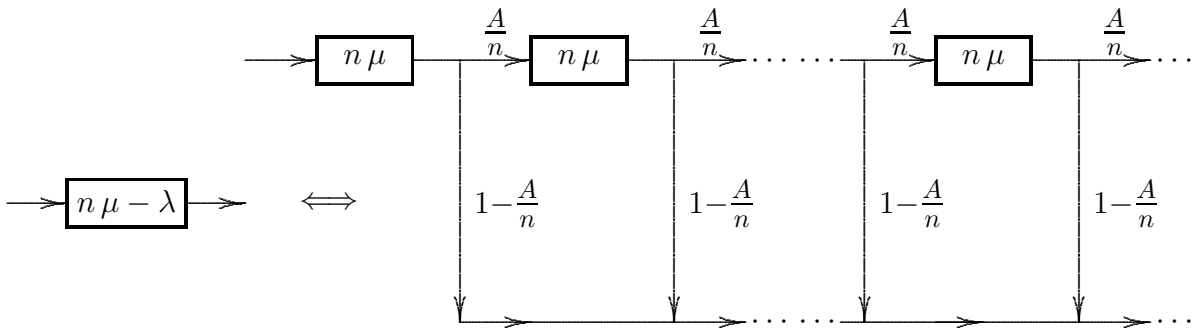


Figure 9.6: *The waiting time distribution for M/M/n–FCFS becomes exponentially distributed with intensity* $(n\mu - \lambda)$. *The phase-diagram to the left corresponds to a weighted sum of Erlang-k distributions (Sec. 2.4.4) as the termination rate out of all phases is* $n\mu \cdot (1 - \frac{A}{n}) = n\mu - \lambda$.

become:

$$L_1 = \frac{A^2}{1-A}, \tag{9.31}$$

$$W_1 = \frac{A\,s}{1-A}. \tag{9.32}$$

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.

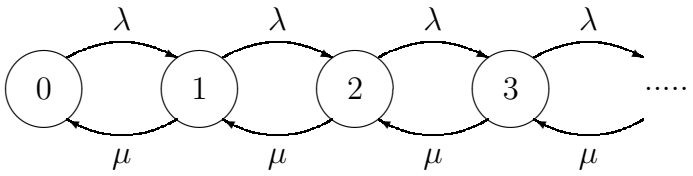


Figure 9.7: *State transition diagram for M/M/1.*

9.5.1 Sojourn time for a single server

When there is only one server, the state probabilities (9.2) are given by a geometric series (9.30) 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 *sojourn* time in the system (waiting time + service time), which also is called the *response time*, is exponentially distributed with intensity $(\mu - \lambda)$ (cf. Fig. 2.12):

$$F(t) = 1 - \mathrm{e}^{-(\mu - \lambda)t}\,, \quad \mu > \lambda\,, \quad t \geq 0\,. \tag{9.33}$$

This is identical with the waiting time distribution of delayed customers. The mean sojourn time may be obtained directly using W_1 from (9.32) and the mean service time s:

$$\begin{aligned} m_1 &= W_1 + s = \frac{A\,s}{1-A} + s\,, \\ m_1 &= \frac{s}{1-A} = \frac{1}{\mu - \lambda}\,, \end{aligned} \tag{9.34}$$

where $\mu = 1/s$ is the service rate. We notice that mean sojourn time is equal to mean waiting time for delayed customers (9.17). The mean sojourn time is by Little's law also equal to the mean value of state probabilities divided by λ.

9.6 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 the Russian Gnedenko in 1933 and published in 1934. It became widely known when C. Palm published a paper in 1947 [95] in connection with a theoretical analysis of manpower allocation for servicing automatic machines. A number of 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 minimized (or the profit optimized). 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 by Feller (1950 [32]). The model corresponds to a simple closed queueing network and is successfully applied to solve traffic engineering problems in computer systems. By using Kendall's notation (Sec. 10.1) 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 correspond to *clients* whereas the repairmen correspond to servers. In computer terminal systems the machines correspond to terminals and a repairman corresponds to a computer managing the terminals. In a

computer system the machines may correspond to disc storages and the repairmen correspond to input/output (I/O) channels. In the following we will consider a computer terminal system as the background for development of the theory.

9.6.1 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. 9.8).

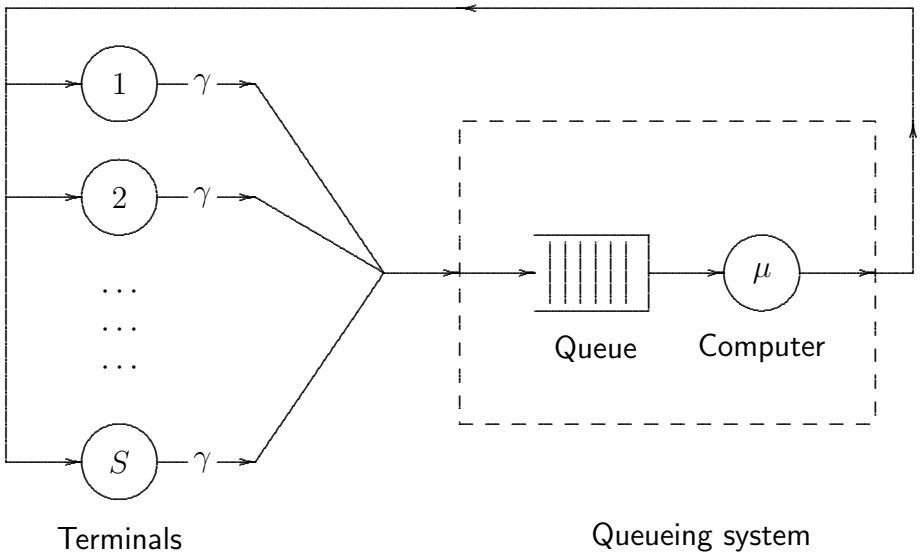


Figure 9.8: *Palm's machine-repair model. A computer system with S terminals (an interactive system) corresponds to a waiting time system with a limited number of sources.*

The individual terminal all the time changes between two states (interact) (Fig. 9.9):

- 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. 9.9). 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. 10.2.1).

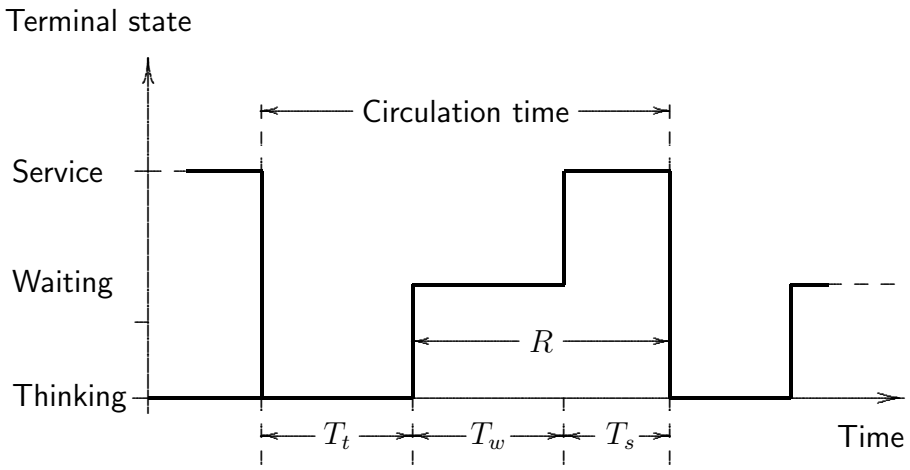


Figure 9.9: *The individual terminal may be in three different states. Either the user is active working at the terminal (thinking), or he is waiting for response from the computer. The latter time interval (response time) is divided into two phases: a waiting phase and a service phase.*

9.6.2 State probabilities – single server

We consider now a system with S terminals, which are connected to one computer ($n = 1$). The thinking time for each thinking terminal is so far assumed to be exponentially distributed with intensity $\gamma = 1/m_t$, and the service (execution) time 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 in queue have arrival intensity zero.

State $[\,i\,]$ is defined as the state, where there are i terminals in the queueing system (Fig. 9.8), 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 modeled by a pure birth & death process, and the state transition diagram is shown in Fig. 9.10. 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.

The steady state probabilities are found by applying cut equations to Fig. 9.10 and expressing all states in terms of state S:

$$(S - i)\,\gamma \cdot p(i) = \mu \cdot p(i + 1), \quad i = 0, 1, \ldots, S - 1\,. \tag{9.35}$$

By the additional normalization constraint requiring that the sum of all probabilities must

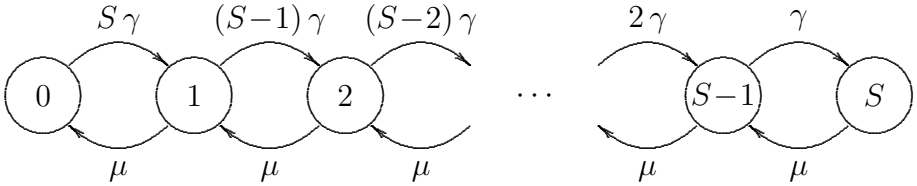


Figure 9.10: *State transition diagram for the queueing system shown in 9.8. State [i] denotes the number of terminals being either served or waiting, i.e. $S - i$ denotes the number of terminals thinking.*

be equal to one we find, introducing $\varrho = \mu/\gamma$:

$$p(S-i) = \frac{\varrho^i}{i!}\,p(S)$$

$$= \frac{\dfrac{\varrho^i}{i!}}{\displaystyle\sum_{j=0}^{S}\frac{\varrho^j}{j!}}, \quad i = 0, 1, \ldots, S\,, \tag{9.36}$$

$$p(0) = E_{1,S}\,(\varrho)\ . \tag{9.37}$$

This is the *truncated Poisson distribution* (4.9).

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 (9.37). Erlang's B-formula is valid for arbitrary holding times (Sec. 4.6.2) and therefore we have:

Theorem 9.1 *The state probabilities of the machine repair model (9.36)(9.37) with one computer and S terminals is valid for arbitrary thinking time distributions when the service time of the computer are exponentially distributed. Only the mean thinking time is of importance.*

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 (9.36) & (9.37) is of course as for Erlang's B-formula (4.29).

Example 9.6.1: Information system
We consider an information system which is organized 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 time required for reading 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. We can never get a higher utilization for this system. □

9.6.3 Terminal states and traffic characteristics

The performance measures are easily obtained from the analogy with Erlang's classical loss system (9.37). The computer is working with probability $1 - p(0) = \{1 - E_{1,S}(\varrho)\}$. Thus the average number of terminals being served by the computer (utilization of computer) is given by:

$$n_s = 1 - E_{1,S}(\varrho)\,. \tag{9.38}$$

The average number of thinking terminals corresponds to the traffic carried in Erlang's loss system:

$$n_t = \frac{\mu}{\gamma}\,\{1 - E_{1,S}(\varrho)\} = \varrho\,\{1 - E_{1,S}(\varrho)\}. \tag{9.39}$$

The average number of waiting terminals becomes:

$$\begin{aligned} n_w &= S - n_s - n_t && (9.40)\\ &= S - \{1 - E_{1,S}(\varrho)\} - \varrho \cdot \{1 - E_{1,S}(\varrho)\} \\ &= S - \{1 - E_{1,S}(\varrho)\}\{1 + \varrho\}\,. && (9.41) \end{aligned}$$

If we consider a random terminal at a random point of time, we get:

$$p\{\text{terminal served}\} = p_s = \frac{n_s}{S} = \frac{1 - E_{1,S}(\varrho)}{S}\,, \tag{9.42}$$

$$p\{\text{terminal thinking}\} = p_t = \frac{n_t}{S} = \frac{\varrho\,(1 - E_{1,S}(\varrho))}{S}\,, \tag{9.43}$$

$$p\{\text{terminal waiting}\} = p_w = \frac{n_w}{S} = 1 - \frac{\{1 - E_{1,S}(\varrho)\}\{1 + \varrho\}}{S}\,. \tag{9.44}$$

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):

$$\frac{1}{\lambda} = \frac{m_t}{n_t} = \frac{m_w}{n_w} = \frac{m_s}{n_s} = \frac{m_r}{n_w + n_s}\,, \tag{9.45}$$

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 (9.45) and (9.38) $\left\{\dfrac{n_t}{n_s} = \dfrac{m_t}{m_s}\right\}$ we get:

$$m_r = \frac{S}{n_s} \cdot m_s - m_t$$

$$m_r = \frac{S}{1 - E_{1,S}(\varrho)} \cdot m_s - m_t\,. \tag{9.46}$$

Thus the mean response time is insensitive to the time distributions as it is based on (9.38) and (9.45) (Little's Law). However, $E_{1,S}(\varrho)$ 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 $E_{1,S}(\varrho)$ will be given by (9.37). Fig. 9.11 shows the response time as a function of the number of terminals in this case.

If all time intervals are constant and synchronized, the computer may work all the time serving K terminals without any delay when:

$$K = \frac{m_t + m_s}{m_s} \tag{9.47}$$

$$= \varrho + 1\,. \tag{9.48}$$

A number of terminals equal to 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 (9.46):

$$m_w = m_r - m_s$$

Example 9.6.2: 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 utilization of the computer, whereas many terminals connected will waste the time of the users.
Fig. 9.12 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.

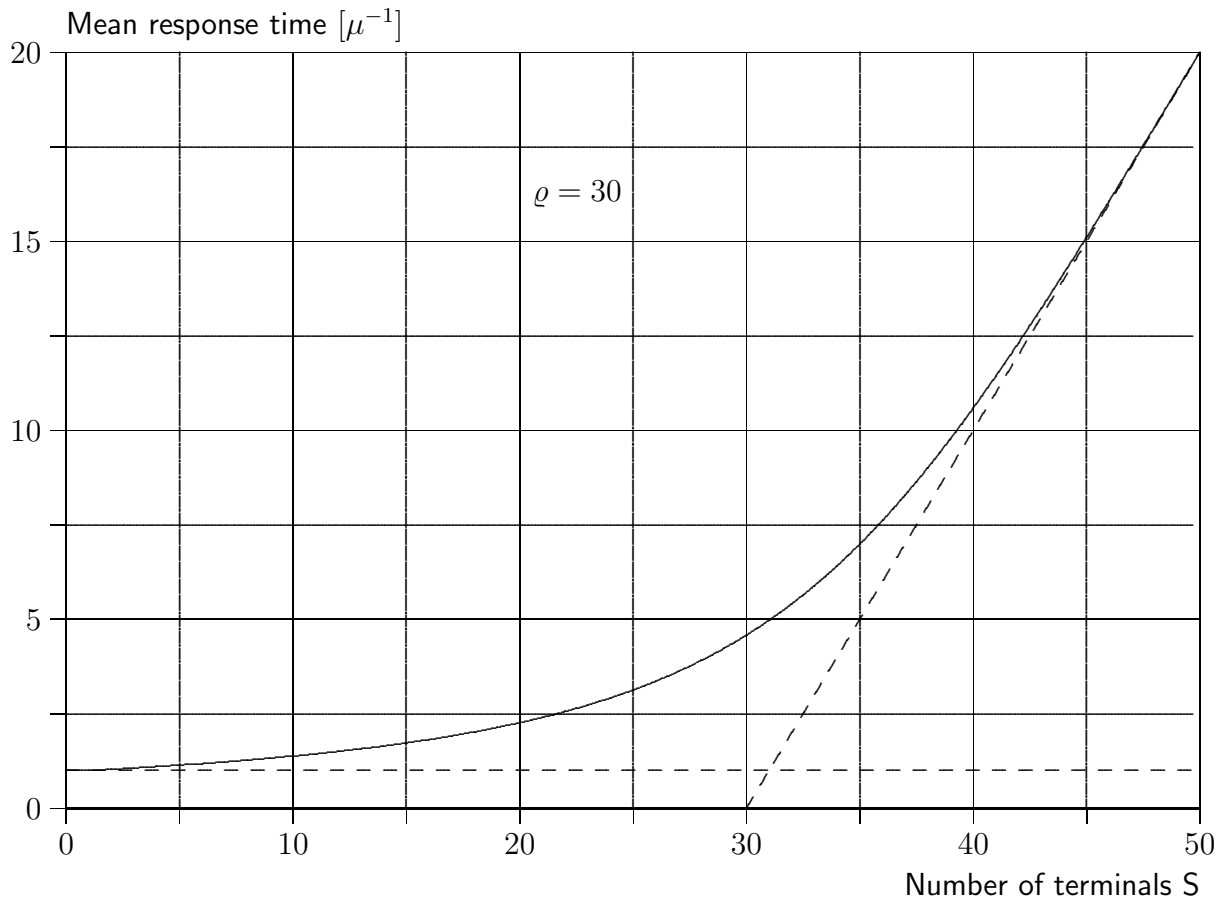


Figure 9.11: *The actual average response time experienced by a terminal as a function of the number of terminals. The service-ratio is $\varrho = 30$. The average response time converges to a straight line, cutting the x-axes in $S = 30$ terminals. The average virtual response time for a system with S terminals is equal to the actual average response time for a system with $S+1$ terminals (the Arrival theorem, theorem 5.1).*

For the example in Fig. 9.12 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 9.6.3: Traffic congestion
We may define the traffic congestion in the usual way (Sec. 1.9). The offered traffic is the traffic carried when there is no queue. The offered traffic per source is (5.10):

$$a = \frac{\beta}{1+\beta} = \frac{m_s}{m_t + m_s}$$

The carried traffic per source is:

$$y = \frac{m_s}{m_t + m_w + m_s}\,.$$

The traffic congestion becomes:

$$\begin{aligned} C &= \frac{a-y}{a} \\ &= 1 - \frac{m_t + m_s}{m_t + m_w + m_s} = \frac{m_w}{m_t + m_w + m_s}\,, \\ C &= p_w \end{aligned}$$

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. □

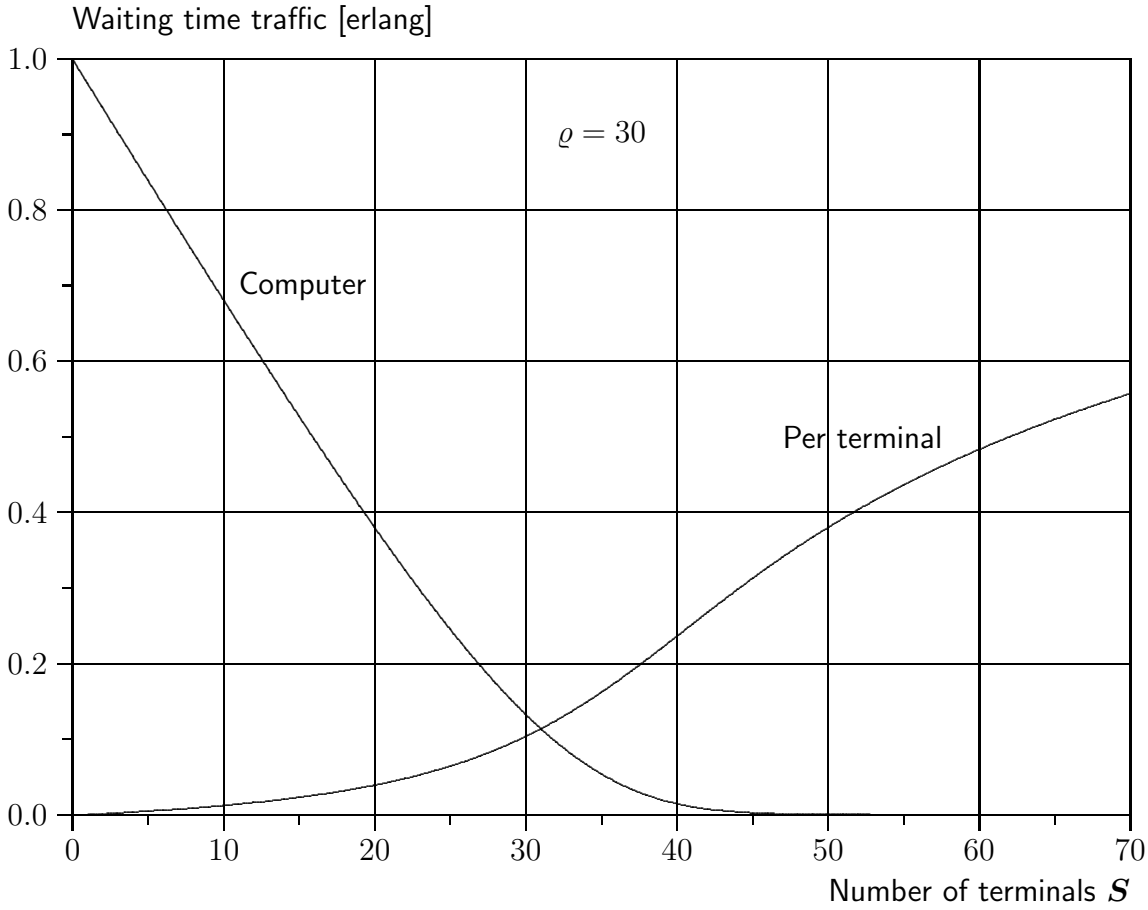


Figure 9.12: *The waiting time traffic (the proportion of time spent waiting) measured in [erlang] for the computer, respectively the terminals in an interactive queueing system (Service ratio $\varrho = 30$).*

9.6.4 Machine–repair model with n servers

The above model is easily generalized to n computers. The transition diagram is shown in Fig. 9.13.

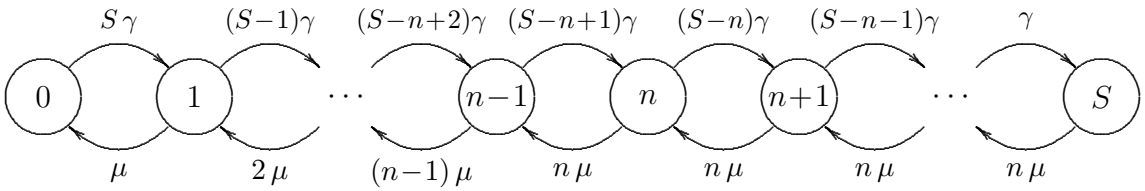


Figure 9.13: *State transition diagram for the machine-repair model with S terminals and n computers.*

The steady state probabilities become:

$$\begin{aligned} p(i) &= \binom{S}{i}\left(\frac{\gamma}{\mu}\right)^i p(0)\,, && 0 \le i \le n\,, \\ p(i) &= \frac{(S-n)!}{(S-i)!}\left(\frac{\gamma}{n\mu}\right)^{i-n}\cdot p(n)\,, && n \le i \le S\,. \end{aligned} \tag{9.49}$$

where we have the normalization constraint:

$$\sum_{i=0}^{S} p(i) = 1\,. \tag{9.50}$$

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:

$$p_s = \frac{1}{S}\left\{\sum_{i=0}^{n} i\cdot p(i) + \sum_{i=n+1}^{S} n\cdot p(i)\right\}\,, \tag{9.51}$$

$$p_t = p_s\cdot\frac{\mu}{\gamma}\,, \tag{9.52}$$

$$p_w = 1 - p_s - p_t\,. \tag{9.53}$$

The mean utilization of the computers becomes:

$$\alpha = \frac{p_s}{n} \cdot S = \frac{n_s}{n} \,. \tag{9.54}$$

The mean waiting time for a terminal becomes:

$$W = \frac{p_w}{p_s} \cdot \frac{1}{\mu} \,. \tag{9.55}$$

Sometimes p_w is called *the loss coefficient of the terminals*, and similarly $(1-\alpha)$ is called *the loss coefficient of the computers* (Fig. 9.12).

Example 9.6.4: Numerical example of scale of economy
The following numerical examples illustrate that we obtain the highest utilization 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 $p_t = \alpha$).

$\boldsymbol{n}$	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

□

9.7 Optimizing 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 optimized in Chap. 4. We will thus see that the same model can be optimized 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:

c_t = *cost per terminal per time unit a terminal is thinking*,

c_w = *cost per terminal per time unit a terminal is waiting*,

c_s = *cost per terminal per time unit a terminal is served*,

c_a = cost of the computer per time unit.

The cost of the computer is supposed to be independent of the utilization and is split uniformly among all terminals.

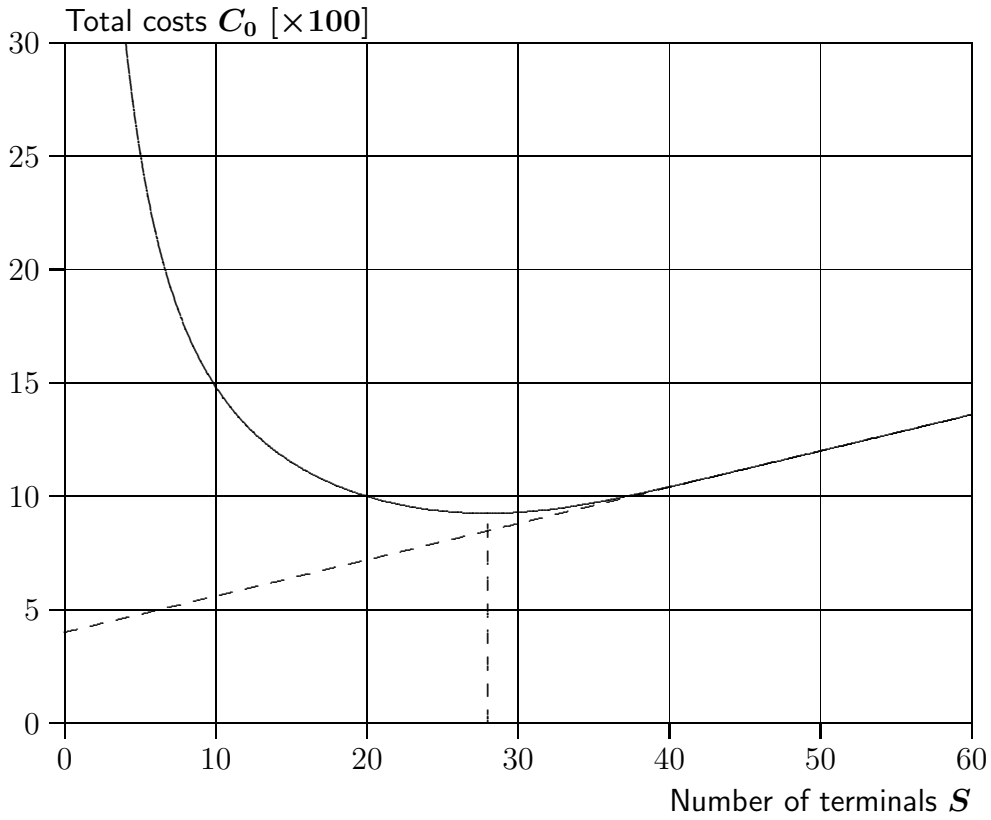


Figure 9.14: *The machine/repair model. The total costs given in (9.59) are shown as a function of number of terminals for a service ratio* $\varrho = 25$ *and a cost ratio* $r = 1/25$ *(cf. Fig. 4.7).*

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:

$$p_t \cdot c_0 = p_t \cdot c_t + p_s \cdot c_s + p_w \cdot c_w + \frac{1}{S} \cdot c_a \,. \tag{9.56}$$

We want to minimize c_0. The service ratio $\varrho = m_t/m_s$ is equal to p_t/p_s. Introducing the cost ratio $r = c_w/c_a$, we get:

$$\begin{aligned} c_0 &= c_t + \frac{p_s}{p_t} \cdot c_s + \frac{p_w \cdot c_w + \frac{1}{S} \cdot c_a}{p_t} \\ &= c_t + \frac{1}{\varrho} \cdot c_s + c_a \cdot \frac{r \cdot p_w + (1/S)}{p_t}, \end{aligned} \tag{9.57}$$

which is to be minimized as a function of S. Only the last term depends on the number of terminals and we get:

$$\begin{aligned}
\min_S \{c_0\} &= \min_S \left\{ \frac{r \cdot p_w + (1/S)}{p_t} \right\} \\
&= \min_S \left\{ \frac{r \cdot (n_w/S) + (1/S)}{n_t/S} \right\} \\
&= \min_S \left\{ \frac{r \cdot n_w + 1}{n_t} \right\} \qquad (9.58) \\
&= \min_S \left\{ \frac{r \left[S - \{1 - E_{1,S}(\varrho)\} \{1 + \varrho\} \right] + 1}{\{1 - E_{1,S}(\varrho)\} \cdot \varrho} \right\} \\
&= \min_S \left\{ \frac{r \cdot S + 1}{\{1 - E_{1,S}(\varrho)\} \cdot \varrho} + 1 + \frac{1}{\varrho} \right\} , \qquad (9.59)
\end{aligned}$$

where $E_{1,S}(\varrho)$ is Erlang's B-formula (9.36).

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 (4.50), whereas the denominator corresponds to the carried traffic in the corresponding loss system. Thus we minimize the cost per carried erlang in the corresponding loss system. In Fig. 9.14 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. 4.7), where we optimize the profit.

9.8 Waiting time distribution for M/M/n/S/S–FCFS

We consider a finite-source system with S sources and n channels. Both thinking time and service time are assumed to be exponential distributed with rate γ, respectively μ. Due to the arrival theorem an arriving call observes the the state probabilities of a system with $S-1$ sources. We renumber the states so that the state is defined as number of thinking sources. We denote the state probabilities of a system with $S-1$ sources by:

$$p_{S-1}(i) , \quad i = 0, 1, \ldots, S-1 . \qquad (9.60)$$

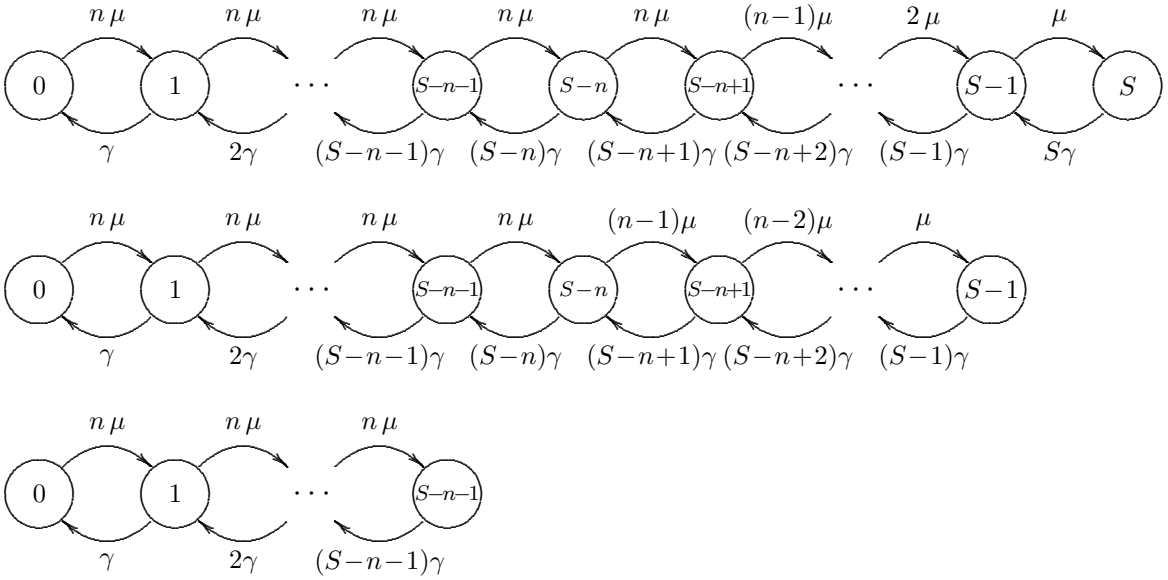


Figure 9.15: *The upper part shows the state transition diagram for the machine-repair model with S terminals and n computers. The state of the system is defined as the number of thinking customers (cf. Fig. 9.13). The middle diagram shows the same model with $S-1$ sources, i.e. the states seen by an arriving customer according to the arrival theorem. The lower part shows a subset of the states from state [0] to state [$S-n$] which corresponds to the diagram for an Erlang loss system with $S-n$ channels.*

The probability of delay p_w, respectively the probability of immediate service p_x ($p_w+p_x=1$), becomes:

$$p_w = \sum_{i=0}^{S-n-1} p_{S-1}(i)\,. \tag{9.61}$$

$$p_x = \sum_{i=S-n}^{S-1} p_{S-1}(i)\,. \tag{9.62}$$

We consider only delayed calls, i.e. an arriving call observes a system with $S-1$ sources and will be delayed if he observe one of the states $\{0,1,\ldots,S-n-1\}$ (9.61) where all servers are occupied. This part of the state transition diagram corresponds to an Erlang loss system with arrival rate $n\mu$, service rate γ, i.e. an offered traffic $A=n\mu/\gamma$, and $S-n-1$ servers. These probabilities may be calculated accurately as described in Sec. 4.4. Thus these conditional state probabilities are given by the truncated Poisson distribution (4.9):

$$p_{S-1,w}(i) = \frac{\dfrac{A^i}{i\,!}}{1+A+\dfrac{A^2}{2\,!}+\ldots+\dfrac{A^{S-1-n}}{(S-1-n)\,!}}\,, \quad i=0,1,\ldots,S-1-n\,, \tag{9.63}$$

where $A=n\mu/\gamma$. The state probabilities (9.63) are a subset of (9.60). In state [0] no customers arrive as they all are waiting or being served. In state [1] all servers are busy and

$S-n-2$ customers are waiting. Thus the waiting time will be Erlang-$(S-n-1)$ distributed. In state $[S-n-1]$ all servers are busy but no one is waiting, so the waiting time becomes Erlang-1 distributed. In general in state i $(0 \leq i \leq S-1-n)$ the waiting time becomes Erlang-$(S-n-i)$ distributed.

The Erlang-k distribution with intensity $n\mu$ is (3.21):

$$\begin{aligned} F_k(t) &= \int_{x=0}^{t} \frac{(n\mu x)^{k-1}}{(k-1)!}\, n\mu \cdot \mathrm{e}^{-n\mu x}\, \mathrm{d}x \\ &= \sum_{j=k}^{\infty} \frac{(n\mu t)^j}{j!} \cdot \mathrm{e}^{-n\mu t} \\ &= 1 - \sum_{j=0}^{k-1} \frac{(n\mu t)^j}{j!} \cdot \mathrm{e}^{-n\mu t}\,. \end{aligned} \tag{9.64}$$

Thus for a given value t we can calculate the distribution function $F(t)$ by calculating the first k terms $(0, 1, \ldots, k-1)$ of a Poisson distribution with parameter $n\mu t$. For small mean values this can be done directly. For large mean values this can be done in a numerical stable way as for example shown in Example 4.4.1. The mean value of this Erlang-k distribution is $k/(n\,\mu)$.

The compound waiting time distribution for delayed customers is obtained by summation over all states:

$$F_w(t) = \sum_{i=0}^{S-1-n} p_{S-1,w}(i) \cdot F_{S-n-i}(t)\,, \tag{9.65}$$

where $p_{i,w}(t)$ is given by (9.63) and $F_k(t)$ is given by (9.64). Both of these can be calculated accurately, and thus the waiting time distribution is obtained by a finite number of terms.

The mean waiting time w for a delayed customer becomes:

$$\begin{aligned} w &= \sum_{i=0}^{S-n-1} p_{S-1,w}(i) \cdot \frac{S-n-i}{n\,\mu} \\ &= \frac{S-n}{n\,\mu} - \frac{1}{n\,\mu} \cdot \sum_{i=0}^{S-n-1} p_{S-1,w}(i) \cdot i\,, \\ w &= \frac{(S-n)-Y}{n\,\mu}\,, \end{aligned} \tag{9.66}$$

where Y is the traffic carried in the above Erlang loss system (9.63) with $S-n-1$ servers. The mean waiting time for all customers then becomes:

$$W = p_w \cdot w\,, \tag{9.67}$$

where p_w is given above (9.61).

Example 9.8.1: Mean waiting times for $(n, S, A) = (2, 60, 60)$
We consider a system with $n = 2$ servers, $S = 60$ sources, and $A = 60$ erlang. We choose mean service time as time unit, $1/\mu = 1$. Thus $1/\gamma = 30$ [time units]. From (9.66) we get:

$$\begin{aligned} w &= \frac{(60-2) - 60 \cdot (1 - E_{57}(60))}{2} \\ &= \frac{(60-2) - 60 \cdot (1 - 0.128376)}{2}\,, \\ w &= 2.851280 \quad \text{[mean service times]}\,. \end{aligned}$$

An arriving customer is either delayed or served immediately. Above we considered states $(0, 1, \ldots, 57)$, where a customer is delayed. These state probabilities add to one, when $p(57) = E_{57}(60) = 0.128376$. We now find states $p(58)$ and $p(59)$ expressed by state probability $p(57)$:

$$\begin{aligned} p(58) &= p(57) \cdot \frac{60}{58} = 0.132803\,, \\ p(59) &= p(58) \cdot \frac{30}{59} = 0.067527\,. \end{aligned}$$

Thus the state probabilities now add to 1.200330, and the normalized probabilities of delay before service, respectively immediate service becomes:

$$\begin{aligned} p_w &= \frac{1}{1.200330} = 0.833105\,, \\ p_x &= \frac{0.200330}{1.200330} = 0.166895\,. \end{aligned}$$

The the mean waiting time for all customers then becomes (9.67):

$$W = 2.375414 \quad \text{[time units]}\,,$$

which is in agreement with Example 9.6.4. The circulation time becomes

$$\begin{aligned} t_c &= \textit{idle time} + \textit{waiting time} + \textit{service time} \\ &= 30 + 2.375414 + 1 = 33.375414 \quad \text{[time units]}\,, \end{aligned}$$

and the state probabilities (time averages) (p_s, p_w, p_t) becomes as given in Example 9.6.4. □

Example 9.8.2: Mean waiting times for $(n, S, A) = (2, 10, 10)$
We consider a system with $n = 2$ servers, $S = 10$ sources, and $A = 10$ erlang. We choose mean service time as time unit, $1/\mu = 1$. Thus $1/\gamma = 5$ [time units]. From (9.66) we get:

$$\begin{aligned} w &= \frac{(10-2) - 10 \cdot (1 - E_7(10))}{2} \\ &= \frac{(10-2) - 10 \cdot (1 - 0.409041)}{2}\,, \\ w &= 1.045205 \quad \text{[mean service times]}\,. \end{aligned}$$

An arriving customer is either delayed or served immediately. Above we considered states $(0, 1, \ldots, 7)$, where a customer is delayed. These state probabilities add to one when $p(7) = E_7(10) = 0.409041$. We now find states $p(8)$ and $p(9)$ expressed by state probability $p(7)$:

$$p(8) = p(7) \cdot \frac{10}{8} = 0.511301\,,$$

$$p(9) = p(8) \cdot \frac{5}{9} = 0.284056\,.$$

Thus the state probabilities now add to 1.795358, and the normalized probabilities of delay before service, respectively immediate service becomes:

$$p_w = \frac{1}{1.795358} = 0.556992\,,$$

$$p_x = \frac{0.79536}{1.79536} = 0.443008\,.$$

The the mean waiting time for all customers then becomes (9.67):

$$W = 0.582171 \quad [\text{time units}]\,.$$

The circulation time becomes

$$\begin{aligned} t_c &= \textit{idle time} + \textit{waiting time} + \textit{service time} \\ &= 5 + 0.582171 + 1 = 6.582171 \quad [\text{time units}]\,. \end{aligned}$$

□

Chapter 10

Applied Queueing Theory

So far we have considered classical queueing systems, where all traffic processes are birth and death processes. They play a key role in queueing theory. 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. To find an simple analytical solution we have to assume either a Poisson arrival process or exponentially distributed service times. In this chapter, we mainly focus on the single server queue.

In Sec. 10.1 we introduce Kendall's notation for queueing systems, and describe queueing disciplines and priority strategies. Sec. 10.2 mentions some general results and concepts as Little's law, work conservation, and load function. The important Pollaczek-Khintchine formula for *M/G/1* is derived in Sec. 10.3, where we also list some results for busy period and moments of waiting time distributions. State probabilities for a finite buffer systems are obtained by Keilson's formula from infinite buffer state probabilities.
The first paper on queueing theory was published by *Erlang in 1909* and dealt with queueing systems with constant service time *M/D/n*. This is more complex than Markovian systems. In Sec. 10.4 we deal with this system in details and derive state probabilities and the waiting time distribution for *FCFS* expressed by state probabilities. A system with Erlang-k arrival process, constant service time and r servers is equivalent to a system with Poisson arrival process, constant service time, and $k \cdot r$ servers. In Sec. 10.5 we consider single-server systems with exponential service times and general renewal arrival processes.

Sec. 10.6 considers more classes of customers with different priorities and different service time distributions. In Sec. 10.6.1 parameters of individual arrival processes and the total arrival process are described. Kleinrock's conservation law is derived in Sec. 10.6.2. Mean waiting times assuming non-preemptive disciplines are derived in Sec. 10.6.3. As a special case we find the mean waiting time for *shortest job first* queueing discipline (Sec. 10.6.4). In Sec. 10.6.5 we consider *M/M/n* with non-preemptive queueing discipline. For preemptive-resume queueing discipline mean waiting times are derived in Sec. 10.6.6. Finally, we consider *round robin* and *processor sharing* queueing disciplines in Sec. 10.7.

10.1 Kendall's classification of queueing models

In this section we shall introduce a compact notations for queueing systems, called Kendall's notation.

10.1.1 Description of traffic and structure

D.G. Kendall (1951 [71]) 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. 2.5):

M	$\sim$	Markov. Exponential time intervals (Poisson arrival process, exponentially distributed service times).
D	$\sim$	Deterministic. Constant time intervals.
E_k	$\sim$	Erlang-k distributed time intervals (E_1 = 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 10.1.1: Ordinary queueing models

$M/M/n$: is a pure delay system with Poisson arrival process, exponentially distributed service times, and n servers. This is the classical Erlang delay system (Chap. 9).

$GI/G/1$: 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 (sometimes only the number of waiting positions),
S = the population size (number of customers),
X = queueing discipline (Sec. 10.1.2).

$K = n$ 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. Index c (clocked) may indicate that the system operates in discrete time. Usually we assume full accessibility.

10.1.2 Queueing strategy: disciplines and organization

Customers waiting in a queue 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.*
This is called an ordered queue, and this discipline is preferred 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. The third discipline does not consider any criteria at all and therefore it requires no 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. The total waiting time for all customers and thus *the mean waiting time is the same for the three above-mentioned disciplines*. The queueing discipline only decides how the waiting time is distributed among customers. In for example a stored-program-controlled 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.

We often try to reduce the total waiting time. This 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 *minimizes 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 = fair queueing.*
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*. When the time slice converges to zero we get:

PS: *Processor Sharing.*
All customers share the service capacity equally.

FB: *Foreground – Background.*
This discipline attempts 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.

10.1.3 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 customer waits until a server becomes idle even if it is serving a customer with lower priority. Furthermore it also waits until all customers with higher priority and customers arriving earlier with same 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 resumed from where it is interrupted when a server becomes available,
- *Preemptive without re-sampling:*
  The service is resumed with the same service time from the beginning,
- *Preemptive with re-sampling:*
  The service is resumed 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. 10.1.2. In queueing literature we meet many other strategies and symbols. *GD* denotes an arbitrary queueing discipline (general discipline).

The behavior of customers is also subject to modeling:

- *Balking* refers to queueing systems, where customers with a queue-length dependent probability may give up joining the queue.
- *Reneging = time-out* refers to systems with impatient customers which after some waiting time abandon the queue without being served.
- *Jockeying* refers to the systems where the customers may jump from one (e.g. a longer) queue to another (e.g. a shorter) queue to obtain faster service.

Thus by combining all options there are many possible models. In this chapter we shall only deal with the most important ones. We mainly consider systems with one server.

Example 10.1.2: Stored Program Controlled (SPC) switching system
In *SPC*–systems tasks of the processors may for example be divided into 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 existing calls has higher priority than detection of new call attempts. □

10.2 General results in 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. 3.3 is the most general result which is valid for an arbitrary queueing system. The theorem is easy to apply and very useful in many applications.

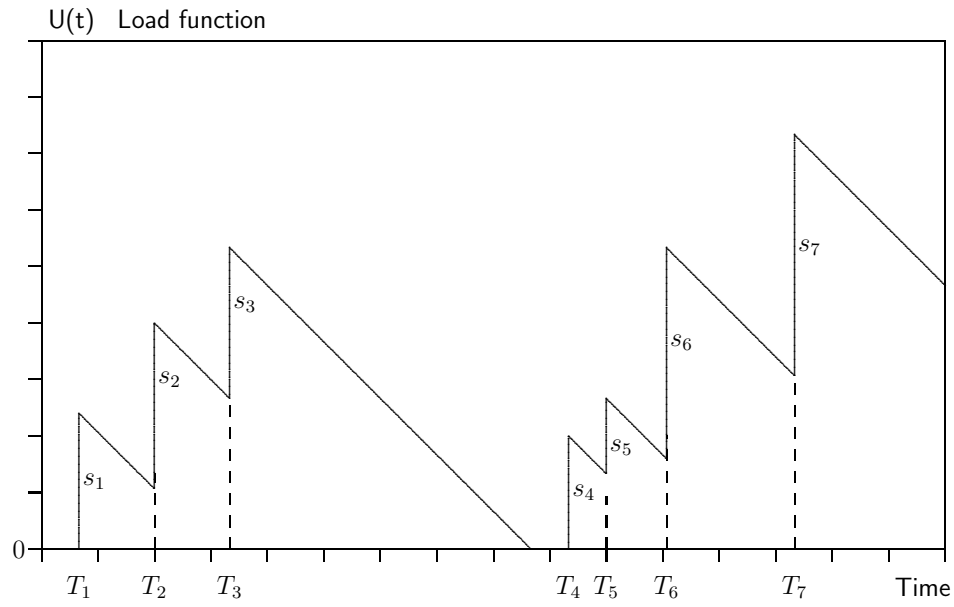


Figure 10.1: *Load function $U(t)$ for the single server queueing system GI/G/1.*

Classical queueing models play a key role in queueing theory, because other systems often converge to these when number of servers increases (Palm-Khintchine theorem 3.1 in Sec. 3.7). Systems that deviate most from the classical models are systems with a single server. However, these systems are also the simplest to deal with.

For 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). When we consider systems with *FCFS* queueing discipline and Poisson arrival processes, the virtual waiting time will be equal to the actual waiting time due to the *PASTA* property: time averages equals call averages).

10.2.1 Load function and work conservation

We introduce two concepts which are widely used in queueing theory.

Work conservation. A system is said to be *work conserving* if

- no servers are idle when there is at last one job waiting,
- service times are independent of the service disciplines.

This will not always be fulfilled in real systems. If the server is a human being, the service rate will often increase with the length of the queue, but after some time the server may become exhausted and the service rate decreases.

Load function. $U(t)$ denotes the time, it will require to serve all customers, which are in the system at time t (Fig. 10.1). At a time of arrival, $U(t)$ increases with a step equal to the service time of the arriving customer. Between arrivals $U(t)$ decreases with a slope depending on the number of working servers 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. For *FCFS* queueing systems the waiting time is equal to the load function at the time of arrival. If the inter-arrival time is $a_i = T_{i+1} - T_i$, then we have *Lindley's equation*:

$$U_{i+1} = max\{0, U_i + s_i - a_i\}\,, \tag{10.1}$$

where U_i is the value of the load function at time T_i.

10.3 Pollaczek-Khintchine's formula

The mean waiting time for a single-server queueing system *M/G/1* with Poisson arrivals and general service times is given by:

Theorem 10.1 *Pollaczek-Khintchine's formula (1930–32):*

$$W = \frac{A \cdot s}{2(1-A)} \cdot \varepsilon\,, \tag{10.2}$$

$$W = \frac{V}{1-A}\,, \tag{10.3}$$

where

$$V = A \cdot \frac{s}{2} \cdot \varepsilon = \frac{\lambda}{2} \cdot m_2\,. \tag{10.4}$$

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 (2.14).

The more regular the service process is, the smaller the mean waiting time will be. The influence of the arrival process is studied in Sec. 10.5. In real telephone traffic the form factor will often be $4-6$, in data traffic $10-100$. Formula (10.2) is one of the most important results in queueing theory, and we will study it carefully. As a special case we have earlier derived the mean waiting time for *M/M/1*, where $\varepsilon = 2$ (Sec. 9.2.4). Later we consider *M/D/1*, where $\varepsilon = 1$ (Sec. 10.4).

10.3.1 Derivation of Pollaczek-Khintchine's formula

We consider the queueing system $M/G/1$ and we want 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 average time it takes for a customer under service to be completed.** Let us at a random point of time consider an arbitrary customer being served. The residual mean service time given by (2.36):

   $$m_{1,r} = \frac{s}{2} \cdot \varepsilon \,,$$

   where s and ε have the same meaning as in (10.2). 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).

   Therefore, the contribution to the mean waiting time from a customer being served becomes:

   $$\begin{aligned} V &= (1-A) \cdot 0 + A \cdot \frac{s}{2} \cdot \varepsilon \,, \\ V &= \frac{\lambda}{2} \cdot m_2 \,. \end{aligned}$$

2. **Waiting time due to customers already arrived and waiting in the queue.** 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 thus becomes:

   $$L \cdot s = \lambda W \, s = A \, W \,.$$

We thus have the total waiting time (10.3) & (10.5):

$$\begin{aligned} W &= V + A\,W \,, \\ W &= \frac{V}{1-A} \\ &= \frac{A\,s}{2(1-A)} \cdot \varepsilon \,, \end{aligned}$$

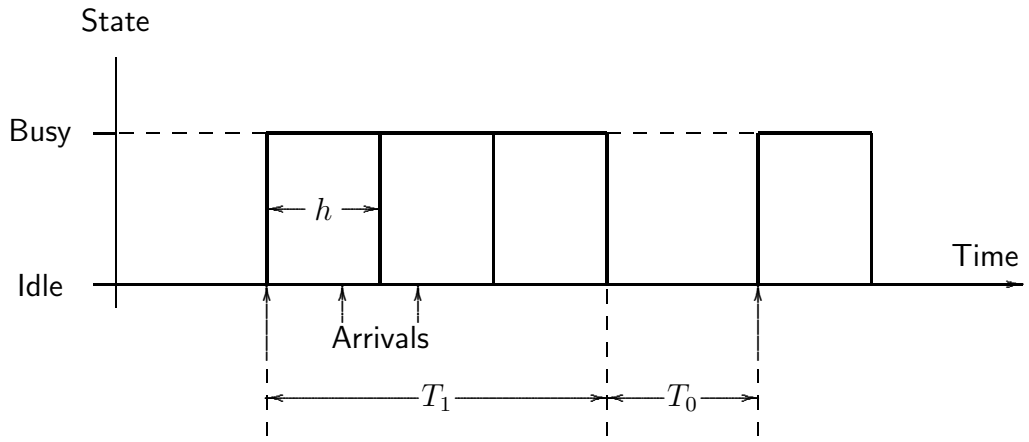


Figure 10.2: *Example of a sequence of events for the system M/D/1 with busy period T_1 and idle period T_0.*

which is Pollaczek-Khintchine's formula (10.2). 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) (2.30):

$$w = \frac{W}{D} = \frac{s}{2(1-A)} \cdot \varepsilon \,. \tag{10.5}$$

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 informative because it shows how ε enters into the formula.

10.3.2 Busy period for *M/G/1*

A busy period of a queueing system with n servers is the time interval from the instant all servers become busy until a server becomes idle again. For a single-server system *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 one cycle from the instant the server changes state from idle to busy till the next time it again 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. 10.2 shows an example with constant service time. The proportion of time the system is busy then becomes:

$$\frac{m_{T_1}}{m_{T_0+T_1}} = \frac{m_{T_1}}{m_{T_0} + m_{T_1}} = A = \lambda \cdot s \,.$$

From $m_{T_0} = 1/\lambda$, we get:

$$m_{T_1} = \frac{s}{1-A}\,. \tag{10.6}$$

During a busy period at least one customer is served, and the average number served is $1/(1-A)$.

10.3.3 Moments of *M/G/1* waiting time distribution

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]). We list the useful results without derivations:

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 ($m_1 = s = 1$):

$$m_{k,F} = \frac{A}{1-A}\sum_{j=1}^{k}\binom{k}{j}\cdot\frac{m_{j+1}}{j+1}\cdot m_{k-j,F}\,, \quad m_{0,F} = 1\,. \tag{10.7}$$

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:

$$\begin{aligned} m_{1,L} &= m_{1,F}\,, \\ m_{2,L} &= \frac{m_{2,F}}{1-A}\,, \\ m_{3,L} &= \frac{m_{3,F} + 3\cdot m_{1,F}\cdot m_{2,F}}{(1-A)^2}\,. \end{aligned} \tag{10.8}$$

10.3.4 Limited queue length: *M/G/1/k*

In real systems the queue length, for example the size of a buffer, will always be finite. Arriving customers are blocked when the buffer is full. 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 [69]):

$$p_k(i) = \begin{cases} \dfrac{p(i)}{1 - A \cdot Q_k}\,, & i = 0, 1, \ldots, k-1\,, \\ \dfrac{(1-A) \cdot Q_k}{1 - A \cdot Q_k} & i = k\,, \end{cases} \tag{10.9}$$

where $A < 1$ is the offered traffic, and:

$$Q_k = \sum_{j=k}^{\infty} p(j) = \sum_{j=0}^{k-1} p(j)\ . \tag{10.10}$$

There exists algorithms for calculating $p(i)$ for arbitrary holding time distributions ($M/G/1$) based on imbedded Markov chain analysis (Kendall, 1953 [72] where the same approach is used for $GI/M/1$).

We notice that $p(i)$ only exists 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. 10.4.8, which is applicable for general holding time distributions.

10.4 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.

10.4.1 Historical remarks on $M/D/n$

The first paper at all on queueing theory was published by Erlang (1909 [29]). He dealt with a system with Poisson arrival Process and constant service times. Intuitively, one would expect 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. 2.1.1), 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 [12]):

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 [35]) 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 applied more theoretical methods based on generating functions. Erlang did not derive state probabilities, but looked for the waiting time distribution.

Crommelin (1932 [21], 1934 [22]), a British telephone engineer, presented a general solution to *M/D/n*. He generalized 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 [73]) dealt with *M/D/n* and derived the waiting time distribution.

10.4.2 State probabilities of *M/D/1*

Under the assumption of statistical equilibrium we now derive the state probabilities of *M/D/1* in a simple way using the approach of Fry. 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: offered traffic = carried traffic = $\lambda \cdot h < 1$ and

$$A = Y = \lambda \cdot h = 1 - p(0)\,, \tag{10.11}$$

as in every state except state zero the carried traffic is equal to one erlang.

To study this system, 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 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 inside the interval $(t,\ t + h)$ of duration h:

$$p(j, h) = p\{j \text{ calls within } h\} = \frac{(\lambda h)^j}{j!} \cdot \mathrm{e}^{-\lambda h}\,, \quad j = 0, 1, 2 \ldots\,. \tag{10.12}$$

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. 10.3).

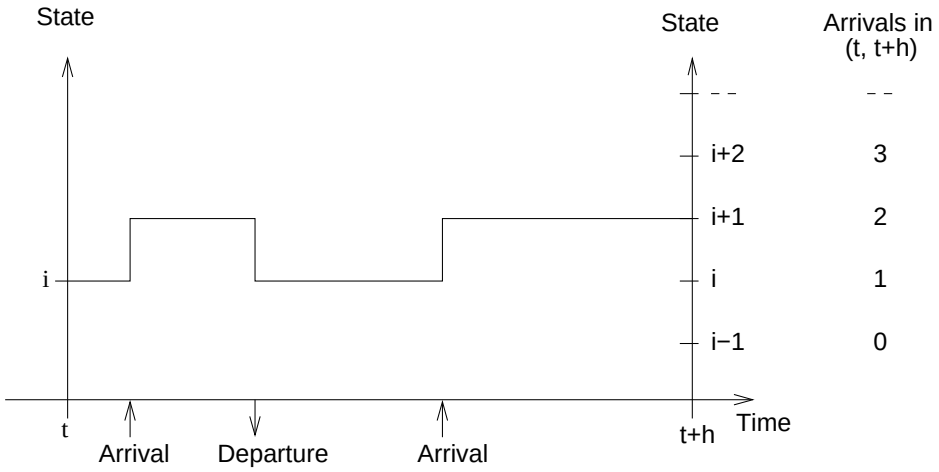


Figure 10.3: *Illustration of Fry's equations of state for the queueing system M/D/1.*

We obtain *Fry's equations of state* for $n = 1$ (Fry, 1928 [35]):

$$p_{t+h}(i) = \{p_t(0) + p_t(1)\}\, p(i,h) + \sum_{j=2}^{i+1} p_t(j) \cdot p(i-j+1,h)\,, \quad i = 0,1,\ldots\,. \tag{10.13}$$

Above (10.11 we found:

$$p(0) = 1 - A\,.$$

Under the assumption of statistical equilibrium and considering mean values for $t \to \infty$ we have $p_t(i) = p_{t+h}(i) = p(i)$. By successively letting $i = 0, 1 \ldots$ we get:

$$\begin{aligned} p(1) &= (1-A)\cdot\left\{\mathrm{e}^{A} - 1\right\}\,, \\ p(2) &= (1-A)\cdot\left\{-\mathrm{e}^{A}\cdot(1+A) + \mathrm{e}^{2A}\right\}\,, \end{aligned}$$

and in general:

$$p(i) = (1-A)\cdot\sum_{j=1}^{i}(-1)^{i-j}\cdot\mathrm{e}^{jA}\cdot\left\{\frac{(jA)^{i-j}}{(i-j)!} + \frac{(jA)^{i-j-1}}{(i-j-1)!}\right\}, \quad i = 2,3,\ldots \tag{10.14}$$

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, but this is not necessary in this case where we know $p(0)$.

10.4.3 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*):

$$D = A = 1 - p(0)\,. \tag{10.15}$$

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 (2.30):

$$w = \frac{W}{D}\,. \tag{10.16}$$

W and w are easily obtained by using Pollaczek-Khintchine's formula (10.2):

$$W = \frac{A \cdot h}{2(1-A)}\,, \tag{10.17}$$

$$w = \frac{h}{2(1-A)}\,. \tag{10.18}$$

The mean value of a busy period was obtained for *M/G/1* in (10.6) and is illustrated for constant service times in Fig. 10.2:

$$m_{T_1} = \frac{h}{1-A}\,. \tag{10.19}$$

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*:

$$B(i) = \frac{(i\,A)^{i-1}}{i!}\,\mathrm{e}^{-i\,A}, \qquad i = 1, 2, \ldots \tag{10.20}$$

10.4.4 Waiting time distribution: *M/D/1, FCFS*

This can be shown to be:

$$p\{W \le t\} = 1 - (1-\lambda) \cdot \sum_{j=1}^{\infty} \frac{\{\lambda(j-\tau)\}^{T+j}}{(T+j)!} \cdot \mathrm{e}^{-\lambda(j-\tau)}\,, \tag{10.21}$$

where $h = 1$ is chosen as time unit, $t = T + \tau$, T is an integer, and $0 \le \tau < 1$.

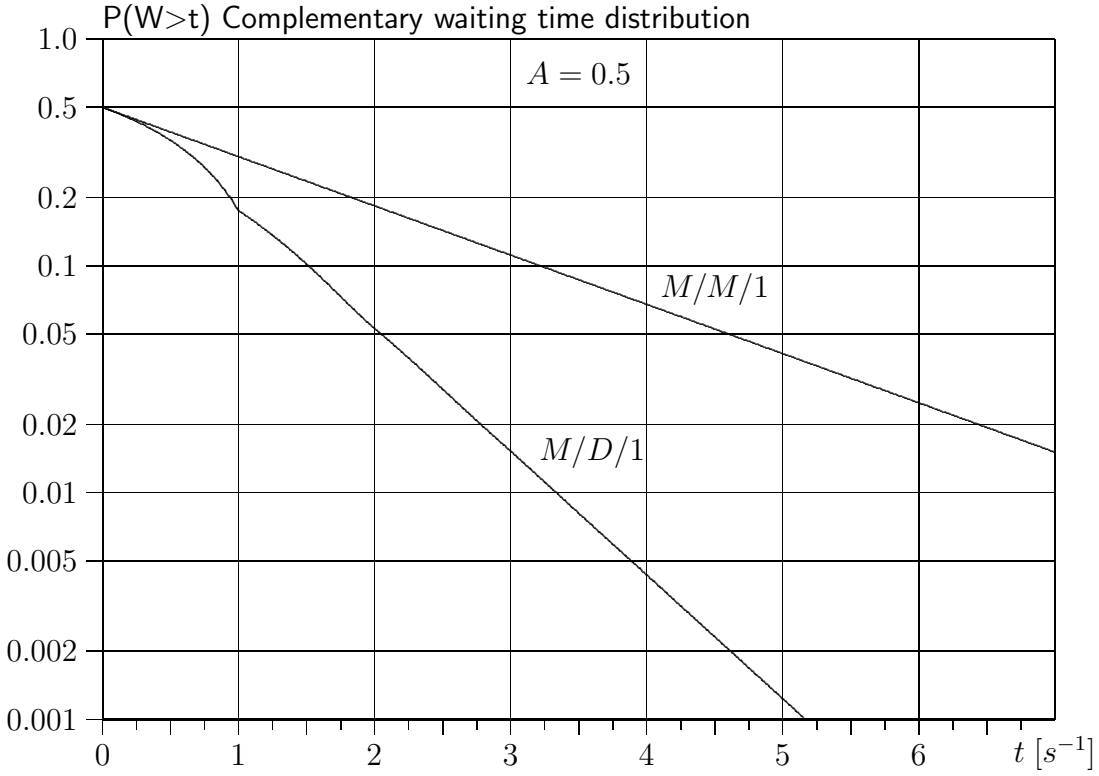


Figure 10.4: *The complementary waiting time distribution for all customers in the queueing system M/M/1 and M/D/1 for ordered queue (FCFS). Time unit = mean service time. We notice that the mean waiting time for M/D/1 is only half of that for M/M/1.*

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. 10.4.

Formula (10.21) is not suitable for numerical evaluation. It can be shown (Iversen, 1982 [45]) that the waiting time can be written in a closed form, as given by Erlang in 1909:

$$p\{W \leq t\} = (1-\lambda) \cdot \sum_{j=0}^{T} \frac{\{\lambda(j-t)\}^j}{j!} \cdot \mathrm{e}^{-\lambda(j-t)}\,, \tag{10.22}$$

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 [45]) that for an integral value of t we have:

$$p\{W \leq t\} = p(0) + p(1) + \cdots + p(t)\,. \tag{10.23}$$

The state probabilities $p(i)$ are calculated accurately by using a recursive formula based on

Fry's equations of state (10.14):

$$p(i+1) = \frac{1}{p(0,h)} \left\{ p(i) - \{p(0)+p(1)\} \cdot p(i,h) - \sum_{j=2}^{i} p(j) \cdot p(i-j+1,h) \right\} . \qquad (10.24)$$

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 by a Binomial expansion (10.22) may be written in powers of τ, where

$$t = T + \tau, \quad T \text{ integer}, \quad 0 \leq \tau < 1 .$$

We find:

$$p\{W \leq T + \tau\} = \mathrm{e}^{\lambda\tau} \sum_{j=0}^{T} \frac{(-\lambda\tau)^j}{j!} \cdot p\{W \leq T - j\} , \qquad (10.25)$$

where $p\{W \leq T - j\}$ is given by (10.23).

The numerical evaluation is very accurate when using (10.23), (10.24) and (10.25).

10.4.5 State probabilities: *M/D/n*

When setting up *Fry's equations of state* (10.13) we obtain more combinations:

$$p_{t+h}(i) = \left\{ \sum_{j=0}^{n} p_t(j) \right\} p(i,h) + \sum_{j=n+1}^{n+i} p_t(j) \cdot p(n+i-j,h) . \qquad (10.26)$$

On the assumption of statistical equilibrium ($A < n$) we can leave out of account the absolute points of time:

$$p(i) = \left\{ \sum_{j=0}^{n} p(j) \right\} p(i,h) + \sum_{j=n+1}^{n+i} p(j) \cdot p(n+i-j,h), \qquad i = 0, 1, \ldots \qquad (10.27)$$

The system of equations (10.27) 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 (10.27) 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*, [12] pp. 75–83).

10.4.6 Waiting time distribution: *M/D/n, FCFS*

The waiting time distribution is given by Crommelin's distribution:

$$p\{W \le t\} = 1 - \sum_{i=0}^{n-1} \sum_{k=0}^{i} p(k) \cdot \sum_{j=1}^{\infty} \frac{\{A(j-\tau)\}^{(T+j+1)n-1-i}}{\{(T+j+1)n-1-i\}!}\,, \tag{10.28}$$

where A is the offered traffic and

$$t = T \cdot h + \tau, \qquad 0 \le \tau < h\,. \tag{10.29}$$

Formula (10.28) can be written in a closed form in analogy with (10.22):

$$p\{W \le t\} = \sum_{i=0}^{n-1} \sum_{k=0}^{i} p(k) \sum_{j=0}^{T} \frac{\{A(j-t)\}^{j\cdot n+n-1-i}}{\{j\cdot n+n-1-i\}!} \cdot \mathrm{e}^{-A(j-t)}\,. \tag{10.30}$$

For integral values of the waiting time t we have:

$$p\{W \le t\} = \sum_{j=0}^{n(t+1)-1} p(j)\,. \tag{10.31}$$

For non-integral waiting times $t = T + \tau,\ T$ integer, $0 \le \tau < 1$, we are able to express the waiting time distribution in terms of integral waiting times as for *M/D/1*:

$$p\{W \le t\} = p\{W \le T+\tau\} = \mathrm{e}^{\lambda\tau} \sum_{j=0}^{k} \left\{ \frac{(-\lambda\tau)^j}{j!} \cdot \sum_{i=0}^{k-j} p(i) \right\}\,, \tag{10.32}$$

where $k = n(T+1)-1$ and $p(i)$ is the state probability (10.27).

The exact mean waiting time of all customers W is difficult to derive. An approximation was given by Molina:

$$W \approx \frac{n}{n+1} \cdot E_{2,n}(A) \cdot \frac{h}{n-A} \cdot \frac{1-\left(\frac{A}{n}\right)^{n+1}}{1-\left(\frac{A}{n}\right)^{n}}\,. \tag{10.33}$$

For any queueing system with infinite queue we have (2.30):

$$w = \frac{W}{D}\,,$$

where for all values of n:

$$D = 1 - \sum_{j=0}^{n-1} p(j)\,.$$

10.4.7 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 r servers made up from the servers:

$$x, x+k, x+2\cdot k, \ldots, x+(r-1)\cdot k\,, \qquad 0 < x \leq k\,. \tag{10.34}$$

will serve just every $k'th$ customer. If we consider the servers (10.34), 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 (10.34). State probabilities and mean waiting times 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 10.4.1: 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:

$$\begin{aligned}
&\mathrm{E}_4/\mathrm{E}_1/2: && L = 4.5174\ ,\\
&\mathrm{E}_4/\mathrm{E}_2/2: && L = 2.6607\ ,\\
&\mathrm{E}_4/\mathrm{E}_3/2: && L = 2.0493\ ,\\
&\mathrm{E}_4/\mathrm{D}/2: && L = 0.8100\ .
\end{aligned}$$

□

10.4.8 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).

As mentioned in Sec. 10.3.4 the state probabilities $p_k(i)$ of the finite buffer system are obtained from the state probabilities $p(i)$ of the infinite buffer system by using (**??**) & (**??**). Integral waiting times are obtained from the state probabilities, and non-integral waiting times from integral waiting times as shown above (Sec. 10.4.4).

For the infinite buffer system the state probabilities only exist when the offered traffic is less than the capacity ($A < n$). But for a finite buffer system the state probabilities also exist for $A > n$, but we cannot obtain them by the above-mentioned method.

For *M/D/1/k* the finite buffer state probabilities $p_k(i)$ can be obtained for any offered traffic in the following way. In a system with one server and $(k-1)$ queueing positions we have $(k+1)$ states $(0, 1, \cdots, k)$. Fry's balance equations for state probabilities $p_k(i), i = 0,\ 1,\ \ldots\ , k-2$, yielding $k-1$ linear equations between the states $\{p_k(0), p_k(1), \ldots, p_k(k-1)\}$. But it is not possible to write down simple time-independent equations for state $k-1$ and k. However, the first $(k-1)$ equations (10.13) together with the normalization requirement

$$\sum_{j=0}^{k} p_k(j) = 1 \tag{10.35}$$

and the fact that the offered traffic equals the carried traffic plus the rejected traffic (*PASTA property*):

$$A = 1 - p_k(0) + A \cdot p_k(k) \tag{10.36}$$

results in $(k+1)$ independent linear equations, which are easy to solve numerically. The two approaches yields of course the same result. The first method is only valid for $A < 1$, whereas the second is valid for any offered traffic.

Example 10.4.2: 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. 10.1). 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. □

10.5 Single server queueing system: *GI/G/1*

In Sec. 10.3 we showed that the mean waiting time for all customers in queueing system *M/G/1* is given by Pollaczek-Khintchine's formula:

$$W = \frac{A \cdot s}{2(1-A)} \cdot \varepsilon \tag{10.37}$$

where ε is the form factor of the holding time distribution.

We have earlier analyzed the following cases:

M/M/1 (Sec. 9.2.4): $\varepsilon = 2$:

$$W = \frac{A \cdot s}{(1-A)}, \qquad \text{Erlang 1917.} \tag{10.38}$$

M/D/1 (Sec. 10.4.3): $\varepsilon = 1$:

$$W = \frac{A \cdot s}{2(1-A)}, \qquad \text{Erlang 1909.} \tag{10.39}$$

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.

10.5.1 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:

$$GI/G/1: \qquad W \leq \frac{A \cdot s}{2(1-A)} \cdot \left\{ \frac{v_a + v_d}{s^2} \right\}. \tag{10.40}$$

This formula shows that it is the stochastic variations, that results in waiting times.

Formula (10.40) gives the upper theoretical boundary. A realistic estimate of the actual mean waiting time is obtained by *Marchal's approximation* (Marchal, 1976 [88]):

$$W \approx \frac{A \cdot s}{2(1-A)} \cdot \left\{ \frac{v_a + v_d}{s^2} \right\} \cdot \left\{ \frac{s^2 + v_d}{a^2 + v_d} \right\} . \tag{10.41}$$

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 $M/G/1$.

10.5.2 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. 3.2.2), 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 [72]):

$$\pi(i) = (1-\alpha)\alpha^i , \qquad i = 0, 1, 2, \ldots \tag{10.42}$$

where α is the positive real root satisfying the equation:

$$\alpha = \int_0^\infty \mathrm{e}^{-\mu(1-\alpha)t} f(t)\, \mathrm{d}t . \tag{10.43}$$

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. 10.4.5).

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:

$$\begin{aligned}
\pi_{t_2}(0) &= \sum_{j=0}^{\infty} \pi_{t_1}(j) \cdot \left\{ 1 - \sum_{i=0}^{j} p(i) \right\} , \\
\pi_{t_2}(1) &= \sum_{j=0}^{\infty} \pi_{t_1}(j) \cdot p(j) , \\
&\vdots \quad \vdots \\
\pi_{t_2}(i) &= \sum_{j=0}^{\infty} \pi_{t_1}(j) \cdot p(j-i+1) .
\end{aligned} \tag{10.44}$$

The normalization condition is as usual:

$$\sum_{i=0}^{\infty} \pi_{t_1}(i) = \sum_{j=0}^{\infty} \pi_{t_2}(j) = 1 . \tag{10.45}$$

It can be shown that the above-mentioned geometric distribution is the only solution to this system of equations (Kendall, 1953 [72]).

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).

10.5.3 Characteristics of *GI/M/1*

The probability of immediate service becomes:

$$p\{\text{immediate}\} = \pi(0) = 1 - \alpha . \tag{10.46}$$

The corresponding probability of being delayed the becomes:

$$D = p\{\text{delay}\} = \alpha . \tag{10.47}$$

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:

$$\begin{aligned} L_1 &= \sum_{i=1}^{\infty} (1-\alpha)\,\alpha^i\,(i-1)\,, \\ L_1 &= \frac{\alpha^2}{1-\alpha}\,. \end{aligned} \tag{10.48}$$

The average number of customers in the system before an arrival epoch is:

$$\begin{aligned} L_2 &= \sum_{i=0}^{\infty} (1-\alpha)\,\alpha^i \cdot i \\ &= \frac{\alpha}{1-\alpha}\,. \end{aligned} \tag{10.49}$$

The average waiting time for all customers then becomes:

$$W = \frac{1}{\mu} \cdot \frac{\alpha}{1-\alpha}\,. \tag{10.50}$$

The average queue length taken over the whole time axis (the virtual queue length) therefore becomes (Little's theorem):

$$L = A \cdot \frac{\alpha}{1-\alpha}\,. \tag{10.51}$$

The mean waiting time for customers, who experience a positive waiting times, becomes

$$\begin{aligned} w &= \frac{W}{D}\,, \\ w &= \frac{1}{\mu} \cdot \frac{1}{1-\alpha}\,. \end{aligned} \tag{10.52}$$

Example 10.5.1: 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 (10.50):

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. □

10.5.4 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 (10.52), when the queueing discipline is *FCFS* (Sec. 9.4 and Fig. 2.12).

10.6 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 organizing the queue, the waiting times can be distributed among the customers according to our preferences.

10.6.1 Combination of several classes of customers

We now classify the customers in N different 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 offered traffic is $A_i = \lambda_i \cdot s_i$. The second moment of the service time distribution is denoted by m_{2i}.

In stead of considering the individual arrival processes, we may consider the total arrival process, which also is a Poisson arrival process with intensity:

$$\lambda = \sum_{i=1}^{N} \lambda_i \,. \tag{10.53}$$

The resulting service time distribution then becomes a weighted sum of service time distributions of the individual classes (Sec. 2.3.2: combination in parallel). The total mean service time becomes (2.62):

$$s = \sum_{i=1}^{N} \frac{\lambda_i}{\lambda} \cdot s_i \,, \tag{10.54}$$

and the total second moment is (2.61):

$$m_2 = \sum_{i=1}^{N} \frac{\lambda_i}{\lambda} \cdot m_{2i} \,. \tag{10.55}$$

The total offered traffic becomes:

$$A = \sum_{i=1}^{N} A_i = \sum_{i=1}^{N} \lambda_i \cdot s_i = \lambda \, s \,. \tag{10.56}$$

The remaining mean service time at a random point of time becomes (10.4):

$$V_{1,N} = \frac{1}{2} \cdot \lambda \cdot m_2 \tag{10.57}$$

$$= \frac{1}{2} \cdot A \cdot \frac{m_2}{s}$$

$$= \frac{1}{2} \cdot A \cdot \frac{\sum_{i=1}^{N} \frac{\lambda_i}{\lambda} \cdot m_{2i}}{\sum_{i=1}^{N} \frac{\lambda_i}{\lambda} \cdot s_i} = \frac{1}{2} \cdot A \cdot \frac{\sum_{i=1}^{N} \lambda_i \cdot m_{2i}}{\sum_{i=1}^{N} A_i}$$

$$V_{1,N} = \sum_{i=1}^{N} \frac{1}{2} \cdot \lambda_i \cdot m_{2i} \tag{10.58}$$

$$V_{1,N} = \sum_{i=1}^{N} V_i \ , \tag{10.59}$$

where index $(1, N)$ on left hand side indicates that we include all streams from 1 to N.

10.6.2 Kleinrock's conservation law

We now consider a system with several classes of customers. We assume that the queueing discipline is independent of the service time. This excludes for example preemptive resume queueing discipline as the probability of preemption increases with the service time. The waiting time is composed 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 waiting time becomes:

$$W = V_{1,N} + \sum_{i=1}^{N} L_i \cdot s_i \ .$$

L_i is the average queue length for customers of type i. By applying Little's law we get:

$$W = V_{1,N} + \sum_{i=1}^{N} \lambda_i \cdot W_i \cdot s_i$$

$$W = V_{1,N} + \sum_{i=1}^{N} A_i \cdot W_i \ . \tag{10.60}$$

We may also combine all customer classes into one class and apply Pollaczek-Khintchine's formula to get the same mean waiting time (10.5):

$$W = V_{1,N} + A \cdot W \ , \tag{10.61}$$

Under these general assumptions we get Kleinrock's conservation law (Kleinrock, 1964 [75]):

Theorem 10.2 *Kleinrock's conservation law:*

$$\sum_{i=1}^{N} A_i \cdot W_i = A \cdot W = A \cdot \frac{V_{1,N}}{1-A} = \textit{constant}. \tag{10.62}$$

The average waiting time for all classes weighted by the traffic (load) of the mentioned class, is independent of the queue discipline.

For the total traffic process we have Pollaczek-Khintchine's formula. We may thus give a small proportion of the traffic a very low mean waiting time, without increasing very much the average waiting time of the remaining customers. By various strategies we may allocate waiting times to individual customers according to our preferences.

10.6.3 Non-preemptive queueing discipline

In the following we look at *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 mean service time s_p and arrival intensity λ_p. In Sec. 10.6.1 we derived parameters for the total traffic process.

The total average waiting time W_p of a class p customers is made up of the following three contributions:

a) Residual service time $V_{1,N}$ for the customer under service.

b) Waiting time, due to the customers in the queue with priority p or higher, which already are in the queues (Little's theorem):

$$\sum_{i=1}^{p} s_i \cdot (\lambda_i \cdot W_i) \ .$$

c) Waiting time due to customers with higher priority, which overtake the customer we consider while this is waiting:

$$\sum_{i=1}^{p-1} s_i \cdot L_i = \sum_{i=1}^{p-1} \lambda_i \, W_p \cdot s_i \, .$$

In total we get:

$$W_p = V_{1,N} + \sum_{i=1}^{p} s_i \cdot \lambda_i \cdot W_i + \sum_{i=1}^{p-1} s_i \cdot \lambda_i \cdot W_p \,. \tag{10.63}$$

For highest priority customers of class one we get under the assumption of *FCFS*:

$$\begin{aligned} W_1 &= V_{1,N} + L_1 \cdot s_1 \qquad (10.64)\\ &= V_{1,N} + A_1 \cdot W_1 \,, \\ W_1 &= \frac{V_{1,N}}{1 - A_1} \,. \qquad (10.65) \end{aligned}$$

$V_{1,N}$ is the residual service time for the customer being served when the customer we consider arrives (10.58):

$$V_{1,N} = \sum_{i=1}^{N} \frac{\lambda_i}{2} \cdot m_{2i} \,, \tag{10.66}$$

where m_{2i} is the second moment of the service time distribution of the i'th class.

For class two customers we find (10.63):

$$W_2 = V_{1,N} + L_1 \cdot s_1 + L_2 \cdot s_2 + s_1 \cdot \lambda_1 \cdot W_2 \,.$$

Replacing the first two terms by W_1 (10.64), we get:

$$\begin{aligned} W_2 &= W_1 + A_2 \cdot W_2 + A_1 \cdot W_2 \,, \\ W_2 &= \frac{W_1}{1 - A_1 - A_2} \,, \qquad (10.67)\\ W_2 &= \frac{V_{1,N}}{\{1 - A_1\}\,\{1 - (A_1 + A_2)\}} \,. \qquad (10.68) \end{aligned}$$

In general we find (Cobham, 1954 [15]):

$$W_p = \frac{V_{1,N}}{\{1 - A_{0,p-1}\}\,\{1 - A_{0,p}\}} \,, \tag{10.69}$$

where:

$$A_{0,p} = \sum_{i=0}^{p} A_i \,, \qquad A_0 = 0 \,. \tag{10.70}$$

The structure of formula (10.69) can be interpreted directly. All customers wait until the service in progress is completed $\{V_{1,N}\}$ no matter which class they belong to. Furthermore, waiting time is due to already arrived customers of at least have the same priority $\{A_{0,p}\}$, and customers with higher priority arriving during the waiting time $\{A_{0,p-1}\}$.

Example 10.6.1: SPC-system
We consider a computer which serves two types of customers. The first type has the constant service time of $s_1 = 0.1$ second, and the arrival intensity is $\lambda_1 = 1$ customer/second. The second type has the exponentially distributed service time with the mean value of $s_2 = 1.6$ second and the arrival intensity is $\lambda_2 = 0.5$ customer/second.

The load from the two types customers is then $A_1 = 0.1$ erlang, respectively $A_2 = 0.8$ erlang. From (10.66) 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 (10.2):

$$W = \frac{1.2850}{1 - (0.8 + 0.1)} = 12.85 \; s \, .$$

By non-preemptive priority we find:

Type one highest priority:

$$W_1 = \frac{1.285}{1 - 0.1} = 1.43 \; s \, ,$$

$$W_2 = \frac{W_1}{1 - (A_1 + A_2)} = 14.28 \; s \, .$$

Type two highest priority:

$$W_2 = 6.43 \; s \, ,$$

$$W_1 = 64.25 \; s \, .$$

This shows that we can prioritize type one customers, which have a low traffic and small mean service times, almost without influencing type two. But we should never give priority to traffic with high load and large mean service times. The constant in the *Conservation law* (10.62) becomes the same without priority (Pollaczek-Khintchine formula) as with non–preemptive priority:

$$0.9 \cdot 12.85 = 0.1 \cdot 1.43 + 0.8 \cdot 14.28 = 0.8 \cdot 6.43 + 0.1 \cdot 64.25 = 11.57 \, .$$

□

10.6.4 *SJF*-queueing discipline: *M/G/1*

By the SJF-queueing discipline the shorter the service time of a customer is, the higher is the priority. The *SJF* discipline results in the lowest possible total waiting time. By introducing an infinite number of priority classes,

$$(0, \Delta t), \; (\Delta t, 2\Delta t), \; (2\Delta t, 3\Delta t), \; \ldots$$

we obtain from the formula (10.69) that a customer with the service time t has the mean waiting time W_t (Phipps 1956):

$$W_t = \frac{V_{0,\infty}}{(1 - A_{0,t})^2}\,, \tag{10.71}$$

where $A_{0,t}$ is load from the customers with service time less than or equal to t. When Δt is small $A_{0,t} \approx A_{0,t+\Delta t}$.

If different priority classes have different costs per time unit of waiting, 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 10.6.2: $M/M/1$ with SJF queue discipline
We consider exponentially distributed holding times with the mean value $1/\mu$ which are 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. 2.3).

The contribution to the total traffic A from the customers with service time $\leq t$ is obtained from (2.33) multiplied by $A = \lambda \cdot \mu$:

$$A_{0,t} = A\left\{1 - \mathrm{e}^{-\mu t}(\mu t + 1)\right\}\,.$$

Inserting this in (10.71) we find W_t as illustrated in Fig. 10.5, where the *FCFS*–strategy (same mean waiting time as *LCFS* and *SIRO*) is shown for comparison as function of the actual holding time. The mean waiting time for all customers is less for *SJF* than for *FCFS*, but this is not obvious from the figure. The mean waiting time for *SJF* becomes:

$$\begin{aligned} W_{SJF} &= \int_0^\infty W_t \cdot f(t)\,\mathrm{d}t \\ &= \int_0^\infty \frac{V_{0,\infty}}{(1 - A_{0,t})^2} \cdot f(t)\,\mathrm{d}t \\ &= \int_0^\infty \frac{A \cdot \mathrm{e}^{-\mu t}\,\mathrm{d}t}{\left\{1 - A\left(1 - \mathrm{e}^{-\mu t}\left(\mu t + 1\right)\right)\right\}^2} \end{aligned}$$

which it is not elementary to calculate. □

10.6.5 $M/M/n$ with non-preemptive priority

We may generalize the above to Erlang's classical waiting time system $M/M/n$ with non–preemptive queueing disciplines, when all classes of customers have the same exponentially

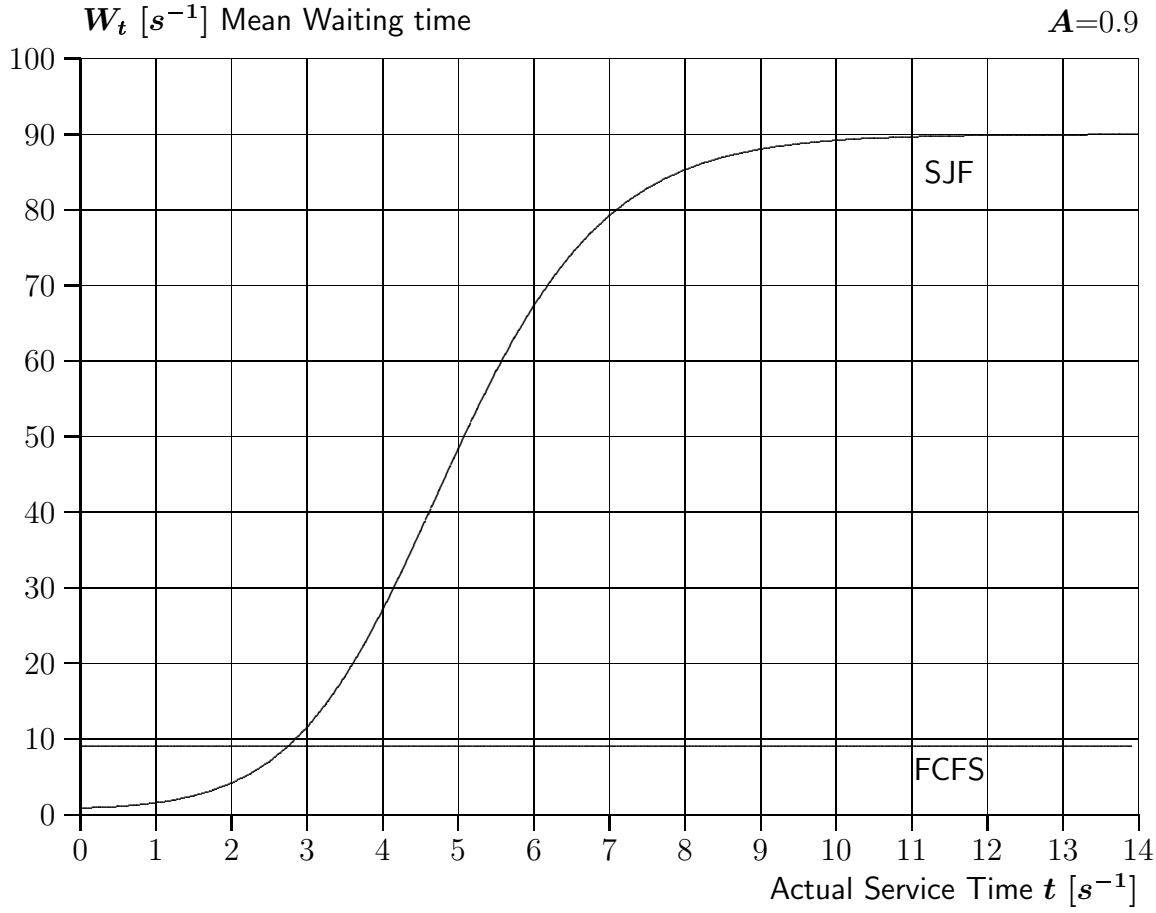


Figure 10.5: *The mean waiting time W_t is a function of the actual service time in a M/M/1–system for SJF and FCFS disciplines, respectively. The offered traffic is 0.9 erlang and the mean service time is chosen as time unit. Notice that for SJF the minimum average waiting time is 0.9 time units, because 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 reduced 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.*

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:

$$W_p = V_{1,N} + \sum_{i=1}^{p} \frac{s}{n} \cdot L_i + W_p \sum_{i=1}^{p-1} \frac{s}{n} \cdot \lambda_i \,,$$

$$W_p = E_{2,n}(A) \cdot \frac{s}{n} + \sum_{i=1}^{p} \left\{ \frac{s \cdot \lambda_i}{n} \cdot W_i \right\} + W_p \sum_{i=1}^{p-1} \frac{s \cdot \lambda_i}{n} \,.$$

A is the total offered traffic for all classes. $V_{1,N}$ is the time until a server becomes idle, which is the probability of delay multiplied by the mean time until a server becomes idle. The probability of delay $E_{2,n}(A)$ is given by Erlang's C-formula, and when all servers are busy customers are served with the mean inter-departure time s/n. For highest priority class $p = 1$ we find:

$$W_1 = E_{2,n}(A) \frac{s}{n} + \frac{1}{n} A_1 \cdot W_1 \,,$$

$$W_1 = E_{2,n}(A) \cdot \frac{s}{n - A_1} \,. \tag{10.72}$$

For $p = 2$ we find in a similar way:

$$W_2 = E_{2,n}(A) \cdot \frac{s}{n} + \frac{1}{n} A_1 \cdot W_1 + \frac{1}{n} A_2 \cdot W_2 + W_2 \cdot \left\{ \frac{s}{n} \cdot \lambda_1 \right\}$$

$$= W_1 + \frac{1}{n} \cdot A_2 \, W_2 + \frac{1}{n} \cdot A_1 \cdot W_2 \,,$$

$$W_2 = \frac{n \cdot s \cdot E_{2,n}(A)}{\{n - A_1\}\{n - (A_1 + A_2)\}} \,. \tag{10.73}$$

In general we find (Cobham, 1954 [15]):

$$W_p = \frac{n \cdot s \cdot E_{2,n}(A)}{\{n - A_{0,p-1}\}\{n - A_{0,p}\}} \tag{10.74}$$

$$= \frac{\frac{s}{n} \cdot E_{2,n}(A)}{\left\{1 - \frac{A_{0,p-1}}{n}\right\}\left\{1 - \frac{A_{0,p}}{n}\right\}} \,.$$

In the last form we see it is similar to (10.69).

10.6.6 Preemptive-resume queueing discipline

We now assume that a customer being served is interrupted by the arrival of a customer with higher priority. Later on, the service continues from where it was interrupted. This situation

is typical for computer systems. For a customer with priority p, the customers with lower priority do no exist. 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:

$$\frac{V_{1,p}}{1-A_{0,p}}\,, \quad \textit{where} \quad V_{1,p}=\sum_{i=1}^{p}\frac{\lambda_i}{2}\cdot m_{2,i}\,, \tag{10.75}$$

is the expected remaining service time due to customers with higher or same priority, and $A_{0,p}$ is given by (10.70).

b) Waiting time due to the customers with higher priority who arrive during the waiting time or service time and interrupt the customer considered:

$$(W_p+s_p)\sum_{i=1}^{p-1}s_i\cdot\lambda_i=(W_p+s_p)\cdot A_{0,p-1}\,.$$

We thus get:

$$W_p=\frac{V_{1,p}}{1-A_{0,p}}+(W_p+s_p)\cdot A_{0,p-1}\,.$$

This can be rewritten as follows:

$$W_p(1-A_{0,p-1})=\frac{V_{1,p}}{\{1-A_{0,p}\}}+s_p\cdot A_{0,p-1}\,,$$

resulting in:

$$W_p=\frac{V_{1,p}}{(1-A_{0,p-1})\,(1-A_{0,p})}+\frac{A_{0,p-1}}{1-A_{0,p-1}}\cdot s_p\,. \tag{10.76}$$

For highest priority customers we get Pollaczek-Khintchine's formula for this class alone, as they are not disturbed by lower priorities ($V_{1,1}=V_1$):

$$W_1=\frac{V_1}{1-A_1}\,. \tag{10.77}$$

The total response time becomes:

$$T_p=W_p+s_p\,.$$

In a similar way as in Sec. 10.6.4 we may write down the formula for average waiting time for *SJF*–queueing discipline with preemptive resume.

Example 10.6.3: SPC–system (example 10.6.1 continued)
We now assume the computer system is working with the discipline preemptive-resume and find:

Type one highest priority:

$$W_1 = \frac{\frac{1}{2}\cdot(0.1)^2}{1-0.1} = 0.0056\ s\,,$$

$$W_2 = \frac{1.2850}{(1-0.1)(1-0.9)} + \frac{0.1}{1-0.1}\cdot 1.6 = 14.46\ s\,.$$

Type two highest priority:

$$W_2 = \frac{\frac{1}{2}\cdot 0.5\cdot 2\cdot(1.6)^2}{1-0.8} + 0 = 6.40\ s\,,$$

$$W_1 = \frac{1.2850}{(1-0.8)(1-0.9)} + \frac{0.8}{1-0.8}\cdot 0.1 = 64.65\ s\,.$$

This shows that by upgrading type one to the highest priority, we can give these customers a very short waiting time, without disturbing type two 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 case with general service time distribution (G) a job may be preempted several times, and therefore the remaining service time will not be given by V. □

10.6.7 $M/M/n$ with preemptive-resume priority

For $M/M/n$ the case of preemptive resume is more difficult to deal with. Only if all customers have the same mean service time, then the mean waiting time can be obtained by first considering class one alone (9.15), then consider class one and two together, which implies the waiting time for class two, etc. The conservation law is valid when all customers have the same exponentially distributed service time.

10.7 Fair Queueing: Round Robin, Processor-Sharing

The Round Robin (RR) queueing model (Fig. 10.6) is a model for a time-sharing computer system, where we want a fast response time for short jobs. This queueing discipline is also called *fair queueing* because the available resources are equally distributed among the jobs (customers) in the system.

New jobs are placed in a *FCFS*–queue, where they wait until they obtain service limited to one time slice (slot) Δs which is the same for all jobs. If a job 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 obtained.

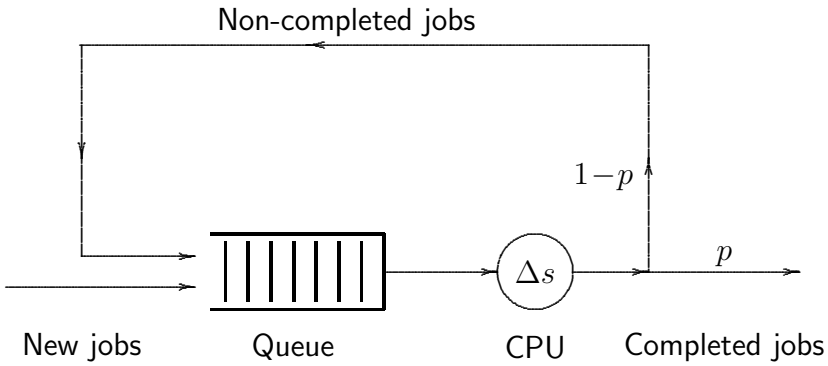


Figure 10.6: *Round robin queueing system. A task is allocated a time slice Δs (at most) every time it is served. If the task is not finished during this time slice, it is returned to a FCFS queue, where it waits on equal terms with new tasks. If we let Δs decrease to zero we obtain the PS (Processor Sharing) queueing discipline.*

We assume that the queue is unlimited, and that new jobs arrive according to a Poisson process (arrival rate λ). The service time distribution can be a general distribution with mean value s.

The size of the time slice can vary. If it becomes infinite, all jobs will be completed the first time, and we have 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 important analytical properties.

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 x jobs in the system, each of them obtain the fraction $1/x$ of the capacity of the computer. So there is no real queue, as all jobs are served all the same, eventually at a lower rate.

In next chapter we deal with processor sharing systems in more detail. In processor sharing systems and queueing networks we consider sojourn time. The job is served all the time, but eventually at a reduced rate. If the job is alone, the service time would be s. The increase in service time is called the *virtual delay* time

The state transition diagrams are identical for the classical *M/M/1* system and for the *M/M/1–PS* system, and thus the performance measures based on state probabilities are identical for the two systems. The is called the *magic property of processor sharing*. When the offered traffic $A = \lambda \cdot s$ is less than one, the steady state probabilities are given by (9.30):

$$p(i) = (1 - A) \cdot A^i \,, \qquad i = 0, 1, \ldots \,, \tag{10.78}$$

i.e. a geometric distribution with mean value $A/(1 - A)$. The mean sojourn time (average response time = time in system) for jobs with duration t becomes:

$$R_t = \frac{t}{1 - A} \,. \tag{10.79}$$

If this job was alone in the system, then its holding time would be t. The average virtual delay for jobs with duration t is:

$$\begin{aligned} W_t &= R_t - t \\ &= \frac{A}{1-A} \cdot t \,. \end{aligned} \tag{10.80}$$

The corresponding mean values for a random job (mean service time s) becomes:

$$R = \frac{s}{1-A} \,, \tag{10.81}$$

$$W = \frac{A}{1-A} \cdot s \,. \tag{10.82}$$

This shows that we obtain the same mean values as for *M/M/1* (Sec. 9.2.4). 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 obtained earlier for *M/G/1* (Pollaczek-Khintchine's formula (10.2)) the results may surprise our intuition.

A very useful property of the Processor-Sharing model is that the departure process is a Poisson process like the arrival process , i.e. we have a reversible system. The Processor-Sharing model is very useful for analyzing time-sharing systems and for modeling queueing networks (Chap. 12). In Chap. 11 we study reversible systems in much more details.

Chapter 11

Multi-service queueing systems

In this chapter we consider queueing systems with more than one type (class, service, stream) of customers. It is analogous to Chap. 7 where we considered loss systems with more types of customers and noticed that the product form was maintained between streams so that the convolution algorithm could be applied. We are mainly interested in reversible systems, where departure processes are of same type as the arrival processes. In queueing terminology, customers of a given type make up a *chain*, and a queueing system is denoted a *node* in a *queueing network* which will be dealt with in Chap. 12.

In classical queueing systems we have *non-sharing*. A customer is either waiting or being served. When being served it has a server alone. In this chapter we consider *sharing* strategies. Customers share the available capacity with other customers so that no one is waiting. All customers are always served with some rate, which may be smaller than the requested rate. We obtain models which are reversible and insensitive to service time distribution.

In Sec. 11.2 we deal with a single server which is offered more traffic streams. By requesting reversibility and usage of all servers whenever possible we get the processor sharing (*PS*) strategy. In Sec. 11.3 we consider the same system with more than one server. By requesting reversibility and usage of all servers whenever possible we get the generalized processor sharing (*GPS*) model. Finally, in Sec. 11.4 we generalize *GPS* to include multi-rate traffic and obtain a very general and robust model. For all three models we derive algorithms for accurate and effective calculation of performance measures.
Finally, in Sec. 11.5 we consider finite number of sources and other possible generalizations.

11.1 Introduction

In this chapter customers in some way share the available capacity, and therefore they are served all the time. But they may obtain less capacity than requested, which results in an increase of sojourn time. The *sojourn time* is not split up into separate waiting time and service time as in previous chapters. In this chapter and next chapter on queueing networks we use the definitions:

$\overline{W}$ = *waiting time is defined as the total sojourn time, including the service time.*

$\overline{L}$ = *queue length is defined as total number of customers (being served & waiting).*

As an example we may think of the time required to transfer a file in the Internet. If the available bandwidth is at least equal to the bandwidth requested, then the mean service time s_j for a customer of type j is defined as the mean transfer (sojourn) time. If the available bandwidth is less than the bandwidth requested, then the mean transfer time $\overline{W}_j$ will be bigger than s_j, and the increase

$$W_j = \overline{W}_j - s_j \,, \tag{11.1}$$

is defined as the *mean virtual waiting time*. We shall introduce the virtual waiting time as the increase in service time due to limited capacity. In a similarly way we define the *mean virtual queue length* as

$$L_j = \overline{L}_j - A_j \,, \tag{11.2}$$

where A_j is the offered traffic of type j.

The systems considered in this chapter are reversible, but they do not have product form. In Sec. 11.2 we consider single server systems with multiple services. The derivations are very simple and worked out in details for two services, and then generalized to more services. In Sec. 11.3 we consider systems with more servers and multiple services. As in Sec. 7.3.3 we choose a *Basic Bandwidth Unit* (*BBU*) and split the available bandwidth into n BBUs. The BBU is a generic term for channel, slot, server, etc. The smaller the basic bandwidth unit, i.e. the finer the granularity, the more accurate we may model the traffic of different services, but the bigger the state space becomes. We use the *BBU*-concept to specify the requested bandwidth. The bandwidth obtained is a continuous variable less than or equal to the requested bandwidth. In service-integrated systems the bandwidth requested depends on the type of service. The following approach is new and very simple. It allows for very general results, including all classical Markovian loss and delay models, and it is applicable to digital broadband systems, for example Internet traffic.

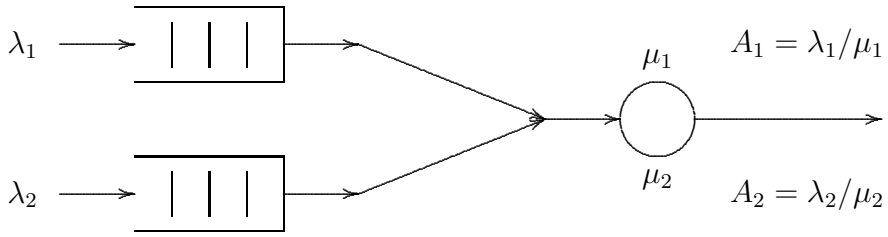


Figure 11.1: *An $\Sigma_{j=1}^{2}\, M_j/M_j/1$–queueing system with two classes of customers.*

11.2 Reversible multi-chain single-server systems

In Fig. 11.1 we consider a single-server queueing system with $N = 2$ streams of customers, i.e. two chains. Customers belonging to chain j arrive to the node according to a Poisson arrival process with intensity λ_j $(j = 1, 2)$. State $[x_1, x_2]$ is defined as a state with x_1 chain 1 customers and x_2 chain 2 customers. As each customer requests one channel the state denotes the number of requested channels. The global state is:

$$x = \sum_{j=1}^{N} x_j \,. \tag{11.3}$$

By the notation $\Sigma_{j=1}^{N}\mathrm{M}_j/\mathrm{M}_j/1$ we indicate that we have N different *PCT-1* arrival processes (chains) with individual values of arrival rates and mean service times. In the following we use index i for the state space and index j for the service (traffic stream).

If the number of servers is infinite, then we get the state transition diagram shown in Fig. 11.2, and state probabilities will be given by (7.13). However, the capacity is limited to one server, so somehow we have to reduce the service rates in all states $(x > 1)$ where more than one server is requested.

11.2.1 Reduction factors for single-server

So far we have only one server $(n = 1)$ which is shared by all customers. In state $[x_1, x_2]$ we reduce the service rate of chain-1 customers by a factor $g_1(x_1, x_2)$, $\{0 < g_1(x_1, x_2) \leq 1\}$, so that the customer don't get x_1 servers, but only $x_1 \cdot g_1(x_1, x_2)$ servers, i.e. only a fraction of the single server if there is more than one customer. The service rate is reduced from the requested service rate $\{x_j\, \mu_j\}$ to $\{g_1(x_1, x_2) \cdot x_j\, \mu_j\}$. In a similar way the service rate of chain-2 customers is reduced by a factor $g_2(x_1, x_2)$. The resulting state transition diagram is shown in Fig. 11.3. The aim is to obtain a reversible multi-dimensional system. For $x_1 + x_2 \leq n$ the system is similar to the models in Chap. 7. For $x_1 + x_2 \geq n$ we construct a reversible system using all n servers. The reduction factors $g_i(x_1, x_2)$ can be specified for various parts of the state transition diagram as follows.

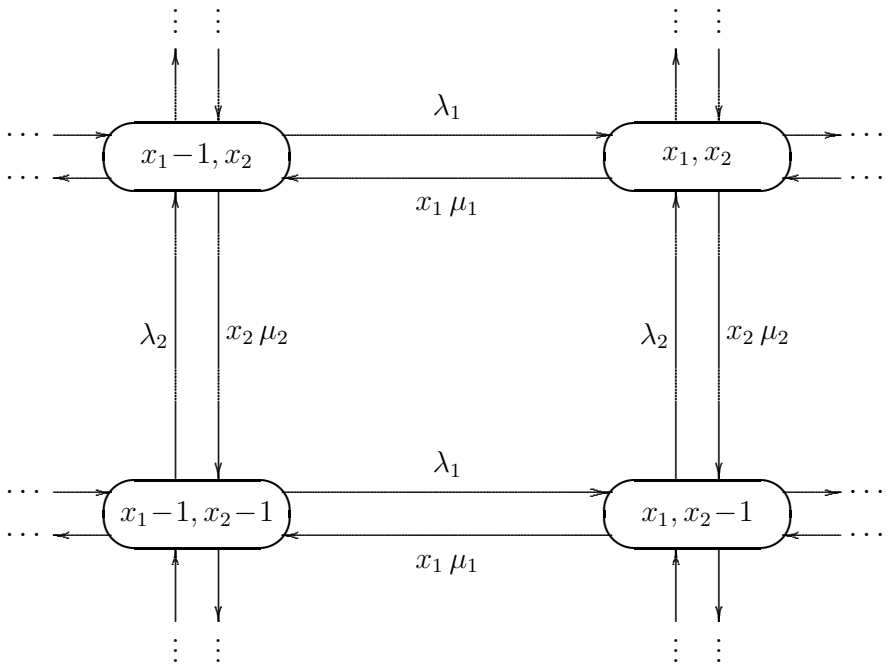


Figure 11.2: *State transition diagram for the system in Fig. 11.1 with two classes (chains) of customers and infinite number of servers (cf. Example. 7.1.1).*

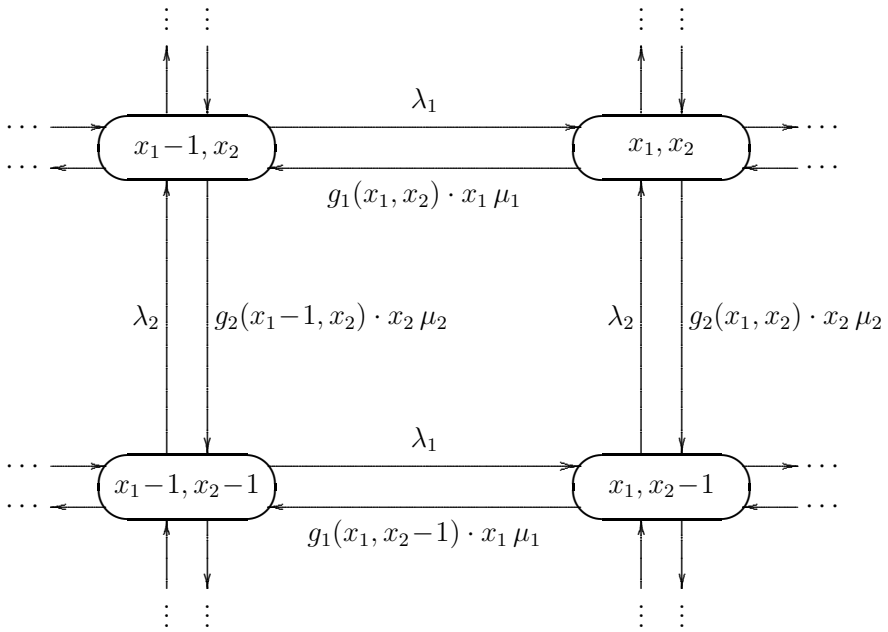


Figure 11.3: *State transition diagram for the system in Fig. 11.1 with two types (chains) of customers and a single server. In state $[x_1, x_2]$ the requested service rate $\{x_j\,\mu_j\}$ for type j is reduced by a factor $g_j(x_1, x_2)$, $(j = 1, 2)$, as compared with Fig. 11.2. As for example $g_2(x_1 - 1, x_2)$ and $g_2(x_1, x_2)$ in general are different, the system does not have product form.*

1. Non-feasible states: $x_1 < 0$ and/or $x_2 < 0$:

$$g_j(x_1, x_2) = 0\,, \quad j = 1, 2. \tag{11.4}$$

The reduction factors are undefined for these states which have probability zero. By choosing the value zero, the recursion formulæ derived below (11.10, 11.11) are correctly initiated.

2. States with demand less than capacity: $\{x_j \geq 0,\ j = 1, 2\}$ and $\{0 < x_1 + x_2 \leq 1\}$:

$$g_j(x_1 + x_2) = 1\,, \quad j = 1, 2. \tag{11.5}$$

Every call get the capacity required and there is no reduction of service rates.

3. States with only one service:

$$x_2 = 0 \text{ and } x_1 \geq 1: \quad g_1(x_1, 0) = 1/x_1\,, \quad x_1 \geq 1\,, \tag{11.6}$$

$$x_1 = 0 \text{ and } x_2 \geq 1: \quad g_2(0, x_2) = 1/x_2\,, \quad x_2 \geq 1\,. \tag{11.7}$$

Along the axes we have a classical *M/M/1* system with only one type of customers, and we assume all customers share the capacity equally as they all are identical. The state transition diagram is the same as for *M/M/1–PS* (*PS* = Processor sharing).

4. States with demand bigger than capacity: $\{x_j > 0,\ j = 1, 2\}$ and $\{x_1 + x_2 > 1\}$. These are states with both types of customers in total requiring more servers than being available. If possible, we want to choose $g_j(x_1, x_2)$ so that the following two properties are fulfilled:

   - Flow balance: The state transition diagram is constructed to be reversible: We consider four states including $[x_1, x_2]$ and neighboring states below (Fig. 11.3):

   $$\{[x_1 - 1, x_2 - 1],\ [x_1, x_2 - 1],\ [x_1, x_2],\ [x_1 - 1, x_2]\}\,.$$

   By applying the Kolmogorov cycle requirement for reversibility (Sec. 7.2), we get after canceling out the arrival and service rates (Fig. 11.3):

   $$g_2(x_1, x_2) \cdot g_1(x_1, x_2 - 1) = g_1(x_1, x_2) \cdot g_2(x_1 - 1, x_2)\,. \tag{11.8}$$

   - Normalization: All capacity is used. This requirement implies for $n = 1$ server:

   $$x_1 \cdot g_1(x_1, x_2) + x_2 \cdot g_2(x_1, x_2) = 1\,, \qquad x_1 + x_2 \geq 1\,. \tag{11.9}$$

   In state $[x_1, x_2]$ we would like to use $x_1 + x_2$ servers, but this is reduced to one server by the reduction factors.

We have two independent equations (11.8) (11.9) with two unknown reduction factors. Assume that we know the reduction factors $g_1(x_1, x_2 - 1)$ and $g_2(x_1 - 1, x_2)$, then we are able to

find a unique solution for the reduction factors $g_1(x_1, x_2)$ and $g_2(x_1, x_2)$ (Fig. 11.3). Solving the equations, we get:

$$\begin{aligned} g_1(x_1, x_2) &= \frac{1 \cdot g_1(x_1, x_2 - 1)}{x_1 \cdot g_1(x_1, x_2 - 1) + x_2 \cdot g_2(x_1 - 1, x_2)} \\ &= \frac{1}{x_1 + x_2 \cdot \frac{g_2(x_1-1,x_2)}{g_1(x_1,x_2-1)}}\,, \end{aligned} \tag{11.10}$$

$$\begin{aligned} g_2(x_1, x_2) &= \frac{1 \cdot g_2(x_1 - 1, x_2)}{x_1 \cdot g_1(x_1, x_2 - 1) + x_2 \cdot g_2(x_1 - 1, x_2)} \\ &= \frac{1}{x_1 \cdot \frac{g_1(x_1,x_2-1)}{g_2(x_1-1,x_2)} + x_2}\,. \end{aligned} \tag{11.11}$$

From the initial values specified above (11.4–11.7), we may by these recursion formulæ calculate all reduction factors. From $g_1(1, 0)$ and $g_2(0, 1)$ we calculate $g_1(1, 1)$ and $g_2(1, 1)$. Then we may calculate $g_1(2, 1)$ and $g_2(2, 1)$, and in this way we horizontally calculate all reduction factors for $g_1(x_1, 1)$ and $g_2(x_1, 1)$. From these we may then calculate all $g_1(x_1, 2)$ and $g_2(x_1, 2)$ reduction factors, and so on. Alternatively, we may use the recursion vertically or diagonally. We notice that the reduction factors are independent of the traffic parameters.

Using the known initial values we find a simple unique solution:

$$g_j(x_1, x_2) = \frac{1}{x_1 + x_2}\,, \qquad x_1 + x_2 \geq 1\,, \quad j = 1, 2\,. \tag{11.12}$$

Thus the two chains (services) are reduced by the same factor, and all customers share the capacity equally. The reversible state transition diagram is shown in Fig. 11.4.

It is easy to extend the above derivations of reduction factors to a system with N traffic streams. The state of the system is given by :

$$\underline{x} = (x_1, x_2, \ldots, x_{j-1}, x_j, x_{j+1}, \ldots, x_N)\,. \tag{11.13}$$

where x_i denotes number of channels occupied by stream i, which for single-rate traffic is equal to number of connections. For states $\{\sum_{j=1}^{N} x_j > n,\ x_j \geq 0\}$ we get for $n = 1$ server a simple unique expression for the reduction factors, which is a generalization of (11.12):

$$g_j(\underline{x}) = \frac{1}{\sum_{j=1}^{N} x_j}\,, \qquad j = 1, 2, \ldots, N\,. \tag{11.14}$$

Thus all customers share the single server equally. In the following section we show this unique state transition diagram can be interpreted as corresponding to various queueing strategies.

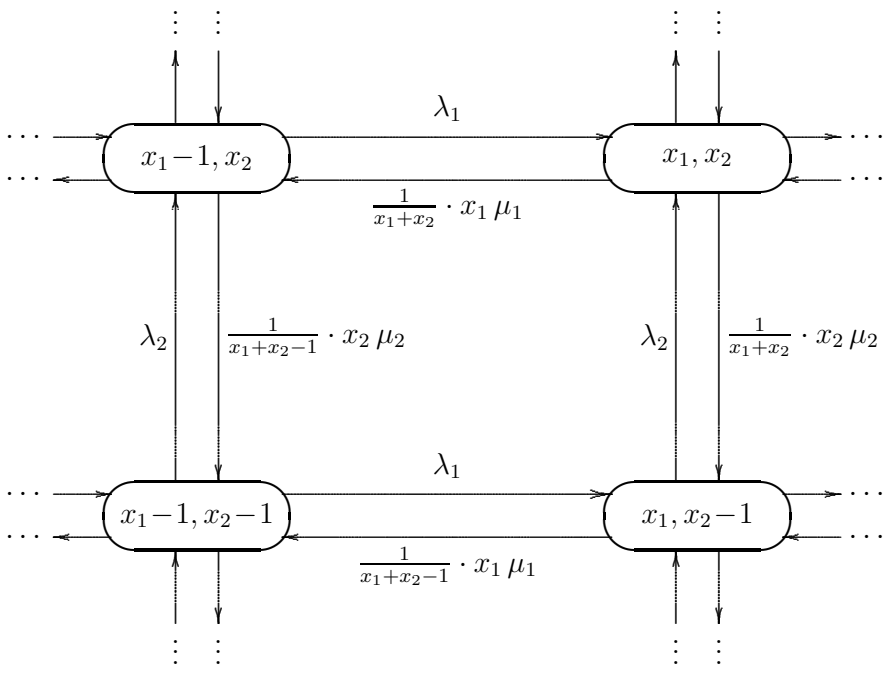


Figure 11.4: *State transition diagram for a multi–dimensional single-server system which is reversible. The system does not have product form.*

11.2.2 Single-server Processor Sharing = PS

The above result corresponds to a Processor Sharing (*PS*) system (Sec. 10.7). All $(x_1 + x_2)$ customers share the server equally and the capacity of the system is constant (one server). The total service rate μ_{x_1,x_2} in state $[x_1, x_2]$ becomes (Fig. 11.4):

$$\mu_{x_1,x_2} = \frac{x_1\,\mu_1}{x_1 + x_2} + \frac{x_2\,\mu_2}{x_1 + x_2} = \frac{x_1\,\mu_1 + x_2\,\mu_2}{x_1 + x_2}\,. \tag{11.15}$$

The total service rate is state-dependent when classes of customers have different service rates. The number of customers served per time unit depends on the mix the customers currently being served. For a system with N traffic streams the total service intensity in state $\underline{x}$ is:

$$\mu_{\underline{x}} = \frac{\sum_{j=1}^{N} x_j\,\mu_j}{\sum_{j=1}^{N} x_j}\,, \quad j = 1, 2, \ldots, N\,. \tag{11.16}$$

This model is reversible and valid for individual arbitrary service times distributions, and the system will be insensitive to the service time distributions. This property is called the "*magic property*" of processor sharing and was originally dealt with by Kleinrock (1964 [75]). In Sec. 10.7 we had only one type of customers and a one-dimensional state transition diagram. Now we have N types of customers, and to define the state of the system in a unique way we need an N-dimensional state transition diagram.

Theorem 11.1 *$\Sigma_j\, M_j/G_j/1$–PS single-server system with processor sharing (PS) is reversible and insensitive to the service time distributions. Each class may have individual mean service times.*

11.2.3 Non-sharing single-server

Let us assume that the server is occupied by one customer at a time, i.e. there is no sharing of the capacity. Then for Poisson arrival processes and classical queueing systems with queueing disciplines as for example *FCFS, LCFS, SIRO* the customer being served in state $\underline{x}$, i.e. the next customer departing, will be a random one of the $x = \sum_j x_j$ customers residing in the system.

From the state transition diagram for two services (Fig 11.4) we see that the customer being served is of type-1, respectively type-2, with the following probabilities:

$$p\{\text{type-1 served}\} = \frac{\frac{x_1\,\mu_1}{x_1+x_2}}{\frac{x_1\,\mu_1}{x_1+x_2}+\frac{x_2\,\mu_2}{x_1+x_2}} = \frac{x_1\,\mu_1}{x_1\,\mu_1+x_2\,\mu_2}\,, \tag{11.17}$$

$$p\{\text{type-2 served}\} = \frac{\frac{x_2\,\mu_2}{x_1+x_2}}{\frac{x_1\,\mu_1}{x_1+x_2}+\frac{x_2\,\mu_2}{x_1+x_2}} = \frac{x_2\,\mu_2}{x_1\,\mu_1+x_2\,\mu_2}\,. \tag{11.18}$$

We notice that this is only a random one of the $x_1 + x_2$ customers when $\mu_1 = \mu_2$. Thus the two classes must have the same mean service time for the state transition diagram to describe an *M/M/1* non-sharing system. In all other cases when $\{\mu_1 \neq \mu_2\}$, the customer being served will not be a random one among the $\{x_1 + x_2\}$ customers in the system. It is also obvious that the system is only reversible when the service times are exponentially distributed, as the distribution of inter-departure times during saturation periods will be equal to the service time distribution. This interpretation corresponds to a classical *M/M/1* system with total arrival rate $\lambda = \lambda_1 + \lambda_2$ and mean service time $\mu^{-1} = \mu_1^{-1} = \mu_2^{-1}$.

By superposition of Poisson processes it is obvious that this is also valid for N traffic streams. We thus have:

Theorem 11.2 *The non-sharing $\Sigma_j\, M_j/M/1$ system (FCFS, LCFS, SIRO) is only reversible if all customers have exponentially distributed service times with same mean service time.*

11.2.4 Single-server LCFS-PR

The state transition diagram in Fig. 11.4 can also be interpreted as the state transition diagram of an $\Sigma_j\, M_j/G_j/1$–*LCFS–PR* (preemptive resume, non-sharing) system. It is obvious

that this system is reversible because the process follows exactly the same path in the state transition diagram away from state zero due to arriving customers as back towards state zero due to departing customers. Thus we always have local balance. The latest arriving customer in state $[x_1, x_2]$ belongs with probability $(x_j/(x_1 + x_2))$ to class j $(j = 1, 2)$. This is valid for any number N of services.

Theorem 11.3 *Σ_j $M_j/G_j/1$–LCFS-PR single-server system with LCFS-PR is reversible and insensitive to the service time distributions, and the services may have individual mean values.*

11.2.5 Summary of reversible single server

The multi-dimensional state transition diagram for single-server systems with N different services can be interpretated in similar way as for two services. In conclusion, for a single-server queueing systems with N classes of customers to be reversible, the state transition diagram must be as shown in Fig. 11.4 in N dimensions. For this diagram we have the following interpretations:

- $\Sigma_{j=1}^{N}$ $M_j/G_j/1$–PS,
- $\Sigma_{j=1}^{N}$ $M_j/M/1$ non-sharing with same exponential service time for all customers,
- $\Sigma_{j=1}^{N}$ $M_j/G_j/1$–LCFS–PR (non-sharing).

These systems are also called symmetric queueing systems. Reversibility implies that the departure processes of all classes are identical with the arrival processes. In principle we may introduce new interpretations of the state transition diagram. Due to reversibility, the departure process will be of same type as the arrival process for each chain, i.e. also Poisson processes as the arrival processes. Of course, this is also valid for a system with one type of customers (Sec. 10.7).

11.2.6 State probabilities for multi-services single-server

All three single-server systems mentioned above are interpretations of the same state transition diagram. Thus they have the same state probabilities and mean performance measures. Part of the state transition diagram for two services is shown in Fig. 11.4. The diagram is reversible, since flow clockwise equals flow counter-clockwise. Hence, there is *local balance*. All state probabilities can be expressed by state zero. For two services we find:

$$p(x_1, x_2) = p(0, 0) \cdot \frac{A_1^{x_1}}{x_1!} \cdot \frac{A_2^{x_2}}{x_2!} \cdot (x_1 + x_2)! \tag{11.19}$$

In comparison with the multi-dimensional Erlang–B formula (7.10) we now have the additional factor $(x_1+x_2)!$. The product form between classes is lost because the state probability cannot be written as the product of state probabilities of two independent systems:

$$p(x_1, x_2) \neq p_1(x_1) \cdot p_2(x_2)\,.$$

This lack of product form will complicate the evaluation of queueing networks as the detailed state space of a node becomes very large and cannot be expressed in a compact way. We find $p(0,0)$ by normalization:

$$\sum_{x_1=0}^{\infty} \sum_{x_2=0}^{\infty} p(x_1, x_2) = 1\,.$$

Using the Binomial expansion we get a simple expression for global state probabilities:

$$p(x_1 + x_2 = x) \;=\; p(0,0) \cdot (A_1 + A_2)^x \tag{11.20}$$

$$\;=\; (1 - A) \cdot A^x\,, \tag{11.21}$$

where $A = A_1 + A_2$. State probability $p(0,0) = 1 - A$ is known explicitly without need of normalization. This is identical with the state probabilities of *M/M/1* with the offered traffic $A = A_1+A_2$ (9.30).

If there are N different traffic streams, the state probabilities become:

$$p(x_1, x_2, \ldots, x_N) \;=\; p(\underline{0}) \cdot \frac{A_1^{x_1}}{x_1!} \frac{A_2^{x_2}}{x_2!} \cdots \frac{A_N^{x_N}}{x_N!} \cdot (x_1 + x_2 + \ldots + x_N)\,! \tag{11.22}$$

$$p(\underline{x}) \;=\; p(\underline{0}) \cdot \left\{ \prod_{j=1}^{N} A_j^{x_j} \right\} \cdot \frac{\left\{ \sum_{j=1}^{N} x_j \right\}!}{\left\{ \prod_{j=1}^{N} x_j! \right\}}\,, \tag{11.23}$$

where $p(\underline{x}) = p(x_1, x_2, \ldots, x_N)$. This can be expressed by the multinomial distribution (2.97):

$$p(\underline{x}) = p(\underline{0}) \cdot \left\{ \prod_{j=1}^{N} A_j^{x_j} \right\} \cdot \binom{x_1 + x_2 + \cdots + x_N}{x_1, x_2, \ldots, x_N}\,. \tag{11.24}$$

For an unlimited number of queueing positions the global state probabilities of the total number of customers becomes:

$$p(x) = p\{x_1 + x_2 + \cdots + x_N = x\}\,, \quad x = 0, 1, 2, \ldots\,.$$

By the multinomial expansion we observe that the global state probabilities are identical with state probabilities of the *M/M/1* system:

$$p(x) \;=\; p(0) \cdot (A_1 + A_2 + \cdots + A_N)^x \tag{11.25}$$

$$\;=\; (1 - A) \cdot A^x\,, \,, \quad \text{where} \tag{11.26}$$

$$A \;=\; A_1 + A_2 + \cdots + A_N\,. \tag{11.27}$$

11.2.7 Generalized algorithm for state probabilities

To evaluate the global state probabilities we may use the following trivial algorithm for N traffic classes. In general, we first find the relative non-normalized state probabilities $q(x)$ where we typically put $q(0)$ equal to one. In this case ($n = 1$) the normalization of global states are very simple as we have $p(0) = 1 - A$.

$$p(x) \;=\; \begin{cases} 0\,, & x < 0\,, \\ 1 - A\,, & x = 0\,, \\ \displaystyle\sum_{j=1}^{N} p_j(x)\,, & x = 1, 2, \ldots\,, \end{cases} \tag{11.28}$$

where

$$p_j(x) \;=\; \begin{cases} 0\,, & x < 1\,, \\ A_j \cdot p(x-1)\,, & x = 1, 2, \ldots\,, \end{cases} \tag{11.29}$$

This algorithm is very simple, but below we find for more complex models in a similar way more ingenious algorithms.

11.2.8 Performance measures

We deal with more types of customers as described in Sec. 10.6.1 The mean queue length $\overline{L}$, which includes all customers in the system (partly being served, partly waiting), becomes as for *M/M/1*. This is a geometric distribution (11.26) with mean value (cf. caption to Table 3.1):

$$\overline{L} = \frac{A}{1 - A}\,,$$

where total offered traffic is $A = (A_1 + A_2 + \cdots + A_N)$. In state x the average number of type-j calls is $x \cdot p_j(x)$. The mean queue length for stream j (including all customers) becomes:

$$\overline{L}_j \;=\; \sum_{x=0}^{\infty} x \cdot p_j(x) = \frac{A_j}{A} \cdot \overline{L}\,,$$

$$\frac{\overline{L}_j}{A_j} \;=\; \frac{\overline{L}}{A} \qquad \text{where} \qquad \overline{L} = \sum_{j=1}^{N} \overline{L}_j\,. \tag{11.30}$$

The mean queue length $\overline{L}$ includes all customers waiting & being served. As carried traffic is equal to offered traffic, the increase in $\overline{L}$ due to limited capacity is $L = \overline{L} - A$ which is

equal to (9.31). For stream j the increase is $L_j = \overline{L}_j - A_j$. The mean sojourn time for type j customers becomes by Little's law:

$$\overline{W}_j = \frac{\overline{L}_j}{\lambda_j} = s_j \cdot \frac{\overline{L}}{A}\,,$$

or

$$\frac{\overline{W}_j}{s_j} = \frac{\overline{L}}{A} = \frac{\overline{W}}{s}\,, \tag{11.31}$$

where the overall mean sojourn time is:

$$W = \sum_{j=1}^{N} \frac{\lambda_j}{\lambda} \cdot W_j\,. \tag{11.32}$$

The global mean service time is obtained in the same way (10.54).

In similar way we have for mean sojourn times $W = \overline{W} - s$ and $W_j = \overline{W}_j - s_j$. Subtracting one from both sides of (11.30) and (11.31) we get:

$$\frac{L_j}{A_j} = \frac{W_j}{s_j} = \frac{L}{A} = \frac{W}{s} = \textit{constant}. \tag{11.33}$$

L_j and W_J corresponds to the the usual definition of waiting times in non-sharing queueing systems. For a given stream the mean waiting time is proportional to the mean service time of this stream. This is an important property of processor sharing.

11.3 Reversible multi–chain/server systems

We now consider a system with n servers and infinite queue. All customers request one server (*BBU*, channel), but may obtain less when the total demand is bigger than the capacity. The state of the system is defined by

$$\underline{x} = (x_1, x_2, \ldots, x_j, \ldots, x_N)\,, \quad x = \sum_{j=1}^{N} x_j\,,$$

where x_j is the number of type j customers in the system and x is the global state. Customers of type j arrive according to a Poisson arrival process with intensity λ_j, and the service time is exponentially distributed with intensity μ_j (mean value $1/\mu_j$) ($j = 1, 2, \ldots, N$). If the number of servers were infinite, then we would get the state transition diagram shown in Fig. 11.2. However, the capacity is limited to n servers, so we have to reduce service rates in all states requiring more than n servers (saturation). In the following we deal with the general case with N different services. The approach is the same as for the above single-server system.

11.3.1 Reduction factors for multi-server

In state $\underline{x} = (x_1, x_2, \ldots, x_j, \ldots, x_N)$ the requested service rate $\{x_j \cdot \mu_j\}$ for type j customers is reduced by a factor $g_j(\underline{x})$ $(0 < g_j(\underline{x}) \leq 1)$. The reduction factors are chosen so that we maintain reversibility and always utilize all n servers when needed. They can be specified for various parts of the state transition diagram as follows.

1. Non-feasible states: $x_j \leq 0$ for at least one value $j \in \{1, 2, \ldots, N\}$:

$$g_j(\underline{x}) = 0\,, \quad j = 1, 2, \ldots, N\,. \tag{11.34}$$

   The reduction factors are undefined for these states which have probability zero. By choosing the value zero, the recursion formula derived below (11.40) is initiated in a correct way.

2. Feasible states with demand less than capacity: $\{x_j \geq 0\ \forall\, j\}$ and $\{0 \leq \sum_{j=1}^{N} x_j \leq n\}$:

$$g_j(\underline{x}) = 1\,, \quad j = 1, 2, \ldots, N\,. \tag{11.35}$$

   Every connection (customer) obtains the capacity requested (one channel), and there is no reduction of the requested service rate.

3. States with only one type of customers: $\{x_i = 0\ \forall\ i \neq j\}$ and $\{x_j \geq n\}$:

$$g_j(\underline{x}) = \frac{n}{x_j}\,. \tag{11.36}$$

   Along the axes we have a classical $M/M/n$–system with only one type of service, and we assume that the calls share the capacity equally as they all are identical.

4. States with more types of customers (say $x_j > 0, x_k > 0$) which in total require more than n channels:

$$x_j \geq 0\ \forall\ j \quad \text{and} \quad x = \sum_{j=1}^{N} x_j > n\,.$$

   - Flow balance: the state transition diagram is required to be reversible:
     We consider four states in a square below state $(x_1, \ldots, x_j, \ldots, x_k, \ldots x_N)$ keeping number of connections constant except for services j and k (Fig.11.3):

$$(x_1, \ldots, x_j - 1, \ldots, x_k, \ldots x_N) \qquad (x_1, \ldots, x_j, \ldots, x_k, \ldots x_N)$$

$$(x_1, \ldots, x_j - 1, \ldots, x_k - 1, \ldots x_N) \qquad (x_1, \ldots, x_1, \ldots, x_k - 1, \ldots x_N)$$

     We introduce the notation:

$$\underline{x} - 1_j = \{x_1, x_2, \ldots x_{j-1}, x_j - 1, x_{j+1} \ldots x_N\}\,. \tag{11.37}$$

A necessary and sufficient condition for reversibility (Kingman 1969) is that all two-dimensional flow paths are in local balance. In total we may choose j and k in so many ways:

$$\binom{N}{2} = \frac{N\,(N-1)}{2}\,.$$

We apply Kolmogorov cycles requirement for reversibility (Sec. 7.2) for any pair of services for the two-dimensional planes $\{1,j\},(j=2,3,\ldots,N)$, which gives us $N-1$ independent equations. After reduction we get the following flow balance equations:

$$\begin{aligned} g_1(\underline{x})\cdot g_j(\underline{x}-1_1) &= g_j(\underline{x})\cdot g_1(\underline{x}-1_j)\,, \\ g_j(\underline{x}) &= g_1(\underline{x})\cdot \frac{g_j(\underline{x}-1_1)}{g_1(\underline{x}-1_j)} \qquad j=1,2,\ldots,N\,. \end{aligned} \tag{11.38}$$

We assume that we know the reduction factors for states $\underline{x}-1_j$ below state $\underline{x}$, and we want to find the reduction factors in state $\underline{x}$. To find the N reduction factors in state $\underline{x}=\{x_1,x_2,\ldots,x_N\}$ we need only N independent equations. Above we have $N-1$ equations, and below we find $g_1(\underline{x})$.

- Normalization:
  We obtain one equation more by requiring that the total capacity n must be used in all global states $x \geq n$:

$$\begin{aligned} n &= \sum_{i=1}^{N} \{x_i \cdot g_i(\underline{x})\} \\ &= \sum_{i=1}^{N} \left\{ x_i \cdot g_1(\underline{x}) \cdot \frac{g_i(\underline{x}-1_1)}{g_1(\underline{x}-1_i)} \right\}\,. \end{aligned} \tag{11.39}$$

$$g_1(\underline{x}) = \frac{n}{\displaystyle\sum_{i=1}^{N} \left\{ x_i \cdot \frac{g_i(\underline{x}-1_1)}{g_1(\underline{x}-1_i)} \right\}}\,, \tag{11.40}$$

From (11.38) we then find all other reduction factors.

We know reduction factors for all global states x up to and including global state n, i.e. states where $\{x=\sum_{i=1}^{N} x_i \leq n\}$ (11.35). We also know all reduction factors for states where only one type of customer is present (11.36). Recursively we can then calculate all other reduction factors. Knowing the reduction factors we find the relative state probabilities, and finally by normalization the detailed state probabilities. This is equivalent to calculation of all relative state probabilities, and then by global normalization obtain the detailed state probabilities.

As seen above, the reduction factors are independent of the traffic processes, and the approach includes *BPP*–traffic, and any state-dependent Poisson arrival process.

Using the above initial values it can easily be shown that we get the following unique solution:

$$g_j(\underline{x}) = \begin{cases} 1 & 0 \le x \le n\,, \\ \dfrac{n}{x} & n \le x\,, \end{cases} \qquad j = 1, 2, \ldots, N\,, \tag{11.41}$$

$$\text{where} \quad x = \sum_{j=1}^{N} x_j\,, \quad x_j \ge 0\,.$$

Theoretically it is known that reversible Markov chains has one and only one solution. Thus during overload all customers are reduced by the same factor, and the customers share the capacity equally. In Fig. 11.5 we consider a multi-server queueing system with $N = 2$ traffic streams (*chains*). We notice that the diagram is reversible. In the following section we show this unique state transition diagram may be interpreted as corresponding to different classical queueing strategies.

11.3.2 Generalized processor sharing = GPS

The state transition diagram in Fig. 11.5 can be interpreted in the following way. In states $[x_1, x_2]$ below saturation ($x_1 + x_2 \le n$) every user occupy one server. Above saturation all users share the available capacity equally. The state transition diagram Fig. 11.5 is reversible. The state probabilities are insensitive to the service time distribution and each service may have individual mean service time. This model is called the *GPS* (*Generalized Processor Sharing*) model. For state $x_1 + x_2 > n$, traffic stream one wants a total service rate $x_1 \cdot \mu_1$, and traffic stream two wants a service rate $x_2 \cdot \mu_2$. But the service rate of both streams are reduced by the same factor $n/(x_1 + x_2)$.

Theorem 11.4 *$\Sigma_{j=1}^{N}$ $M_j/G_j/n$–GPS multi-server system with generalized processor sharing (GPS) is reversible and insensitive to the service time distributions. Each class may have individual mean service time.*

11.3.3 Non-sharing multi–chain/server

We consider $M/M/n$–non-sharing systems. A customer being served always has one server by itself. A customer is either waiting or being served. To maintain reversibility for $x_1 + x_2 > n$ we have to require that all services have the same mean service time $\mu^{-1} = \mu_j^{-1}$, which furthermore must be exponentially distributed. Otherwise, the next departing customer will not be a random one among the customers in the system (Fig. 11.5). Thus this system is sensitive to the service time distribution. The proof is the same as for the single server case in Sec. 11.2.3. This corresponds to an $M/M/n$ system with total arrival rate $\lambda = \sum_j \lambda_j$ and

service rate μ. The state probabilities are given by (9.2) and (9.4), and the state transition diagram is reversible. The system $M/M/\infty$ may be considered as a special case of $M/M/n$ and this has already been dealt with in connection with classical delay systems (Chap. 9).

Theorem 11.5 *The $\Sigma_{j=1}^{N} M_j/M/n$ system (FCFS, LCFS, SIRO) is only reversible if all customers have the same mean service time, and this service time must be exponentially distributed.*

11.3.4 Symmetric queueing systems

For multiple servers the non-sharing system $\Sigma_{j=1}^{N} M_j/G_j/n$–*LCFS-PR* will in general not be reversible, because the latest arriving customer may not be the first to finish service as there are more servers working in parallel. If all streams have same mean holding time this system is included in Theorem 11.5. Otherwise, it is only reversible for single-server systems (Sec. 11.2.4).

In conclusion, multi-server queueing systems with several classes of customers will only be reversible when the system is one of the following queueing system:

- $\Sigma_{j=1}^{N} M_j/G_j/n$–*GPS*, which includes $\Sigma_{j=1}^{N} M_j/G_j/1$–*PS*,
- $\Sigma_{j=1}^{N} M_j/M/n$–non-sharing with same exponentially distributed service time for all customers, which includes the single server system for $n = 1$.
- $\Sigma_{j=1}^{N} M_j/G_j/1$–*LCFS–PR*. This is only valid for single-server systems.

These systems are all reversible. They are also called *symmetric queueing systems*. Reversibility implies that the departure processes of all classes are Poisson processes like the arrival processes. For the classical *non-sharing M/M/n systems* we have a reversible system which is sensitive to the service time distribution.

11.3.5 State probabilities

For a node with N services and n servers we may exploit local balance and get the following detailed state probabilities:

$$\frac{p(x_1, x_2, \ldots, x_N)}{p(0, 0, \ldots 0)} = \begin{cases} \dfrac{A_1^{x_1}}{x_1!}\dfrac{A_2^{x_2}}{x_2!}\cdots\dfrac{A_N^{x_N}}{x_N!}\,, & \Sigma_{j=1}^{N} x_j \leq n\,, \\ \dfrac{A_1^{x_1}}{x_1!}\dfrac{A_2^{x_2}}{x_2!}\cdots\dfrac{A_N^{x_N}}{x_N!}\cdot\dfrac{(x_1+x_2+\ldots+x_N)!}{n!\cdot n^{(x_1+x_2+\ldots+x_N)-n}}\,, & \Sigma_{j=1}^{N} x_j \geq n\,. \end{cases} \tag{11.42}$$

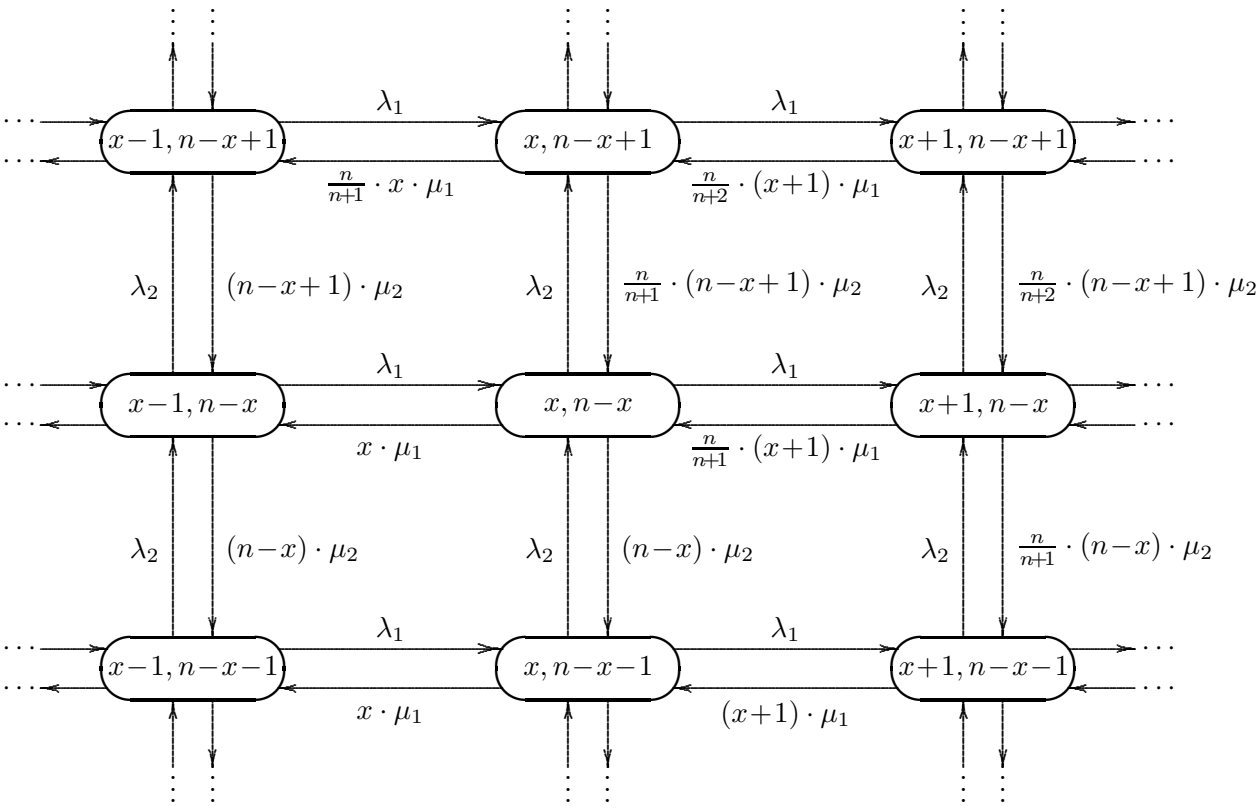


Figure 11.5: *State transition diagram for a reversible multi–dimensional $\Sigma_j M_j/M_j/n$–system. The detailed states shown, correspond to global states below and above global state n.*

State probability $p(0, 0, \ldots 0)$ is obtained by normalization. When $n = 1$ we of course get (11.23). We define the probability of global state x by:

$$p(x) = \sum_{\sum_i x_i = x} p(x_1, x_2, \ldots x_j, \ldots x_N).$$

By the multinominal theorem (2.99) we get (9.2):

$$\frac{p(x)}{p(0)} = \begin{cases} \dfrac{A^x}{x!}\,, & 0 \le i \le n\,, \\ \dfrac{A^x}{n! \cdot n^{x-n}}\,, & i \ge n\,. \end{cases} \tag{11.43}$$

11.3.6 Generalized algorithm for state probabilities

We now consider a system with n servers and N traffic streams. The relative global state probabilities are obtained by the recursion:

$$q(x) = \begin{cases} 0 & x < 0\,, \\ 1 & x = 0\,, \\ \sum_{j=1}^{N} q_j(x) & x = 1, 2, \ldots, \infty\,, \end{cases} \tag{11.44}$$

where

$$q_j(x) = \begin{cases} \frac{A_j}{x} \cdot q(x-1) & x \leq n\,, \\ \frac{A_j}{n} \cdot q(x-1) & x > n\,. \end{cases} \tag{11.45}$$

Here $q_j(x)$ is the contribution of stream j to the relative global state probability $q(x)$:

$$q_j(x) = \sum_{\sum x_j = x} \frac{x_j}{x} \cdot q(x_1, x_2, \ldots x_j, \ldots x_N) \tag{11.46}$$

State probability $p(0)$ is obtained by normalization:

$$Q = \sum_{i=0}^{\infty} q(i) = \sum_{i=0}^{\infty} \sum_{j=1}^{N} q_j(i)\,. \tag{11.47}$$

By normalizing all relative state probabilities $q_j(x)$ and $q(x)$ by Q we obtain the true state probabilities $p_j(x)$ and $p(x)$. To get a numerical robust and accurate algorithm, the normalization should be carried out in each step (increase of x) as described in Sec. 4.4.1. The algorithm is an extension of the generalized algorithm in Sec. 7.6.2 for single-slot Poisson traffic in loss systems. We derive the algorithm for the more general case with multi-rate traffic in Sec. 11.4.2.

11.3.7 Performance measures

Performance measures are derived in the same way as for single-server systems in Sec. 11.2.8. The total mean queue length $\overline{L}$, which includes all customers in the system becomes as for $M/M/n$:

$$\overline{L} = \sum_{x=0}^{\infty} x \cdot p(x)$$

The contribution of type j calls in state x to the average number of customers in the system is $x \cdot p_j(x)$. The mean queue length for stream j denoted by $\overline{L}_j$ which includes all customers in the system becomes:

$$\overline{L}_j = \sum_{x=0}^{\infty} x \cdot p_j(x) = \frac{A_j}{A} \cdot \overline{L}\,, \qquad \text{or}$$

$$\frac{\overline{L}_j}{A_j} = \frac{\overline{L}}{A} \quad \text{where} \quad \overline{L} = \sum_{j=1}^{N} \overline{L}_j \quad \text{and} \quad A = \sum_{j=1}^{N} A_j\,. \tag{11.48}$$

The mean sojourn time for type j customers becomes by Little's law:

$$\overline{W}_j = \frac{\overline{L}_j}{\lambda_j} = s_j \cdot \frac{\overline{L}}{A}\,,$$

or

$$\frac{\overline{W}_j}{s_j} = \frac{\overline{L}}{A}\,. \tag{11.49}$$

The mean queue length $\overline{L}$ includes all customers partly waiting and partly being served. As the carried traffic is equal to the offered traffic, the increase in $\overline{L}$ due to limited capacity is $L = \overline{L} - A$. For stream j the increase is $L_j = \overline{L}_j - A_j$. In a similar way we get the increase in mean sojourn times due to limited capacity as: $W = \overline{W} - s$ and $W_j = \overline{W}_j - s_j$. Subtracting one on both sides of (11.48) and (11.49) we get:

$$\frac{L_j}{A_j} = \frac{W_j}{s_j} = \frac{L}{A} = \frac{W}{s} = \textit{constant}. \tag{11.50}$$

$\{L_j, L\}$ and $\{W_J, W\}$ corresponds to the the usual definitions of mean queue lengths and waiting times in non-sharing queueing systems (Chap. 9 & 10). For a given stream the mean waiting time is proportional to the mean service time of this stream. This is an important property of processor sharing.

11.4 Reversible multi–rate/chain/server systems

We now consider a queueing system with n servers which is offered N multi-rate traffic streams. We assume traffic stream j has constant arrival rate λ_j, service rate $d_j\,\mu_j$, and requires d_j simultaneous channels for full service. The state of the system can be specified either in [connections] or in [channels]. Let i_j denote the number of connections of type j, then the number of channels occupied by service j is $x_j = d_j \cdot i_j$. The state of the system is

then defined by one of the following options:

$$\begin{aligned} \underline{i} &= (i_1, i_2, \ldots, i_j, \ldots, i_N)\,, \\ \underline{x} &= (x_1, x_2, \ldots, x_j, \ldots, x_N)\,, \\ &= (i_1\, d_1, i_2\, d_2, \ldots, i_j\, d_j, \ldots, i_N\, d_N)\,, \\ x &= \sum_{j=1}^{N} x_j\,, \end{aligned}$$

where x is the global state of the system in channels. In the following *we use number of channels* because global number of connections gives no information on number of busy channels.

11.4.1 Reduction factors

If the demand is bigger than the capacity, then the service rate is reduced by a state dependent reduction factor. The reduction factors becomes becomes more complex than for systems with single-rate traffic, but they are derived in a similar way as for single-slot traffic (Sec. 11.2.1 and 11.3.1). The service rate in state $\underline{x} = (x_1, x_2, \ldots, x_j, \ldots, x_N)$ is for type j customers reduced by a factor $g_j(\underline{x})$. The reduction factors $g_j(\underline{x})$ are chosen so that we maintain reversibility, and use all capacity whenever needed. They can be specified for various parts of the state transition diagram as follows.

1. Non-feasible states: $x_j \leq 0$ for at least one value $j \in \{1, 2, \ldots, N\}$:

$$g_j(\underline{x}) = 0\,, \quad j = 1, 2, \ldots, N\,. \tag{11.51}$$

   The reduction factors are undefined for these states which have probability zero. By choosing the value zero, the recursion formula derived below (11.56) is initiated in a correct way.

2. States with demand less than capacity: $\{x_j \geq 0\ \forall\ j\}$ and $\{0 \leq x = \sum_{j=1}^{N} x_j \leq n\}$:

$$g_j(\underline{x}) = 1\,, \quad j = 1, 2, \ldots, N\,. \tag{11.52}$$

   Every customer get the requested capacity and there is no reduction of the requested service rate.

3. States with only one type of customers: $\{x_i = 0\ \forall\ i \neq j\}$ and $\{x_j \geq n\}$:

$$g_j(\underline{x}) = \frac{n}{x_j}\,, \quad j = 1, 2, \ldots, N\,. \tag{11.53}$$

   Along the axes we have a classical $M/M/n$–system with only one type of customers, and we assume that the calls share the capacity equally as they all are identical.

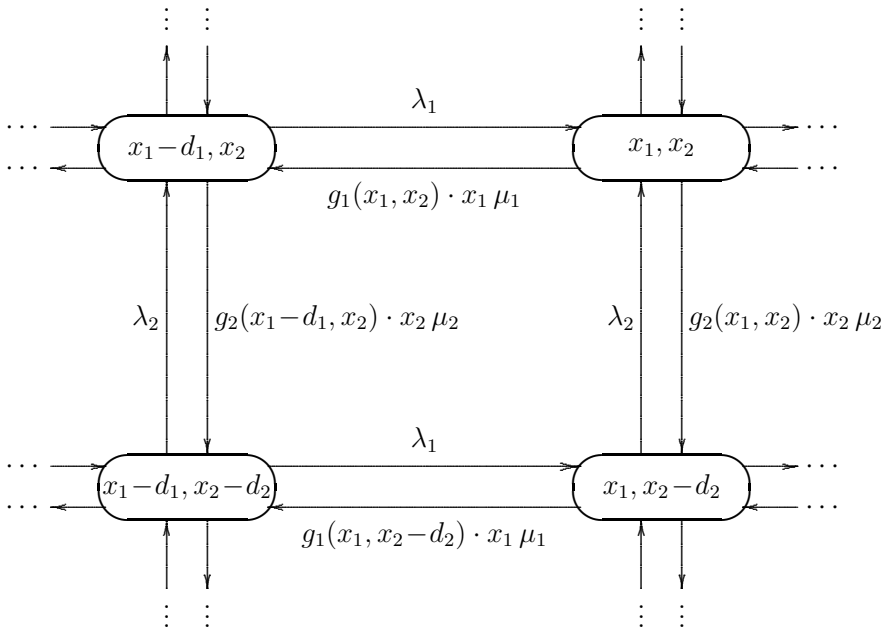


Figure 11.6: *State transition diagram for a system with two types (chains) of customers with multi-rate traffic and n servers. In state* (x_1, x_2) *the requested service rate* $x_j \cdot \mu_j$ *for type* j *is reduced by a factor* $g_j(x_1, x_2)$*. As for example* $g_2(x_1 - 1, x_2)$ *and* $g_2(x_1, x_2)$ *will be different, the system does not have product form. Note that we have chosen the service rate* $d_j\,\mu_j$ *for a* d_j-*slot call.*

4. States with more types of customers, in total requiring more than n channels (we assume that at least two values x_j and x_k are positive):

$$x_j \geq 0 \;\forall\; j \quad \text{and} \quad x = \sum_{j=1}^{N} x_j > n\,.$$

- Flow balance: The state transition diagram is required to be reversible: We consider four neighboring states in a square below state $(x_1, \ldots, x_j, \ldots, x_k, \ldots, x_N)$ keeping all other dimensions constant except for i and j (Fig.11.6):

$$(x_1, \ldots, x_j - d_j, \ldots, x_k, \ldots x_N) \qquad (x_1, \ldots, x_j, \ldots, x_k, \ldots x_N)$$

$$(x_1, \ldots, x_j - d_j, \ldots, x_k - d_k, \ldots x_N) \qquad (x_1, \ldots, x_j, \ldots, x_k - d_k, \ldots x_N)$$

A necessary and sufficient condition for reversibility (Kingman 1969) is that all two-dimensional flow paths are in local balance. In total we may choose j and k in so many ways:

$$\binom{N}{2} = \frac{N\,(N-1)}{2}\,,$$

corresponding to different Kolmogorov cycles and thus different local balance equations.

We assume that we know all reduction factors for states $\underline{x} - d_j$ below state $\underline{x}$. To find the N reduction factors in state $\underline{x} = \{x_1, x_2, \ldots, x_N\}$ we need N independent equations. We may apply Kolmogorov cycles requirements for reversibility for the two services $\{1, j\}, j = 2, 3, \ldots, N$. This yields $N - 1$ independent equations. We get the following flow balance equations:

$$\begin{aligned} g_1(\underline{x}) \cdot g_j(\underline{x} - d_1) &= g_j(\underline{x}) \cdot g_1(\underline{x} - d_j)\,, \\ g_j(\underline{x}) &= g_1(\underline{x}) \cdot \frac{g_j(\underline{x} - d_1)}{g_1(\underline{x} - d_j)}\,, \qquad j = 1, 2, \ldots N\,. \end{aligned} \tag{11.54}$$

- Normalization:
  We furthermore have the normalization equation requiring that the total capacity used is n whenever the global state $x \geq n$. The capacity normalization equation becomes:

$$\begin{aligned} n &= \sum_{i=1}^{N} \{x_i \cdot g_i(\underline{x})\} \\ &= \sum_{i=1}^{N} \left\{ x_i \cdot g_1(\underline{x}) \cdot \frac{g_i(\underline{x} - d_1)}{g_1(\underline{x} - d_i)} \right\}, \\ g_1(\underline{x}) &= \frac{n}{\sum_{i=1}^{N} \left\{ x_i \cdot \frac{g_i(\underline{x} - d_1)}{g_1(\underline{x} - d_i)} \right\}}\,. \end{aligned} \tag{11.55}$$

As we know all reduction factors up to global state n, and also all reduction factors for states with only one type of active customers, then we can recursively calculate all reduction factors. This is equivalent to calculating the relative state probabilities, and thus by global normalization the detailed state probabilities.

For two traffic streams and single-slot traffic we of course get the reduction factors given in (11.10) and (11.11). As mentioned above the reduction factors are independent of the traffic processes. The approach includes *BPP*-traffic, and any state-dependent Poisson arrival process. In Fig. 11.6 we consider a multi-server queueing system with $N = 2$ types of customers (*chains*).

11.4.2 Generalized algorithm for state probabilities

We now consider a multi-rate system with n servers and N traffic streams. The relative state probabilities are denoted by letter q. The true state probabilities are denoted by letter p and

they are obtained from q by normalization. The global state probability $q(x)$ is made up of contributions $q_j(x)$ from each traffic type:

$$q(x) \quad = \quad \begin{cases} 0 & x < 0\,, \\ 1 & x = 0\,, \\ \sum_{j=1}^{N} q_j(x) & x = 1, 2, \ldots, k\,, \end{cases} \tag{11.56}$$

where we find $q_j(x)$ by the following recursion formula derived in Sec. 11.4.3:

$$q_j(x) = \frac{1}{\min\{x, n\}} \left\{ \frac{d_j}{x} \cdot \lambda_j \cdot q(x - d_j) + \sum_{i=1}^{N} \left(\frac{x - d_i}{x} \cdot \lambda_i \cdot q_j(x - d_i) \right) \right\}. \tag{11.57}$$

This is valid for $x > 0$. Remember that the requested service rate of a type j call is $d_j\,\mu_j$, and that the number of servers obtained is less when buffers are used, i.e. when demand is bigger than capacity. The above recursion can be used to find global performance measures. To find performance measures for an individual streams j we split the contribution of stream j to state x up into two contributions:

$$q_j(x) = q_{j,y}(x) + q_{j,l}(x)\,, \qquad j = 1, 2, \ldots N\,. \tag{11.58}$$

- The contribution $q_{j,y}(x)$ with index y relates to proportion of servers used by stream j in global state x (carried traffic), and from this we find the average number of servers allocated to a type j customer. Given that we are in global state x, then the mean number of channels serving type j calls is:

  $$n_{j,y}(x) = \frac{p_{j,y}(x)}{p_j(x)} \cdot x\,,$$

  We have of course:

  $$\sum_{j=1}^{N} n_{j,y}(x) = n\,, \qquad x \geq n\,.$$

  For global state x we have the following total contributions to relative state probabilities:

  $$q_y(x) = \sum_{j=1}^{N} q_{j,y}(x)\,,$$

- The contribution $q_{j,l}(x)$ with index l relates to the proportion of queueing positions used by stream j. Given that we are in global state x, then the mean number of buffers occupied by type j calls is:

  $$n_{j,l}(x) = \frac{q_{j,l}(x)}{q_j(x)} \cdot x\,.$$

  For global state x we have the following total contributions to relative state probabilities:

  $$q_l(x) = \sum_{j=1}^{N} q_{j,l}(x)\,,$$

Of course we have:

$$\sum_{j=1}^{N} \{n_{j,y}(x) + n_{j,l}(x)\} = x\,.$$

For all traffic streams we have:

$$q(x) = q_y(x) + q_l(x)\,. \tag{11.59}$$

We find $q_{j,y}(x)$ in the following way. If we look at the local balance for type-j customers between state $[x - d_j]$ and state $[x]$, then under the assumption of statistical equilibrium we have:

$$\lambda_j \cdot q(x - d_j) = (x \cdot q_{j,y}(x)) \cdot \mu_j\,.$$

The requested service rate for type j calls is $d_j\,\mu_j$ per connection. The total bandwidth obtained by service j is $x \cdot q_{j,y}(x)$. Thus we find (cf. (7.41) for $Z = 1$):

$$q_{j,y}(x) \;=\; \frac{\lambda_j}{x\,\mu_j} \cdot q(x - d_j)\,, \tag{11.60}$$

$$q_{j,l}(x) \;=\; q_j(x) - q_{j,y}(x)\,. \tag{11.61}$$

Initialization values

The initial values of the relative state probabilities are

$$q_j(x) = q_{j,y}(x) = g_{j,l}(x) = 0\,, \quad x < d_j\,. \tag{11.62}$$

Thus (11.56) is not valid for $x = 0$, as it is undefined how the traffic streams contribute to state zero. We had the same initial conditions in the generalized algorithm for a multi-rate loss system in Sec. 7.6.2.

For $x \leq n$ the above algorithm is identical with the generalized algorithm for a multi-rate loss system with Poisson arrival processes (7.41). For states $x \leq n$ when all customers obtain the requested rate and there is no delay we have:

$$q_j(x) = q_{j,y}(x) \quad \text{and} \quad q_{j,l}(x) = 0\,, \qquad x \leq n\,. \tag{11.63}$$

Iteration and normalization

Knowing the state probability $q(0)$ for a system with $x = 0$ channels (11.56) we calculate the relative state $q_j(1)$ probabilities of the next state $x = 1$ using (11.57). From (11.56) we get the global state $q(1)$. We notice that state probability $q_j(x)$ depends only upon the global term $q(x - d_j)$, and the local term $q_j(x - d_j)$ and the traffic parameters of all traffic streams.

Then we increase the number of [BBU] by one to x and calculate the performance for this system. We first find the relative values $q(x)$ of state probabilities of state x. At the same

time we may calculate $q_{j,y}(x)$, $q_{j,l}(x)$, and the contributions to all performance measures given below. To obtain the new true state probabilities for x channels from the normalized state probabilities with $x-1$ channels, we have to divide all state probabilities from zero to x by the normalization constant $\{1+q(x)\}$. Thus all these states add to one. Also performance measures accumulated from individual state probabilities should be re-normalized during each step.

In this way we only need to store probabilities for states $(x-d_j, x-d_j+1, \ldots, x-1)$ to calculate the following normalized state probabilities. As special cases we have for example the accurate recursion formulæ for Erlang-B and Engset. The recursion is very stable and eliminates errors because we always divide with a factor $1+q(x)$ which is bigger than one. If we use the recursion in opposite direction, the numerical errors will accumulate as we then divide with a factor less than one $1-q(x)$.

To obtain accurate numerical values for any values of parameters we should normalize the state probabilities in each step as described in Sec. 4.4.1. Let us assume we have obtained the normalized state probabilities for all states from zero to $x-1$. From these we then calculate the relative values $q(x)$ of state probabilities of state x. To obtain the new true state probabilities we have to divide all state probabilities from zero to x by $\{1+q(x)\}$. Also performance measures accumulated from individual state probabilities should be re-normalized. In this way we only need to store probabilities for states $(x-d_j, x-d_j+1, \ldots, x-1)$ to calculate the following normalized state probabilities.

Wrapping up the algorithm in this way the computer memory requirement becomes of the order of size:

$$2\,(N+1)\cdot\max\{d_j\}\,.$$

The number of operations becomes of the order of size:

$$2\,(N+1)\cdot\max\{x\}\,.$$

If the number of states is unlimited, then we continue until, say, $p(x) < 10^{-c}$. Then the number of corrects digits with be at least $c-2$. Due to reversibility we may truncate the state space and limit the maximum state to $k \geq n$. Then we have a finite system with n channels and $k-n$ buffers operating in reversible processor sharing mode (mixed delay and loss system).

11.4.3 Derivation of recursion formula

The above recursion formula for $q_j(x)$ (11.57) is based on local balance for each service (reversibility). We know the total flow into state x due to arrival of calls of type j. This flow must be equal to the total service rate of calls of type j out from state x. We rewrite the

formula (11.57) to:

$$q_j(x) \cdot \min\{x, n\} = \left\{ \frac{d_j}{x} \cdot \lambda_j \cdot q(x - d_j) + \sum_{i=1}^{N} \left(\frac{x - d_i}{x} \cdot \lambda_i \cdot q_j(x - d_i) \right) \right\}$$

- **Left hand side:** Flow out of state x due to departure of any type of customer.
  This is the flow down from state x due to termination of any type of connection. We choose all service rates $\mu_i = 1$ and the service rate of a d_j-channel connection then becomes d_j. The total service rate in state x is $\min\{x, n\}$.
- **Right hand side:** Flow into state x due to arrival of customers.

  – Right hand side term one:
  This is new contribution to $q_j(x)$ because a new call type j arrives. Arrival rate of type j is λ_j. A new call type-j adds d_j channels to the new state x. Therefore the ratio of type-j channels in state x is increased by d_j/x. Channels occupied by type j which already exist when a new call arrive, is taken account of by the following term.

  – Right hand side term two:
  Already existing occupied channels of type-j in state $x - d_i$ is given by the proportion $q_j(x - d_i)$. If a call type i arrives in state $x - d_i$, then the contribution of these slots are shifted up to the new state x, so that the relative ratio of type j channels in state x is increased by $(x - d_i)/x$. The additional contribution when it happens to be a type j call is taken into account by term one.

11.4.4 Performance measures

For Poisson arrival processes we do not need to calculate the detailed state probabilities to obtain detailed performance measures for each service (11.57). We assume that the true state probabilities p have been obtained from the relative state probabilities q by normalization. In the following we specify the performance measures in [channels = BBU]. To get the measures in [connections] for service j we have to divide with d_j.

Contribution to the carried traffic of type j from state x is

$$y_j(x) = x \cdot p_{j,y}(x)\,.$$

Total carried traffic of type j becomes:

$$Y_j = \sum_{i=0}^{n+k} y_j(i) = \sum_{i=0}^{n+k} x \cdot p_{j,y}(x)\,. \tag{11.64}$$

If we have a finite buffer (k limited), then $Y_j < A_j$ and the traffic congestion C_j for type-j customers will be positive:

$$C_j = \frac{A_j - Y_j}{A_j} \,. \tag{11.65}$$

The time congestion can easily be obtained from the global state probabilities. Due to the *PASTA*-property, time, call, and traffic congestion are equal.

Contribution to the virtual queue length of traffic type j from state x is :

$$l_j(x) = x \cdot p_{j,l}(x) \,.$$

Total virtual queue length expressed in channels of traffic type j becomes:

$$L_j = \sum_{i=0}^{n+k} l_j(i) = \sum_{i=0}^{n+k} x \cdot p_{j,l}(x) \,. \tag{11.66}$$

The mean number of channels (servers and buffers in [BBU]) occupied by type j becomes (11.1):

$$\overline{L}_j = Y_j + L_j \tag{11.67}$$

From the mean virtual queue length, we obtain by using Little's theorem the virtual mean waiting time, expressed in mean service times. The average number of customers type j in the system is $\overline{L}_j/d_j$, and the average number of customers in the virtual queue is L_j/d_j. We get the following mean sojourn time, respectively mean virtual waiting time:

$$\overline{W}_j = \frac{\overline{L}_j}{d_j \, \lambda_j (1 - C_j)} \,, \tag{11.68}$$

$$W_j = \frac{L_j}{d_j \, \lambda_j (1 - C_j)} \,. \tag{11.69}$$

For the total system we get by proper weighting:

$$\overline{W} = \sum_{j=1}^{N} \frac{\lambda_j (1 - C_j)}{\lambda (1 - C)} \cdot \overline{W}_j = \frac{1}{\lambda (1 - C)} \cdot \sum_{j=1}^{N} \frac{\overline{L}_j}{d_j} \,, \tag{11.70}$$

$$W = \sum_{j=1}^{N} \frac{\lambda_j (1 - C_j)}{\lambda (1 - C)} \cdot W_j = \frac{1}{\lambda (1 - C)} \cdot \sum_{j=1}^{N} \frac{L_j}{d_j} \,. \tag{11.71}$$

11.5 Generalizations

From the state transition diagram it is obvious that the above results can be generalized to *BPP*-traffic and state dependent Poisson processes as the reductions factors are independent

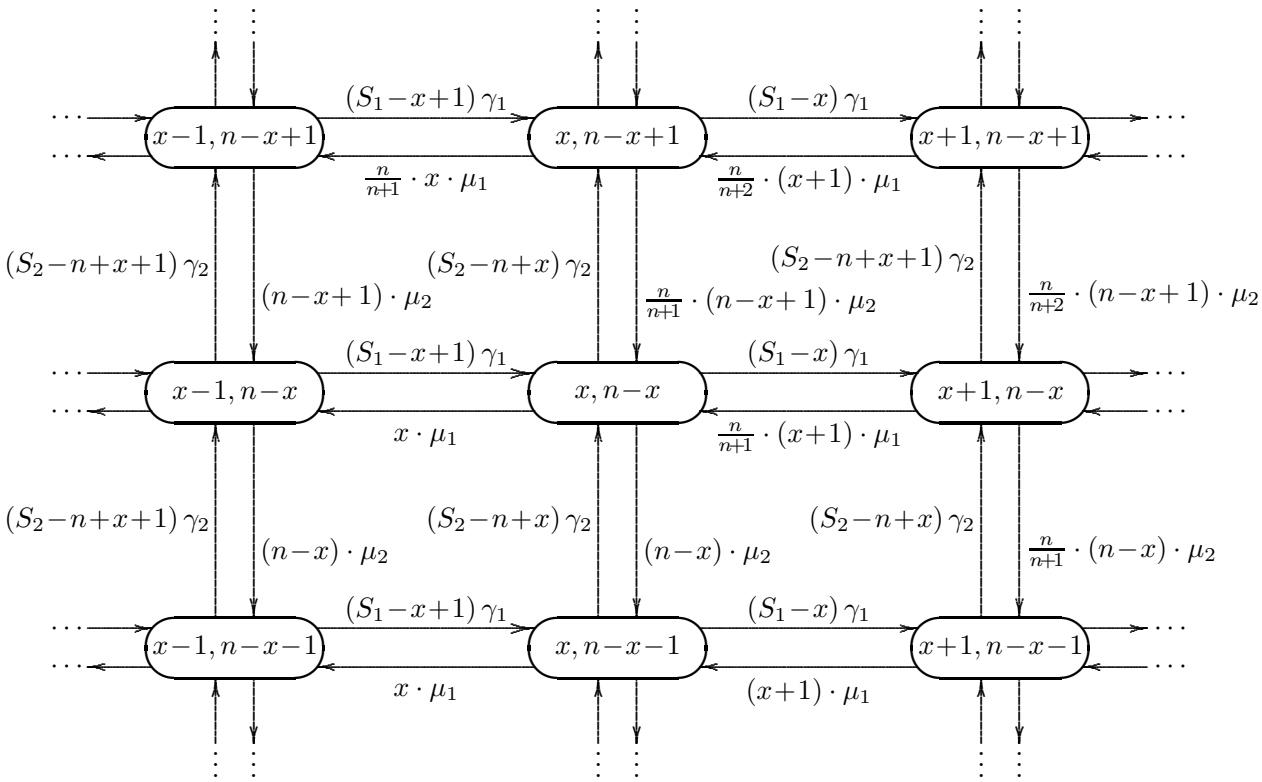


Figure 11.7: *State transition diagram for a reversible multi–dimensional system n servers, finite number of sources.*

of the traffic and only depend on the bandwidth demands. In Fig. 11.7 we show the state transition diagram for a system with two finite-source traffic streams and single-slot traffic. It is easy to see that it is reversible.

By calculating the reduction factors we can find the detailed state probabilities. However, we cannot set up generalized algorithms for global states, because it is not enough to know the average number of busy sources in a global state to find the actual arrival rate. For Poisson arrival processes we don't have this problem. We may easily truncate the system to a maximum of k channels. We still use the generalized algorithm, stopping at state k. We may put limits on the number of channels occupied by each service. We may also allocate each service a guaranteed minimum bandwidth. The state transition diagram is still reversible, but we have to calculate the detailed state probabilities.

In next chapter (Chap 12) we consider closed queueing networks, where the nodes are the queueing models described in this chapter. We include finite number of users in each chain by first assuming Poisson arrival processes, and then fix the number of customers in each chain during convolution and re-normalize the state probabilities. We do not directly use the finite source case shown in Fig. 11.7.

Chapter 12

Queueing networks

Many systems behave in such a way that a job achieves services from several successive nodes, i.e. once it has obtained service at one node, then it goes on to the next 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 *F*lexible *M*anufacturing *S*ystems (*FMS*). The terms job, customer, source, messages and others are used synonymously.

In queueing networks we define the queue-length in a node as the total number of jobs in the node, including delayed and served jobs. In the same way we define the waiting time as the total sojourn time, including both delay and service time. This is because the nodes in general operate as generalized processor sharing nodes, and not as classical non-sharing queueing systems (cf. Chap. 11).

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 large amount of parameters. In this chapter we shall give a simple introduction to general analytical queueing network models based on product forms. We also describe the convolution algorithm and the *MVA*–algorithm, illustrating the theory with examples. The theory of queueing networks is of same complexity as the theory of multi–dimensional loss networks(Chap. 7).

12.1 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. Erlang's classical waiting system, $M/M/n$, is an example of an open queueing network with one node, whereas Palm's machine/repair model with S terminals is a closed network with two nodes. If there are more than one type of customers, a network can be a mixed open and closed 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 be modeled as a Poisson process. This was analyzed in Chap. 11, and we will review the results in the section on symmetric queueing systems (Sec. 12.2).

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 $p(x_1, x_2, \ldots, x_K)$ where x_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 reversible (symmetric) queueing system, for example a Jackson network (Sec. 12.3), then we will have product form. The state probabilities of networks with product form can be aggregated and detailed performance measures obtained by using the convolution algorithm (Sec. 12.5.1) or the *MVA*–algorithm (Sec. 12.5.2).

Jackson networks can be generalized to *BCMP*–networks (Sec. 12.6), where there are N types of customers. Customers of one specific type all belongs to a so-called *chain*. Fig. 12.1 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 or jobs can be exact calculated. In case of a multi-chain network, the state of each node becomes multi-dimensional (Chap. 11). Within a node we do not have product form between the chains. But the product form between nodes is maintained, and the *convolution*–algorithm (Sec. 12.5.1) and the *MVA*–algorithm (Sec. 12.5.2) are applicable. A number of approximate algorithms for large networks are published in the literature.

12.2 Symmetric (reversible) queueing systems

In order to analyze queueing systems, it is important to know when the departure process of a queueing system is a Poisson process. The multi-service reversible queueing models dealt with in Chap. 11 all have this property, and the state probabilities are all given by the state probabilities of $M/M/n$ with special cases for $n = 1$ and $n = \infty$. We summarize the state probabilities for one service obtained in Chap. 9:

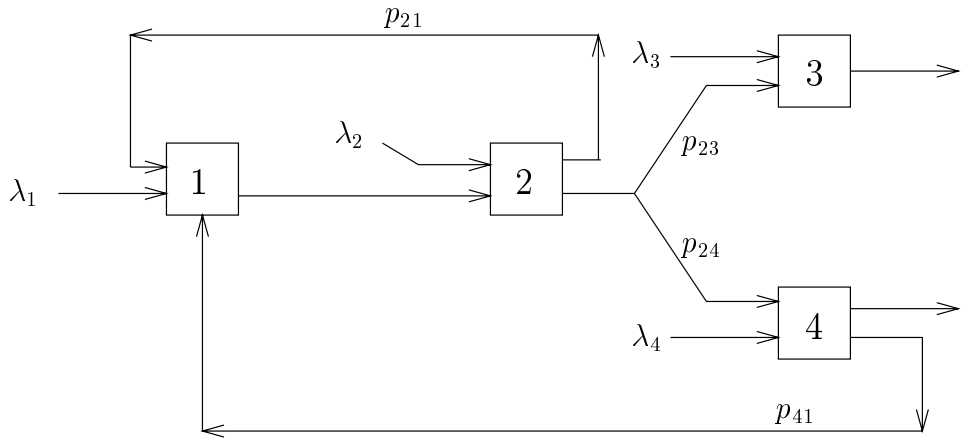


Figure 12.1: *An example of a queueing network with four open chains.*

1. *M/M/n*. This is *Burke's theorem* (Burke, 1956 [13]), which states, that the *departure process* of an *M/M/n*–system is a Poisson process. The state probabilities are given by (9.2) or (11.43):

$$p(x) = \begin{cases} p(0) \cdot \dfrac{A^x}{x!}\,, & 0 \le x \le n\,, \\ p(0) \cdot \dfrac{A^x}{n! \cdot n^{x-n}}\,, & x \ge n\,. \end{cases} \tag{12.1}$$

   where $A = \lambda/\mu$, and $p(0)$ is given by (9.4).

2. *IS* $= M/G/\infty$. *IS* is abbreviation for *I*nfinite *S*erver and this corresponds to the Poisson case (Sec. 4.2). From Sec. 3.6 we know that a random translation of the events of a Poisson process results in a new Poisson process. This model is denoted as a system with the queueing discipline *IS*, *I*nfinite number of *S*ervers. The state probabilities are given by the Poisson distribution (4.6):

$$p(x) = p(0) \cdot \frac{A^x}{x!}\,, \qquad i = 0, 1, 2, \ldots\,. \tag{12.2}$$

   where $p(0) = \mathrm{e}^{-A}$.

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 the same as for *M/M/1* (10.78)($n = 1$ in (12.1):

$$p(x) = p(0) \cdot A^x\,, \qquad x = 0, 1, 2, \ldots\,, \tag{12.3}$$

   where $p(0) = 1 - A$.

4. *M/G/n–GPS*
   This multi-server queueing system has the same state probabilities as *M/M/n* above (12.1).

5. *M/G/1–LCFS-PR* (PR = Preemptive Resume). This system also has the same state probabilities as *M/M/1* (12.3) with $p(0) = 1 - A$.

Above we have expressed all state probabilities by state zero as we later only need the relative state probabilities. Only these four queueing disciplines are easy to deal with in the theory of queueing networks. 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 reversible queueing systems are also 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 [70]). The process is called reversible because it looks the same way when we reverse the time (cf. when a movie 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.

Example 12.2.1: ***M/M/1* departure process**
At first it may seem illogical that the departure process of *M/M/1* with arrival rate λ and service rate μ is a Poisson process with rate λ. During busy periods (probability $A = \lambda/\mu$) the departure process is a Poisson process with rate μ. When the system becomes idle (probability $1 - A$) the inter-departure time become an inhomogeneous Erlang-2 distribution with rate λ in the first phase and rate μ in the second. In a phase diagram we may take the time intervals in reverse order so it looks like Fig. 2.11. From the decomposition principle of Cox-distributions it becomes obvious. A similar decomposition can be worked out for *M/M/n*. □

12.3 Open networks: single chain

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 [53]). He showed that a queueing network of *M/M/n* – nodes has product form. Knowing *Burke's theorem* (1956 [13]), Jackson's result is obvious. Historically, the first paper on queueing systems in series was by another Jackson, R.R.P. Jackson (1954 [52]).

Theorem 12.1 Jackson's theorem: *Consider an open queueing network with K nodes satisfying the following conditions:*

- *Structure: each node is an M/M/*n*–queueing system. Node k has n_k servers, and the average service time is $1/\mu_k$.*
- *Traffic: jobs arrive from outside the system to node k according to a Poisson process with intensity λ_k. Customers may also arrive to node k from other nodes.*

- *Strategy: a job which has just finished his service at node j, is immediately transferred 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 $p_{kk} > 0$.

Flow balance equations:
The total average arrival intensity Λ_k to node k is obtained by solving the flow balance equations:

$$\Lambda_k = \lambda_k + \sum_{j=1}^{K} \Lambda_j \cdot p_{jk} \, . \tag{12.4}$$

Let $p(x_1, x_2, \ldots, x_K)$ denote the state space probabilities under the assumption of statistical equilibrium, i.e. the probability that there is x_k customers at node k. Furthermore, we assume;

$$\frac{\Lambda_k}{\mu_k} = A_k < n_k \, . \tag{12.5}$$

Then the state space probabilities are given by the product form:

$$p\left(x_1, x_2, \ldots, x_K\right) = \prod_{k=1}^{K} p_k\left(x_k\right) \, . \tag{12.6}$$

where for node k, $p_k(x_k)$ is the state probabilities of Erlang's M/M/n queueing system with arrival rate Λ_k and service rate μ_k.

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 (12.5). 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 as for Erlang's delay system (Sec. 12.2). 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 node balance equations under the assumption of statistical equilibrium. Jackson's first model thus only deals with open queueing networks.

In Jackson's second model (Jackson, 1963 [54]) the arrival intensity from outside:

$$\lambda = \sum_{j=1}^{K} \lambda_j \tag{12.7}$$

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

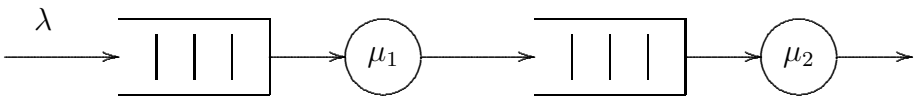


Figure 12.2: *State transition diagram of an open queueing network consisting of two M/M/1–systems in series.*

are either closed, open, or mixed. In all three cases, the state probabilities have product form. The model by Gordon & Newell (1967 [36]), which is often cited in the literature, can be treated as a special case of Jackson's second model.

Example 12.3.1: Two M/M/1 nodes in series
Fig. 12.2 shows an open queueing network of two *M/M/1* nodes in series. The corresponding state transition diagram is given in Fig. 12.3. Clearly, the state transition diagram is not reversible: (between two neighbour states there is only flow in one direction, (cf. Sec. 7.2). If we solve the balance equations to obtain the state probabilities we find that the solution can be written on a product form:

$$p(x_1, x_2) = p_1(x_1) \cdot p_2(x_2)\,,$$

$$p(x_1, x_2) = \left\{(1 - A_1) \cdot A_1^i\right\} \cdot \left\{(1 - A_2) \cdot A_2^j\right\}\,,$$

where $A_1 = \lambda/\mu_1$ and $A_2 = \lambda/\mu_2$. The state probabilities can be expressed in a product form $p(x_1, x_2) = p_1(x_1) \cdot p_2(x_2)$, where $p_1(x_1)$ is the state probabilities for a $M/M/1$ system with offered traffic A_i, and $p_2(x_2)$ is the state probabilities for a $M/M/1$ system with offered traffic A_2. The state probabilities of Fig. 12.3 are identical to those of Fig. 12.4, 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. 12.3. 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 12.3.2: Networks with feed back
Feedback is for example introduced in *Example 12.3.1* by letting a customer, which has just ended its service at node 2, return to node 1 with probability p_{21} (Fig. 12.2). With probability $1 - p_{21}$ the customer leaves the system. The flow balance equations (12.4) 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 $\lambda \to 0$ and $p_{21} \to 1$ we notice that the total arrival process to node 1 is not a Poisson processes:

only rarely a new job will arrive, but once it has entered the system it will circulate very fast many times. The number of times it loops back 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. 2.4.2). Instead of considering a single exponential inter-arrival distribution, we can decompose this into infinitely many phases (Fig. 2.12) and consider each phase as an arrival. Hence, the arrival process has been transformed from a Poisson process to a process with bursty arrivals. The total service time will be exponentially distributed with rate $\mu_1 \cdot (1 - p_{21})$, respectively $\mu_2 \cdot (1 - p_{21})$. But the total service time is split up into phases which are interleaved by waiting times and service time at the other node. □

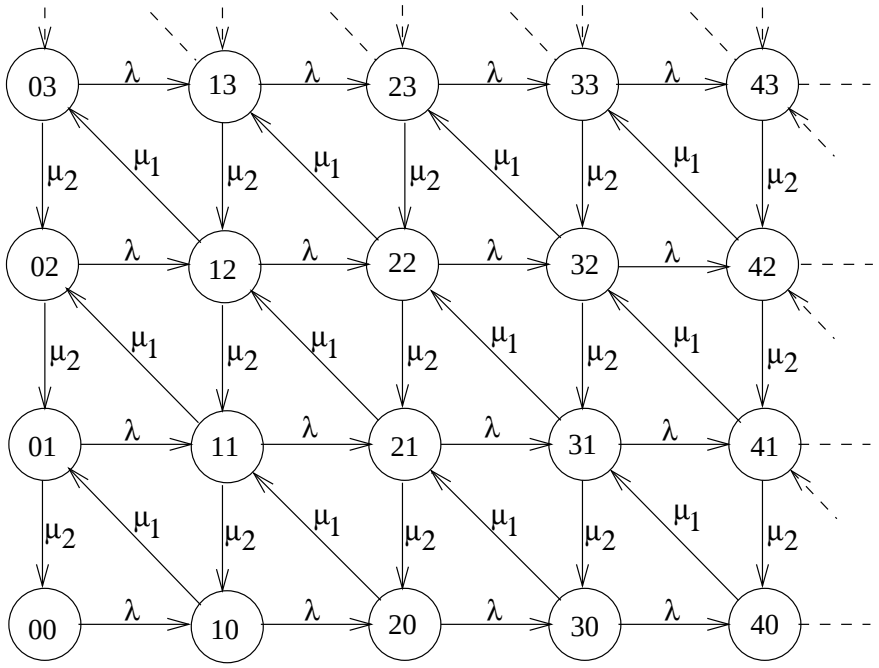


Figure 12.3: *State transition diagram for the open queueing network shown in Fig. 12.2. The diagram is non–reversible.*

12.3.1 Kleinrock's independence assumption

Above we assume a job sample a new service time with rate μ_i when the job arrives at node i, independent of the service time at other nodes. If we consider a real-life data network, then the packets will have the same length (for example in bytes), and therefore the same service time on all links and nodes of equal speed. The theory of queueing networks has to assume that a job 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 [75]), Many analysis show that turns out to be a good approximation to real systems.

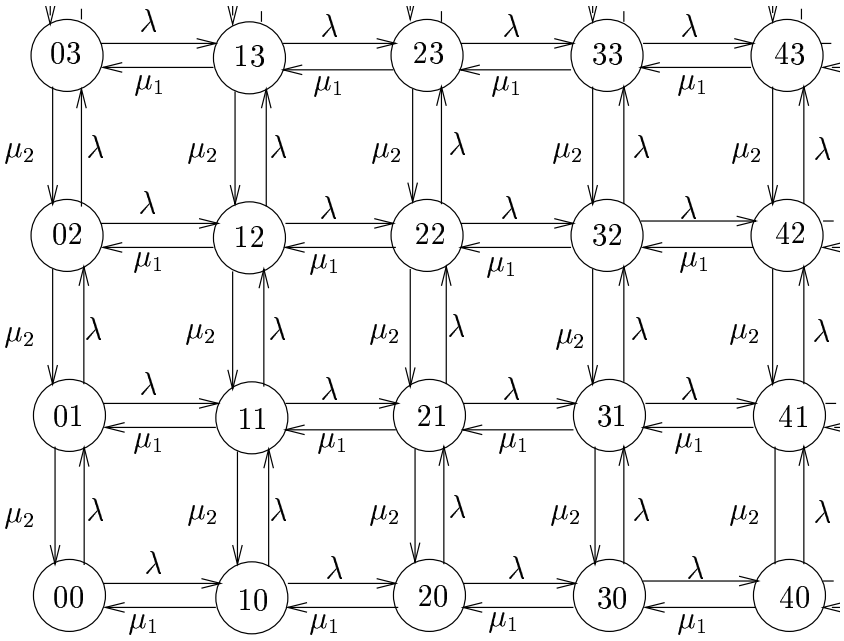


Figure 12.4: *State transition diagram for two independent $M/M/1$–queueing systems with identical arrival intensity, but individual mean service times. The diagram is reversible.*

12.4 Open networks: multiple chains

Dealing with *open systems* is easy. First we solve the flow balance equation (12.4) individually for each chain and obtain the arrival intensity for chain j to node k ($\Lambda_{j,k}$). The state probabilities for a node are then given by (11.42). We still have product form between the nodes, i.e. the nodes are independent, and we can easily calculate any state probability explicitly.

12.5 Closed networks: single chain

Dealing with *closed queueing networks* is much more complicated. We are interested in the state probabilities defined by $p(x_1, x_2, \ldots, x_k, \ldots, x_K)$, where x_k is the number of customers in node k ($1 \leq k \leq K$). With a fixed number of jobs we don't know the true arrival rate to the nodes. If we choose (or know) the arrival rate to a single node, then by solving the flow balance equations we find the relative arrival rate to all other nodes. Thus we can find the relative traffic to each node. To find the true normalized arrival rate and traffic, we have to find the normalization constant for the whole network, which means that we have to add all state probabilities.

12.5.1 Convolution algorithm

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 7).

We will now show how the convolution algorithm can be applied to closed queueing networks. The algorithm corresponds to the convolution algorithm for loss systems (Chapter 7). We consider a queueing network with K nodes and a single chain with S jobs. We assume that the queueing systems in each node are symmetric (Sec. 12.2). The algorithm has three steps:

- *Step 1: Flow balance equations*
  Let the arrival intensity to an arbitrary chosen reference node k be equal to some value Λ_k. By solving the flow balance equation (12.4) for the closed network we obtain the relative arrival rates Λ_k $(1 \leq k \leq K)$ to all nodes. We then obtain the relative offered traffic values $\alpha_k = \Lambda_k/\mu_k$. Often we choose the above arrival intensity of the reference node so that the offered traffic to this node becomes one.

- *Step 2: State probabilities*
  Consider each node as if it is isolated and has the offered random (*PCT-I*) traffic α_k $(1 \leq k \leq K)$. Depending on the actual symmetric queueing system at node k, we find the relative state probabilities $q_k(x_k)$ at node k. The state space will be limited by the total number of customers S: $0 \leq x_k \leq S$.

- *Step 3: convolution*
  Convolve the state probabilities of the nodes recursively. For example, for the first two nodes we have:

$$q_{12} = q_1 * q_2\,, \tag{12.8}$$

  where

$$q_{12}(x) = \sum_{i=0}^{x} q_1(i) \cdot q_2(x-i), \qquad x = 0, 1, \ldots, S\,.$$

  By convolution we reduce the number of nodes to two: The node we are interested in, and all other nodes aggregated into one node. When all nodes except node k have been convolved we have the final convolution:

$$q_{1,2,\ldots,k\ldots K} = q_{1,2,\ldots,k-1,k+1,\ldots K} * q_k\,, \tag{12.9}$$

  During the last convolution we convolve two nodes: the aggregated node consisting of all nodes except node k, and node k, and we obtain the detailed performance measures of node k. By changing the order of convolution of the nodes we can obtain the performance measures of all other nodes. Since the total number of customers is fixed (S) only state $q_{1,2,\ldots,K}(S)$ exists in the total aggregated system and therefore this macro-state must have the probability one. We can then normalize all micro-state probabilities.

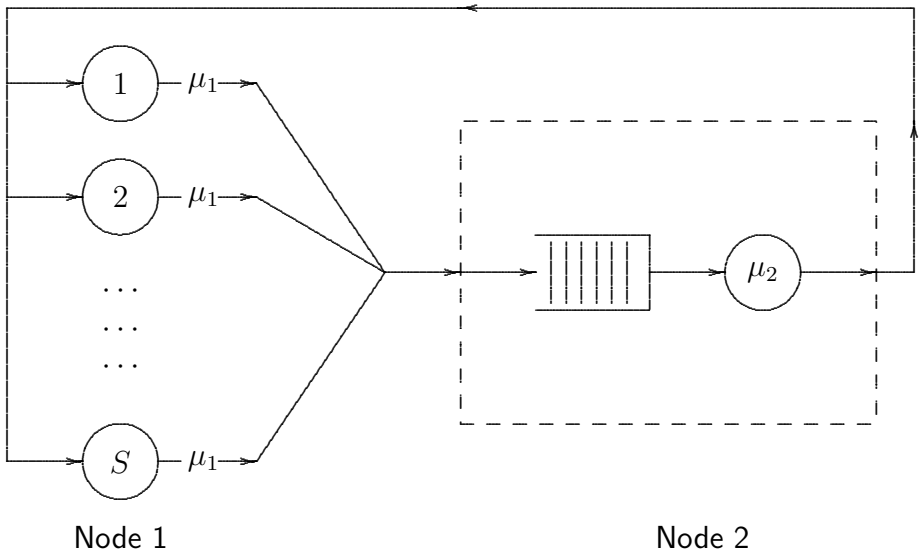


Figure 12.5: *The machine/repair model as a closed queueing networks with two nodes. The terminals correspond to one IS–node, because the tasks always find an idle terminal, whereas the CPU corresponds to an M/M/1–node.*

Example 12.5.1: Palm's machine/repair model
We consider the machine/repair model of Palm introduced in Sec. 9.6 as a closed queueing network (Fig. 12.5). There are S jobs and terminals one server (computer). 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. We choose the relative arrival rate to node one equal to Γ_1 and find $\Gamma_2 = \Gamma_1 = \Gamma$
The relative load at node 1 and node 2 are

$$\alpha_1 = \Lambda/\mu_1 \text{ and } \alpha_2 = \Lambda/\mu_2\,,$$

respectively. We consider each node in isolation and obtain the state probabilities of each node, $q_1(i)$ and $q_2(j)$, as if the arrival processes are Poisson processes. By convolving $q_1(x_1)$ and $q_2(x_2)$ we get $q_{12}(x)$, $(0 \le x \le S)$, as shown in Table 12.1.

The last term with S customers (an unnormalised probability) $q_{12}(S)$ is made up from the terms:

$$\begin{aligned} q_{12}(S) &= \sum_{i=0}^{S} q_1(i) \cdot q_2(S-i) \\ &= 1 \cdot \alpha_2^S + \alpha_1 \cdot \alpha_2^{S-1} + \frac{\alpha_1^2}{2!} \cdot \alpha_2^{S-2} + \ldots \cdot \frac{\alpha_1^i}{i!} \cdot \alpha_2^{x-i} + \ldots + \frac{\alpha_1^S}{S!} \cdot 1\,. \end{aligned}$$

State x	Node 1 $q_1(x_1)$	Node 2 $q_2(x_2)$	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	$\alpha_2^2 + \alpha_1 \cdot \alpha_2 + \frac{\alpha_1^2}{2!}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
x	$\frac{\alpha_1^x}{x!}$	α_2^x	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
S	$\frac{\alpha_1^S}{S!}$	α_2^S	$q_{12}(S)$

Table 12.1: *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 12.5.1).*

We know that this total has probability one, and from the individual contributions we identify the state probabilities of the two nodes. 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

$$\varrho = \frac{\alpha_1}{\alpha_2} = \frac{\mu_2}{\mu_1} .$$

The probability that all terminals are *thinking* is identified as the last term $q_1(S) \cdot q_2(0)$ (S terminals in node 1, zero terminals in node 2) normalized by the sum $q_{12}(S)$:

$$p\{x_1 = S, x_2 = 0\} = \frac{\frac{\varrho^S}{S!}}{1 + \varrho + \frac{\varrho^2}{2!} + \frac{\varrho^3}{3!} + \cdots + \frac{\varrho^S}{S!}} = E_{1,S}(\varrho) ,$$

which is Erlang's *B-formula*. Thus the result is in agreement with the result obtained in Sec. 9.6. 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 normalize. □

Example 12.5.2: Central server system

In 1971 J. P. Buzen introduced the *central server* model illustrated in Fig. 12.6 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. 12.6, 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 units. We assume that when a job is finished it is immediately replaced by a new job. Hence S is constant. The service times are all exponentially distributed with intensity μ_i $(i = 1, \ldots, K)$.

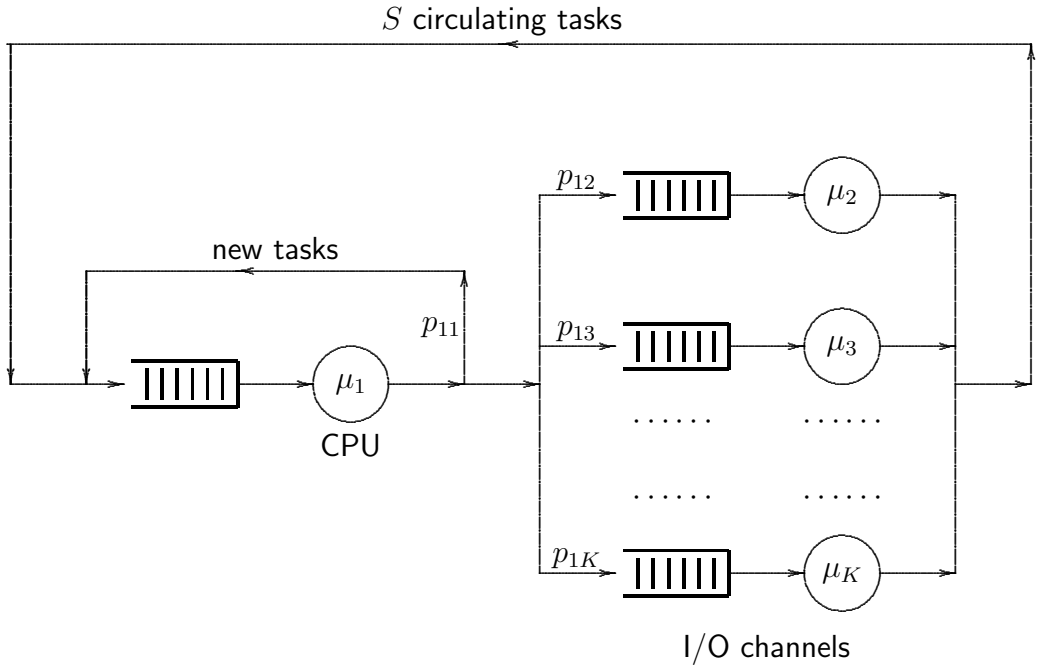


Figure 12.6: *Central server queueing system consisting of one central server (CPU) and* $(K-1)$ *I/O–channels. A fixed number of tasks* S *are circulating in the system.*

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 12.2. The term $q_{123}(4)$ is made up by:

$$q_{123}(4) = 1 \cdot 16 + 2 \cdot 8 + 3 \cdot 4 + 4 \cdot 2 + 5 \cdot 1 = 57.$$

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 12.2: *The convolution algorithm applied to the central server system.*

Node 3 serves customers in all states except for state $q_3(0) \cdot q_{12}(4) = 5$. The utilization of node 3 is therefore $a_3 = 52/57$. Based on the relative loads we now obtain the exact loads:

$$a_1 = \frac{26}{57}\,, \qquad a_2 = \frac{26}{57}\,, \qquad a_3 = \frac{52}{57}\,.$$

The average number of customers at node 3 is:

$$\begin{aligned} L_3 &= \{1 \cdot (4 \cdot 2) + 2 \cdot (3 \cdot 4) + 3 \cdot (2 \cdot 8) + 4 \cdot (1 \cdot 16)\} \,/\, 57\,, \\ L_3 &= \frac{144}{57}\,. \end{aligned}$$

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 utilization 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 $W_k = L_k/\lambda_k$:

$$W_1 = 45.23\,, \quad W_2 = 64.62\,, \quad W_3 = 775.38\,.$$

□

12.5.2 MVA–algorithm

The Mean Value Algorithm (*MVA*) is an algorithm for calculating performance measures of queueing networks where all nodes are single-server systems. It combines in an elegant way two main results in queueing theory: the arrival theorem (5.29) and Little's law (3.20). The algorithm was first published by Lavenberg & Reiser (1980 [82]).

We consider a queueing network with K nodes and S customers (all belonging to a single chain). We choose some value of the arrival rate to some node, for example $\lambda_1 = 1$ to node one. From the flow balance equations we find the relative arrival rates to all other nodes. The relative load of node k is $\alpha_k = \lambda_k \cdot s_k$. $(k = 1, 2, \ldots, K)$. The algorithm is recursive in number of customers as a network with $S + 1$ customers is evaluated from a network with S customers.

Assume that the average number of customers at node k is $L_k(S)$ where S is the total number of customers in the network. Obviously

$$\sum_{k=1}^{K} L_k(S) = S\,. \tag{12.10}$$

The algorithm is recursive in two steps:

Step 1: Arrival theorem
Increase the number of customers from S to $(S + 1)$. According to the arrival theorem, the $(S + 1)th$ customer will see the system as a system with S 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:*
$$W_k(S + 1) = \{L_k(S) + 1\} \cdot s_k\,.$$
- For *M/G/∞:*
$$W_k(S + 1) = s_k\,.$$

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: Little's theorem
We apply Little's law $(L = \lambda \cdot W)$, which is valid for all systems in statistical equilibrium. For node k we have:

$$L_k(S + 1) = c \cdot \lambda_k \cdot W_k(S + 1)\,,$$

where λ_k is the relative arrival rate to node k. The normalizing constant c is obtained from the total number of customers::

$$\sum_{k=1}^{K} L_k(S + 1) = S + 1\,. \tag{12.11}$$

By these two steps we have performed the recursion from S to $(S+1)$ customers. For $S=1$ there will be no waiting time in the system and $W_k(1)$ equals the average service time s_k.

Nodes with a limited number of servers ($n>1$) can only be dealt with approximately by the *MVA*–algorithm, but are easy to deal with by the convolution algorithm.

Example 12.5.3: Central server model
We apply the *MVA*–algorithm to the central server model (Example 12.5.2). The relative arrival rates are:

$$\lambda_1 = 1\,, \qquad \lambda_2 = 0.7\,, \qquad \lambda_3 = 0.2\,.$$

	Node 1		Node 2		Node 3	
$S=1$	$W_1(1)$	$= 28$	$W_2(1)$	$= 40$	$W_3(1)$	$= 280$
	$L_1(1)$	$= c\cdot 1\cdot 28$	$L_2(1)$	$= c\cdot 0.7\cdot 40$	$L_3(1)$	$= c\cdot 0.2\cdot 280$
	$L_1(1)$	$= 0.25$	$L_2(1)$	$= 0.25$	$L_3(1)$	$= 0.50$
$S=2$	$W_1(2)$	$= 1.25\cdot 28$	$W_2(2)$	$= 1.25\cdot 40$	$W_3(2)$	$= 1.50\cdot 280$
	$L_1(2)$	$= c\cdot 1\cdot 1.25\cdot 28$	$L_2(2)$	$= c\cdot 0.7\cdot 1.25\cdot 40$	$L_3(2)$	$= c\cdot 0.2\cdot 1.50\cdot 280$
	$L_1(2)$	$= 0.4545$	$L_2(2)$	$= 0.4545$	$L_3(2)$	$= 1.0909$
$S=3$	$W_1(3)$	$= 1.4545\cdot 28$	$W_2(3)$	$= 1.4545\cdot 40$	$W_3(3)$	$= 2.0909\cdot 280$
	$L_1(3)$	$= c\cdot 1\cdot 1.4545\cdot 28$	$L_2(3)$	$= c\cdot 0.7\cdot 1.4545\cdot 40$	$L_3(3)$	$= c\cdot 0.2\cdot 2.0909\cdot 280$
	$L_1(3)$	$= 0.6154$	$L_2(3)$	$= 0.6154$	$L_3(3)$	$= 1.7692$
$S=4$	$W_1(4)$	$= 1.6154\cdot 28$	$W_2(4)$	$= 1.6154\cdot 40$	$W_3(4)$	$= 2.7692\cdot 280$
	$L_1(4)$	$= c\cdot 1\cdot 1.6154\cdot 28$	$L_2(4)$	$= c\cdot 0.7\cdot 1.6154\cdot 40$	$L_3(4)$	$= c\cdot 0.2\cdot 2.7692\cdot 280$
	$L_1(4)$	$= 0.7368$	$L_2(4)$	$= 0.7368$	$L_3(4)$	$= 2.5263$

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 12.5.4: 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. 9.6.2 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 (4.29). The relative arrival 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	$W_2(1)$	$=$	1
	$L_1(1)$	$=$	$c \cdot 1 \cdot A$	$L_2(1)$	$=$	$c \cdot 1 \cdot 1$
	$L_1(1)$	$=$	$\frac{A}{1+A}$	$L_2(1)$	$=$	$\frac{1}{1+A}$
$S=2$	$W_1(2)$	$=$	A	$W_2(2)$	$=$	$1+\frac{1}{1+A}$
	$L_1(2)$	$=$	$c \cdot 1 \cdot A$	$L_2(2)$	$=$	$c \cdot 1 \cdot (1+\frac{1}{1+A})$
	$L_1(2)$	$=$	$A \cdot \frac{1+A}{1+A+\frac{A^2}{2!}}$	$L_2(2)$	$=$	$2 - A \cdot \frac{1+A}{1+A+\frac{A^2}{2!}}$
$\ldots$			$\ldots$			$\ldots$
$S=x$	$W_1(x)$	$=$	A	$W_2(x)$	$=$	$1 + L_2(x-1)$
	$L_1(x)$	$=$	$c \cdot A$	$L_2(x)$	$=$	$c \cdot \{1 + L_2(x-1)\}$
	$L_1(x)$	$=$	$A \cdot \{1 - E_x(A)\}$	$L_2(x)$	$=$	$x - A \cdot \{1 - E_x(A)\}$

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: From this we have the normalization constant $c = 1 - E_x(A)$ and we get for the $(x+1)$'th customer:

$$\begin{aligned} L_1(x+1) + L_2(x+1) &= c \cdot A + c \cdot \{1 + L_2(x)\} \\ x+1 &= c \cdot A + c \cdot \{1 + x - A \cdot (1 - E_x)\} \\ E_{x+1} &= \frac{A \cdot E_x}{x+1+A \cdot E_x}\,, \end{aligned}$$

because we know $c = 1 - E_{x+1}$. This is just the recursion formula for the Erlang–B formula. □

12.6 BCMP multi-chain 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 (reversible) queueing system (cf. Sec. 12.2: Poisson arrival process ⇒ Poisson departure process).

b) The customers are classified into N *chains*. Each chain is characterized by its own mean service time s_j and transition probabilities p_{ik}^j. A restriction applies if the queueing discipline at a node is a *non-sharing* $M/M/n$ queueing system (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.

12.6.1 Convolution algorithm

The algorithm is essentially the same as in the single chain case:

- Step 1: Flow balance equations
  Consider each chain as if it is alone in the network. Find the relative load at each node by solving the flow balance equation (12.4). At an arbitrary *reference node* we assume the arrival rate 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):

$$\lambda_k^j = \sum_{i=1}^{K} p_{ik}^j \cdot \lambda_i^j, \qquad j = 1, \ldots, N\,, \tag{12.12}$$

  where:

  K = number of nodes,

  N = number of chains,

  p_{ik}^j = the probability that a customer of chain j moves 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. Notice j is an index, *not* a power.

- *Step 2: State probabilities*
  Based on the relative loads found in step 1, we obtain the multi-dimensional state probabilities for each node (Sec. 11.3.5). 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 ($1 \leq k \leq K$):

$$\underline{p}_k = p_k(x_1, x_2, \ldots, x_N)\,, \qquad 0 \leq x_j \leq S_j\,, \quad j = 1, 2, \ldots N\,,$$

  where S_j is the number of customers in chain j.

- *Step 3: Convolution* ¡¡ 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:

$$\prod_{j=1}^{N}(S_j+1)\,. \tag{12.13}$$

The number of ways the customers can be distributed in a queueing network with K nodes and N chains with S_j customers in chain j is:

$$\mathcal{C} = \prod_{j=1}^{N} C(S_j, k_j) \tag{12.14}$$

where k_j ($1 \leq k_j < k$) is the number of nodes visited by chain j and:

$$C(S_j, k_j) = \begin{pmatrix} S_j + k_j - 1 \\ k_j - 1 \end{pmatrix} = \begin{pmatrix} S_j + k_j - 1 \\ S_j \end{pmatrix}. \tag{12.15}$$

The algorithm is best illustrated with an example.

Example 12.6.1: Palm's machine-repair model with two types of customers
As seen in Example 12.5.1, 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 modeled as an Infinite Server *IS*-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 of 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. 7.10):

$$
\begin{aligned}
q_1(0,0) &= 1 & q_1(0,2) &= \frac{\beta_1^2}{2} \\
q_1(1,0) &= \alpha_1 & q_1(1,2) &= \frac{\alpha_1 \cdot \beta_1^2}{2} \\
q_1(2,0) &= \frac{\alpha_1^2}{2} & q_1(2,2) &= \frac{\alpha_1^2 \cdot \beta_1^2}{4} \\
q_1(0,1) &= \beta_1 & q_1(0,3) &= \frac{\beta_1^3}{6} \\
q_1(1,1) &= \alpha_1 \cdot \beta_1 & q_1(1,3) &= \frac{\alpha_1 \cdot \beta_1^3}{6} \\
q_1(2,1) &= \frac{\alpha_1^2 \cdot \beta_1}{2} & q_1(2,3) &= \frac{\alpha_1^2 \cdot \beta_1^3}{12}
\end{aligned}
$$

  For node 2 (single server) (cf. 11.23) 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 $(2,3)$:

$$\begin{aligned} q_{12}(2,3) &= q_1(0,0)\cdot q_2(2,3) + q_1(1,0)\cdot q_2(1,3) \\ &\quad + q_1(2,0)\cdot q_2(0,3) + q_1(0,1)\cdot q_2(2,2) \\ &\quad + q_1(1,1)\cdot q_2(1,2) + q_1(2,1)\cdot q_2(0,2) \\ &\quad + q_1(0,2)\cdot q_2(2,1) + q_1(1,2)\cdot q_2(1,1) \\ &\quad + q_1(2,2)\cdot q_2(0,1) + q_1(0,3)\cdot q_2(2,0) \\ &\quad + q_1(1,3)\cdot q_2(1,0) + q_1(2,3)\cdot q_2(0,0) \end{aligned}$$

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:

$$\begin{aligned} \frac{\frac{1}{12}\cdot\alpha_1^2\cdot\beta_1^3}{q_{12}(2,3)} &= \frac{\alpha^5/12}{10+4\alpha+\frac{1}{2}\alpha^2+6\alpha+3\alpha^2+\frac{1}{2}\alpha^3+\frac{3}{2}\alpha^2+\alpha^3+\frac{1}{4}\alpha^4+\frac{1}{6}\alpha^3+\frac{1}{6}\alpha^4+\frac{1}{12}\alpha^5} \\ &= \frac{\frac{\alpha^5}{5!}}{1+\alpha+\frac{\alpha^2}{2}+\frac{\alpha^3}{3!}+\frac{\alpha^4}{4!}+\frac{\alpha^5}{5!}}\ , \end{aligned}$$

i.e. the Erlang–B formula as expected. □

12.7 Other algorithms for queueing networks

The *MVA*–algorithm is also applicable to queueing networks with more chains, when the nodes are single-server systems. During the last decade several algorithms have been published. An overview can be found in (Conway & Georganas, 1989 [16]). 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.

12.8 Complexity

Queueing networks has the same complexity as circuit switched networks with direct routing (Sec. 8.5 and Tab. 8.2). The state space of the network shown in Tab. 12.3 has the following number of states for every node (12.13):

$$\prod_{i=0}^{N}(S_i + 1)\,. \tag{12.16}$$

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	···	K	Size
1	α_{11}	α_{21}	···	α_{K1}	S_1
2	α_{12}	α_{22}	···	α_{K2}	S_2
···	···	···	···	···	···
N	α_{1N}	α_{2N}	···	α_{KN}	S_N

Table 12.3: *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. 8.2).*

12.9 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]. The mean service time becomes:

$$s = \frac{1/\mu_k}{\varphi_k} = \frac{1}{\mu_k \, \varphi_k} \, .$$

So the mean service rate is $\mu_k \, \varphi_k$ and the mean sojourn time is given by (9.34):

$$m_{1,k} = \frac{1}{\mu_k \, \varphi_k - \lambda_k} \, .$$

We introduce the following linear restriction on the total capacity:

$$F = \sum_{k=1}^{K} \varphi_k \, . \tag{12.17}$$

For every allocation of capacity which satisfies (12.17), we have the following mean sojourn time for all messages (call average):

$$m_1 = \sum_{k=1}^{K} \frac{\lambda_k}{\lambda} \cdot \frac{1}{\mu_k \cdot \varphi_k - \lambda_k} \, , \tag{12.18}$$

where:

$$\lambda = \sum_{k=1}^{K} \lambda_k \, . \tag{12.19}$$

By applying (10.54) we get the total mean service time:

$$\frac{1}{\mu} = \sum_{k=1}^{K} \frac{\lambda_k}{\lambda} \cdot \frac{1}{\mu_k} \, . \tag{12.20}$$

The total offered traffic is then:

$$A = \frac{\lambda}{\mu \cdot F} \, . \tag{12.21}$$

Kleinrock's law for optimal capacity allocation (Kleinrock, 1964 [75]) reads:

Theorem 12.2 Kleinrock's square root law: *The optimal allocation of capacity which minimises m_1 (and thus the total number of messages in all nodes) is:*

$$\varphi_k = \frac{\lambda_k}{\mu_k} + F \cdot (1 - A) \frac{\sqrt{\lambda_k / \mu_k}}{\sum_{i=1}^{K} \sqrt{\lambda_i / \mu_i}} \, , \tag{12.22}$$

under the condition that:

$$F > \sum_{k=1}^{K} \frac{\lambda_k}{\mu_k} \, . \tag{12.23}$$

Proof: This can be shown by introducing *Lagrange multiplier* ϑ and consider:

$$G = m_1 - \vartheta \left\{ \sum_{k=1}^{K} \varphi_k - F \right\} . \tag{12.24}$$

Minimum of G is obtained by choosing φ_k as given in (12.22).

With this optimal allocation we find the mean sojourn time:

$$m_1 = \frac{\left\{ \sum_{k=1}^{K} \sqrt{\lambda_k/\mu_k} \right\}^2}{\lambda \cdot F \cdot (1 - A)} . \tag{12.25}$$

This optimal allocation corresponds to that all nodes first are allocated the necessary minimum capacity λ_i/μ_i. The remaining capacity (12.20):

$$F - \sum_{k=1}^{K} \frac{\lambda_i}{\mu_i} = F \cdot (1 - A) \tag{12.26}$$

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 [75]).

Chapter 13

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 realization 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.

13.1 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 center, etc. (cf. number representation, Sec. 3.1.1).

2. *Time intervals:* examples are conversation times, execution times of jobs in a computer, waiting times, etc. (cf. interval representation, Sec. 3.1.2).

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.

13.1.1 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 13.1.1: 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.

□

13.1.2 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 13.1.2: 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 [66]). 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 analyzing the statistical problems. The classical equipment operating in state space has implied that the statistical analyzes 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 [42]) *(horizontal measurements).*

13.2 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:

$$\bar{X} = \frac{1}{n} \cdot \sum_{i=1}^{n} X_i \tag{13.1}$$

$$s^2 = \frac{1}{n-1} \left\{ \sum_{i=1}^{n} X_i^2 - n \cdot \bar{X}^2 \right\} \tag{13.2}$$

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.:

$$E\{\bar{X}\} = m_1 \tag{13.3}$$

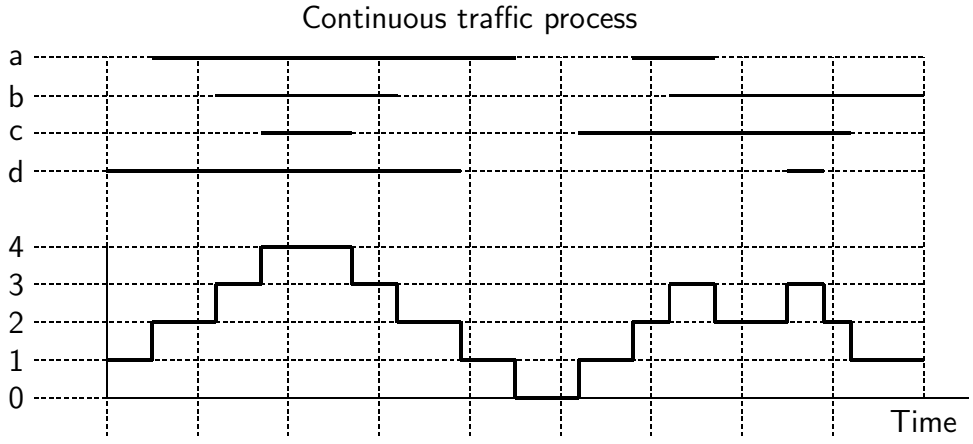
Furthermore, s^2/n is a central estimator of the unknown variance of the sample mean $\bar{X}$, i.e.:

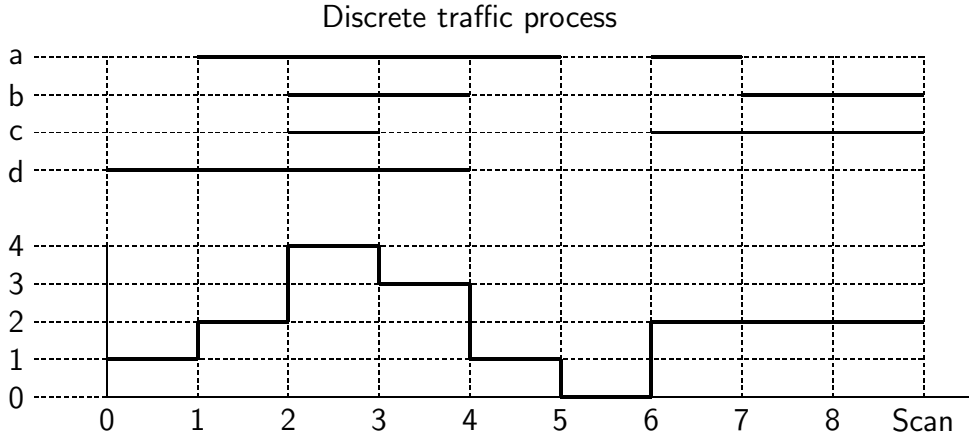
$$\bar{\sigma^2}\{\bar{X}\} = s^2/n. \tag{13.4}$$

We describe the accuracy of an estimate of a sample parameter by means of a confidence interval, which with a given probability specifies how this estimate is placed relatively to the unknown population value. In our case the confidence interval of the mean value becomes:

$$\bar{X} \pm t_{n-1,1-\alpha/2} \cdot \sqrt{\frac{s^2}{n}} \tag{13.5}$$

where $t_{n-1,1-\alpha/2}$ is the upper $(1-\alpha/2)$ percentile of the Student's t-distribution with $n-1$ degrees of freedom. The probability that the confidence interval includes the unknown population mean value is equal to $(1-\alpha)$ and is called the level of confidence. Some values of the Student's t-distribution are given in Table 13.1. When n becomes large, then the Student's t-distribution converges to the Normal distribution, and we may use the percentiles 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





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 13.1: *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

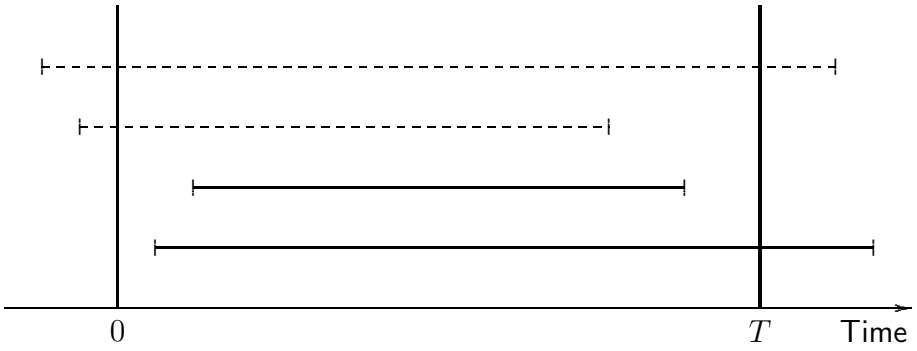
Table 13.1: *Percentiles of the Student's t-distribution with n degrees of freedom. A specific value of α corresponds to a probability mass $\alpha/2$ in both tails of the Student's t-distribution. When n is large, then we may use the percentiles of the Normal distribution.*

the following sections we calculate the mean value and the variance of traffic measurements during for example one hour. This aggregated value 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.

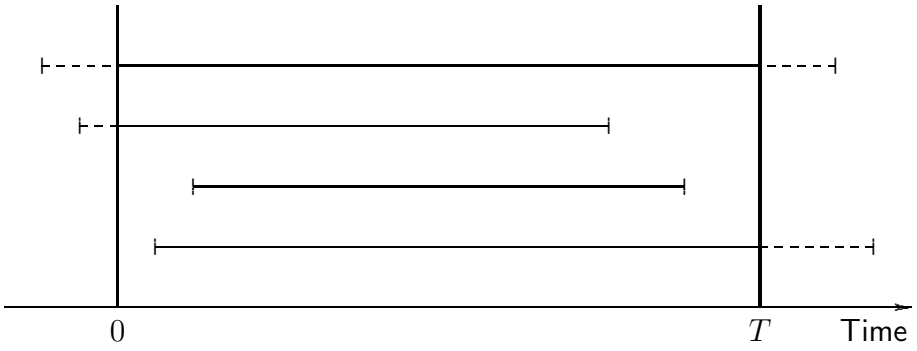
Example 13.2.1: Confidence interval for call congestion
On a trunk group of 30 trunks (channels) we observe the outcome of 500 call attempts. This measurement is repeated 11 times, and we find the following call congestion values (in percentage):

$$\{9.2,\ 3.6,\ 3.6,\ 2.0,\ 7.4,\ 2.2,\ 5.2,\ 5.4,\ 3.4,\ 2.0,\ 1.4\}$$

The total sum of the observations is 45.4 and the total of the squares of the observations is 247.88 . We find (13.1) $\bar{X} = 4.1273$ % and (13.2) $s^2 = 6.0502$ $(\%)^2$. At 95%–level the confidence interval becomes by using the t-values in Table 13.1: (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 13.5), 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). □



a: Unlimited measuring period



b: Limited measuring period

Figure 13.2: *When analyzing 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.*

13.3 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. 13.2 above.

For a *traffic volume* or a *traffic intensity* we can apply the formulæ (2.85) and (2.87) for a stochastic sum. They are quite general, the only restriction being stochastic independence 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. 2.4). 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:

$$\begin{aligned} m_{1,s} &= \lambda T \cdot m_{1,t} \\ \sigma_s^2 &= \lambda T \left\{ m_{1,t}^2 + \sigma_t^2 \right\} \\ &= \lambda T \cdot m_{2,t} = \lambda T \cdot m_{1,t}^2 \cdot \varepsilon_t \,, \end{aligned} \tag{13.6}$$

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:

$$\varepsilon = \frac{m_{2,t}}{m_{1,t}^2} = 1 + \frac{\sigma_t^2}{m_{1,t}^2} \tag{13.7}$$

The distribution of S_T will in this case be a *compound Poisson distribution* (Feller, 1950 [32]).

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:

$$m_{1,i} = A \tag{13.8}$$

$$\sigma_i^2 = \frac{A}{T} \cdot \varepsilon_t \tag{13.9}$$

These formulæ are thus valid for arbitrary holding time distributions. The formulæ (13.8) and (13.9) are originally derived by C. Palm (1941 [93]). In (Rabe, 1949 [101]) 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. 13.2 a).

Example 13.3.1: Accuracy of a measurement
We notice that we always obtain the correct mean value of the traffic intensity (13.8). 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. 2.5):	$\sigma_i^2 = \frac{A}{T} \cdot 3.83\,.$

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. 2.5). 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 13.3.2: 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. 4.8.1). Therefore we measure the same time period on all trunk groups. In Fig. 13.5 the relative accuracy for a continuous measurement is given by the straight line $h = 0$.

13.4 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. 13.3):

Observed time	Real time
$0\ h$	$0\ h - 1\ h$
$1\ h$	$0\ h - 2\ h$
$2\ h$	$1\ h - 3\ h$
$3\ h$	$2\ h - 4\ h$
...	...

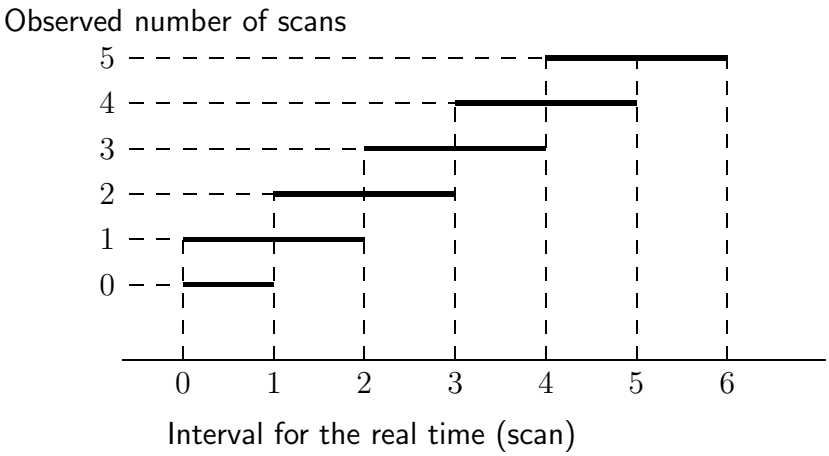


Figure 13.3: *By the scanning method a continuous time interval is transformed into a discrete time interval. The transformation is not unique (cf. Sec. 13.4).*

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 it can be shown that we will observe the following discrete distribution (Iversen, 1976 [42]):

$$p(0) = \frac{1}{h}\int_0^h F(t)\,dt \tag{13.10}$$

$$p(k) = \frac{1}{h}\int_0^h \{F(t+kh) - F(t+(k-1)h)\}\,dt\,, \quad k = 1, 2, \ldots\,. \tag{13.11}$$

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. 3.6.3). 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 $(0 \le t < h)$ and get (13.10). In a similar way we derive $p(k)$ (13.11).

By partial integration it can be shown that for any distribution function $F(t)$ we will *always observe the correct mean value:*

$$h \cdot \sum_{k=0}^{\infty} k \cdot p(k) = \int_0^{\infty} t \cdot dF(t)\,. \tag{13.12}$$

When using Karlsson charging we will therefore always in the long run charge the *correct amount*.

For exponential distributed holding time intervals, $F(t) = 1 - \mathrm{e}^{-\mu t}$, we will observe a discrete distribution, *Westerberg's distribution* (Iversen, 1976 [42]):

$$p(0) = 1 - \frac{1}{\mu h}\left(1 - \mathrm{e}^{-\mu h}\right), \tag{13.13}$$

$$p(k) = \frac{1}{\mu h}\left(1 - \mathrm{e}^{-\mu h}\right)^2 \cdot \mathrm{e}^{-(k-1)\mu h}, \quad k = 1, 2, \ldots \tag{13.14}$$

This distribution can be shown to have the following mean value and form factor:

$$m_1 = \frac{1}{\mu h}, \tag{13.15}$$

$$\varepsilon = \mu h \cdot \frac{\mathrm{e}^{\mu h} + 1}{\mathrm{e}^{\mu h} - 1} \geq 2. \tag{13.16}$$

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. 13.4 shows how the form factor of the observed holding time for exponentially distributed holding times depends on the length of the scanning interval (13.16). 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. 3.2 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 (13.9), 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:

$$\begin{aligned} m_{1,i} &= A, \\ \sigma_i^2 &= \frac{A}{T}\left\{h \cdot \frac{\mathrm{e}^h + 1}{\mathrm{e}^h - 1}\right\}. \end{aligned} \tag{13.17}$$

By the continuous measuring method the variance is $2A/T$. This we also get now by letting $h \to 0$.

Fig. 13.5 shows the relative accuracy of the measured traffic volume, both for a continuous measurement (13.8) & (13.9) and for the scanning method (13.17). Formula (13.17) was derived by (Palm, 1941 [93]), but became only known when it was "re-discovered" by W.S. Hayward Jr. (1952 [38]).

Example 13.4.1: 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 synchronized with the scanning process. This is for example applied for operator handled calls and in coin box telephones.

□

13.5 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. 13.2):

$$m_{1,i} \pm 1,96 \cdot \sigma_i \tag{13.18}$$

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 13.5 we can draw some general conclusions:

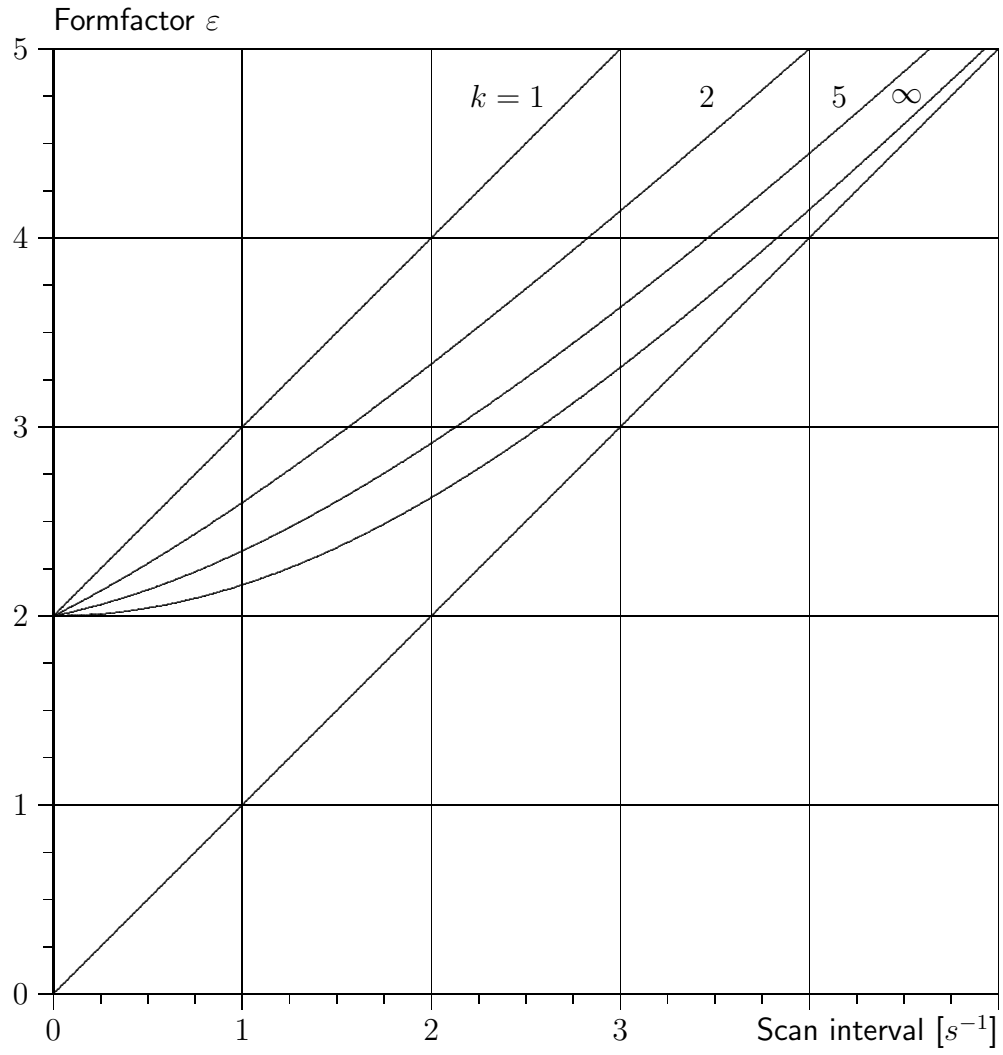


Figure 13.4: *Form factor for exponentially distributed holding times which are observed by Erlang-k distributed scanning intervals in an unlimited measuring period. The case $k = \infty$ corresponds to regular (constant) scan intervals which transform the exponential distribution into Westerberg's distribution. 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).*

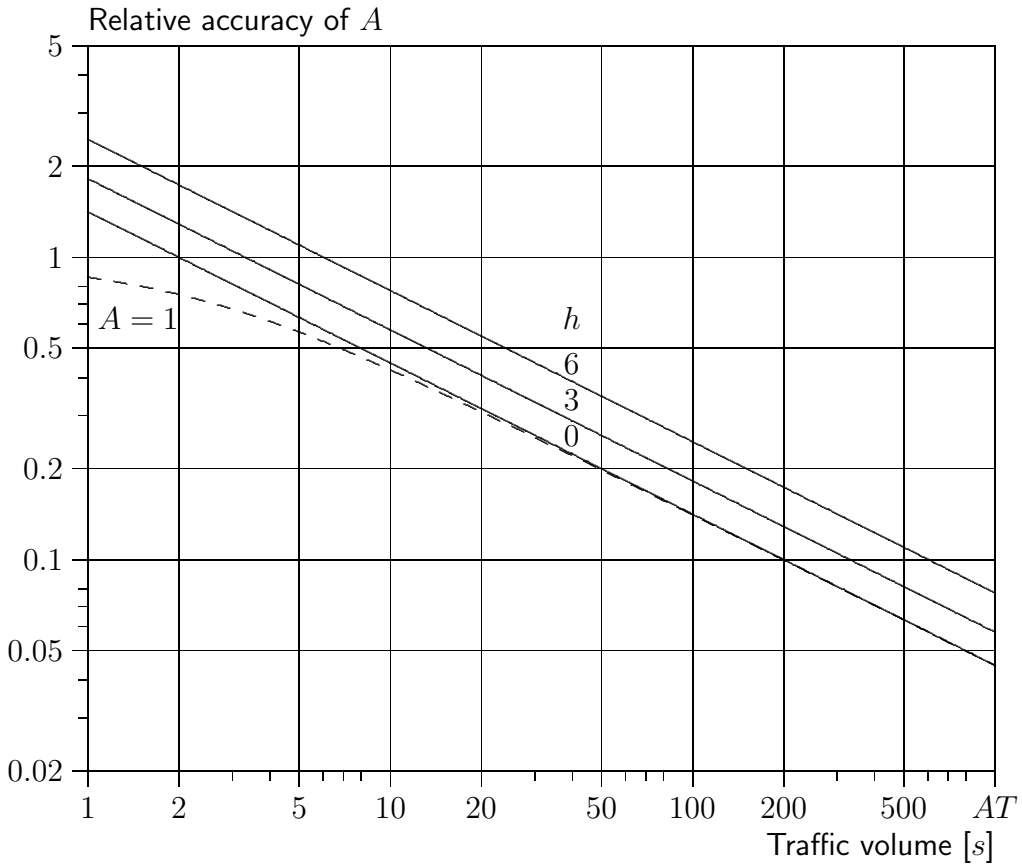


Figure 13.5: *Using double-logarithmic scale we obtain a linear relationship between the relative accuracy of the traffic intensity A and the measured traffic volume $A \cdot T$ when measuring in an unlimited time period. A scan interval $h = 0$ corresponds to a continuous measurement and $h > 0$ corresponds 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.*

- 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. 13.4). A continuous measurement can be considered as an optimal reference for any discrete method.

- 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, [42], 1977 [43]).

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

	Example a		Example b	
	σ_i^2	σ_i	σ_i^2	σ_i
Continuous Method				
Unlimited (13.8)	1.0000	1.0000	5.0000	2.2361
Limited	0.9500	0.9747	3.7729	1.9424
Scanning Method				
Unlimited (13.17)	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 13.2: *Numerical comparison of various measuring principles in different time intervals.*

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 (13.8) with a correct form factor than to take account of the measuring method and the measuring period.

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. (1957): A Survey of Traffic–Measuring Methods in the Copenhagen Exchanges. Teleteknik (Engl. ed.) 1957:1, pp. 92–105.

[11] 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.

[12] 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.

[13] Burke, P.J. (1956): The output of a queueing system. Operations Research, Vol. 4 (1956), 699–704.

[14] Christensen, P.V. (1914): The number of selectors in automatic telephone systems. The Post Office Electrical Engineers Journal, Vol. 7 (1914), 271–281.

[15] Cobham, A. (1954): Priority assignment in waiting line problems. Operations Research, Vol. 2 (1954), 70–76.

[16] 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.

[17] Cooper, R.B. (1972): *Introduction to queueing theory*. New York 1972. 277 pp.

[18] 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.

[19] Cox, D.R. & Miller, H.D. (1965): *The theory of stochastic processes*. Methuen & Co. London 1965. 398 pp.

[20] Cox, D.R.& Isham, V. (1980): *Point processes*. Chapman and Hall. 1980. 188 pp.

[21] Crommelin, C.D. (1932): Delay probability formulae when the holding times are constant. Post Office Electrical Engineers Journal, Vol. 25 (1932), pp. 41–50.

[22] Crommelin, C.D. (1934): Delay probability formulae. Post Office Electrical Engineers Journal, Vol. 26 (1934), pp. 266–274.

[23] 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.

[24] 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.

[25] Eilon, S. (1969): A simpler proof of $L = \lambda W$. Operations Research, Vol. 17 (1969), pp. 915–917.

[26] Elldin, A., and G. Lind (1964): *Elementary telephone traffic theory*. Chapter 4. L.M. Ericsson AB, Stockholm 1964. 46 pp.

[27] Engset, T.O. (1915): Om beregning av vælgere i et automatisk telefonsystem, en undersøkelse angaaende punkter i grundlaget for sandsynlighetsteoriens anvendelse paa bestemmelse av de automatiske centralinretningers omfang. Kristiania (Oslo) 1915. 128 pp. English version: On the calculation of switches in an automatic telephone system. Telektronikk, Vol. 94 (1998) :2 , 99–142.

[28] 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.

[29] Erlang, A.K. (1909): The Theory of Probabilities and Telephone Conversations. Nyt Matematisk Tidsskrift, B, Vol. 20, pp. 33–40 (in Danish). English translation: *The Life and Works of A.K. Erlang*, E. Brockmeyer, H.L. Halstrøm og Arne Jensen, pp. 131–137.. Copenhagen 1948

[30] Esteves, J.S. & Craveirinha, J. & Cardoso, D. (1995): Computing Erlang–B Function Derivatives in the Number of Servers. Communications in Statistics – Stochastic Models, Vol. 11 (1995) : 2, 311–331.

[31] Farmer, R.F. & Kaufman, I. (1978): On the Numerical Evaluation of Some Basic Traffic Formulae. Networks, Vol. 8 (1978) 153–186.

[32] Feller, W. (1950): *An introduction to probability theory and its applications*. Vol. 1, New York 1950. 461 pp.

[33] 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.

[34] Fredericks, A.A. (1980): Congestion in blocking systems – a simple approximation technique. The Bell System Technical Journal, Vol. 59 (1980) : 6, 805–827.

[35] Fry, T.C. (1928): *Probability and its Engineering Uses*. New York 1928, 470 pp.

[36] Gordon, W.J., and & Newell, G.F. (1967): Closed queueing systems with exponential servers. Operations Research, Vol. 15 (1967), pp. 254–265.

[37] 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.

[38] 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.

[39] Hedberg, I. (1981): A Simple Extension of the Erlang Loss Formula with Continuous First Order Partial Derivatives. L.M. Ericsson, Internal Report XF/Sy 81 171. 4 pp.

[40] ITU-T (1993): Traffic intensity unit. ITU–T Recommendation B.18. 1993. 1 p.

[41] Iversen, V.B. (1973): Analysis of real teletraffic processes based on computerized measurements. Ericsson Technics, No. 1, 1973, pp. 1–64. *"Holbæk measurements"*.

[42] 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.

[43] 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.

[44] Iversen, V.B. (1980): The A–formula. Teleteknik (English ed.), Vol. 23 (1980) : 2, 64–79.

[45] 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.

[46] 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.

[47] 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).

[48] 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.

[49] 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.

[50] Iversen, V.B. (2005): Modelling restricted accessibility for wireless multi-service systems. Lecture Notes on Computer Science, vol. 3883, pp- 93-102. Springer 2006.

[51] Iversen B.B. (2007): Reversible fair scheduling: the teletraffic revisited. Proceedings from 20th International Teletraffic Congress, ITC20, Ottawa, Canada, June 17-21, 2007. Springer Lecture Notes in Computer Science. Vol. LNCS 4516 (2007), pp. 1135-1148.

[52] Jackson, R.R.P. (1954): Queueing systems with phase type service. Operational Research Quarterly, Vol. 5 (1954), 109–120.

[53] Jackson, J.R. (1957): Networks of waiting lines. Operations Research, Vol. 5 (1957), pp. 518–521.

[54] Jackson, J.R. (1963): Jobshop–like queueing systems. Management Science, Vol. 10 (1963), No. 1, pp. 131–142.

[55] Jagerman, D.L. (1984): Methods in Traffic Calculations. AT&T Bell Laboratories Technical Journal, Vol. 63 (1984) : 7, 1283–1310.

[56] Jagers, A.A. & van Doorn, E.A. (1986): On the Continued Erlang Loss Function. Operations Research Letters, Vol. 5 (1986) : 1, 43–46.

[57] 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.

[58] Jensen, Arne (1948): Truncated multidimensional distributions. Pages 58–70 in "The Life and Works of A.K. Erlang". Ref. Brockmeyer et al., 1948 [57].

[59] 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.

[60] 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.

[61] Johannsen, Fr. (1908): "Busy". Copenhagen 1908. 4 pp.

[62] Johansen, K. & Johansen, J. & Rasmussen, C. (1991): The broadband multiplexer, "TransMux 1001". Teleteknik, English ed., Vol. 34 (1991) : 1, 57–65.

[63] 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.

[64] 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.

[65] 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.

[66] 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.

[67] Kaufman, J.S. (1981): Blocking in a shared resource environment. IEEE Transactions on Communications, Vol. COM–29 (1981) : 10, 1474–1481.

[68] Kaufman, J.S. & Rege, K.M. (1996): Blocking in a shared resource environment with batched Poisson arrival process. Performance Evaluation, Vol. 24 (1996), pp. 249–263.

[69] 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.

[70] Kelly, F.P. (1979): *Reversibility and stochastic networks*. John Wiley & Sons, 1979. 230 pp.

[71] Kendall, D.G. (1951): Some problems in the theory of queues. Journal of Royal Statistical Society, Series B, Vol. 13 (1951) : 2, 151–173.

[72] 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.

[73] Khintchine, A.Y. (1955): *Mathematical methods in the theory of queueing*. London 1960. 124 pp. (Original in Russian, 1955).

[74] Kingman, J.F.C. (1969): Markov population processes. J. Appl. Prob., Vol. 6 (1969), 1–18.

[75] Kleinrock, L. (1964): *Communication nets: Stochastic message flow and delay*. McGraw–Hill 1964. Reprinted by Dover Publications 1972. 209 pp.

[76] Kleinrock, L. (1975): *Queueing systems*. Vol. I: *Theory*. New York 1975. 417 pp.

[77] Kleinrock, L. (1976): *Queueing systems*. Vol. II: *Computer applications*. New York 1976. 549 pp.

[78] Kosten, L. (1937): Über Sperrungswahrscheinlichkeiten bei Staffelschaltungen. Elek. Nachr. Techn., Vol. 14 (1937) 5–12.

[79] Kruithof, J. (1937): Telefoonverkehrsrekening. De Ingenieur, Vol. 52 (1937) : E15–E25.

[80] Kuczura, A. (1973): The interrupted Poisson process as an overflow process. The Bell System Technical Journal, Vol. 52 (1973) : 3, pp. 437–448.

[81] 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.

[82] 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.

[83] Lévy-Soussan, G. (1968): Numerical Evaluation of the Erlang Function through a Continued-Fraction Algorithm. Electrical Communication, Vol. 43 (1968) : 2, 163–168.

[84] 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.

[85] 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).

[86] Little, J.D.C. (1961): A proof for the queueing formula $L = \lambda W$. Operations Research, Vol. 9 (1961) : 383–387.

[87] Maral, G. (1995): *VSAT networks*. John Wiley & Sons, 1995. 282 pp.

[88] Marchal, W.G. (1976): An approximate formula for waiting time in single server queues. AIIE Transactions, December 1976, 473–474.

[89] Mejlbro, L. (1994): Approximations for the Erlang Loss Function. Technical University of Denmark 1994. 32 pp. NTS–14, Copenhagen 18–20 August 1998. Proceedings pp. 90–102. Department of Telecommunication, Technical University of Denmark.

[90] Messerli, E.J. (1972): Proof of a Convexity Property of the Erlang B Formula. The Bell System Technical Journal, Vol. 51 (1972) 951–953.

[91] Molina, E.C. (1922): The Theory of Probability Applied to Telephone Trunking Problems. The Bell System Technical Journal, Vol. 1 (1922) : 2, 69–81.

[92] Molina, E.C. (1927): Application of the Theory of Probability to Telephone Trunking Problems. The Bell System Technical Journal, Vol. 6 (1927) 461–494.

[93] 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.

[94] 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.

[95] 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.

[96] Palm, C. (1947): *Table of the Erlang loss formula*. Telefonaktiebolaget L M Ericsson, Stockholm 1947. 23 pp.

[97] Palm, C. (1957): Some propositions regarding flat and steep distribution functions, pp. 3–17 in TELE (English edition), No. 1, 1957.

[98] Panken, F.J.M. & van Doorn, E.A.: Arrivals in a loss system with arrivals in geometrically distributed batches and heterogeneous service requirements. IEEE/ACM Trans. on Networking, vol. 1 (1993) : 6, 664–667.

[99] 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.

[100] 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.

[101] Rabe, F.W. (1949): Variations of telephone traffic. Electrical Communications, Vol. 26 (1949) 243–248.

[102] Rapp, Y. (1964): Planning of junction network in a multi–exchange area. Ericsson Technics 1964, pp. 77–130.

[103] Rapp, Y. (1965): Planning of junction network in a multi–exchange area. Ericsson Technics 1965, No. 2, pp. 187–240.

[104] Riordan, J. (1956): Derivation of moments of overflow traffic. Appendix 1 (pp. 507–514) in (Wilkinson, 1956 [121]).

[105] 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.

[106] Roberts, J.W. (2001): Traffic theory and the Internet. IEEE Communications Magazine Vol. 39 (2001) : 1, 94–99.

[107] Ross, K.W. & Tsang, D. (1990): Teletraffic engineering for product–form circuit–switched networks. Adv. Appl. Prob., Vol. 22 (1990) 657–675.

[108] 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.

[109] 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.

[110] 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.

[111] 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.

[112] Störmer, H. (1963): Asymptotische Näherungen für die Erlangsche Verlustformel. AEÜ, Archiv der Elektrischen Übertragung, Vol. 17 (1963) : 10, 476–478.

[113] Sutton, D.J. (1980): The application of reversible Markov population processes to teletraffic. A.T.R. Vol. 13 (1980) : 2, 3–8.

[114] Szybicki, E. (1967): Numerical Methods in the Use of Computers for Telephone Traffic Theory Applications. Ericsson Technics 1967, pp. 439–475.

[115] Techguide (2001): Inverse Multiplexing – scalable bandwidth solutions for the WAN. Techguide (The Technologu Guide Series), 2001, 46 pp. <www.techguide.com>

[116] 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.

[117] Veirø, B. (2002): Proposed Grade of Service chapter for handbook. ITU–T Study Group 2, WP 3/2. September 2001. 5 pp.

[118] 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.

[119] Wallström, B. (1964): A distribution model for telephone traffic with varying call intensity, including overflow traffic. Ericsson Technics, 1964, No. 2, pp. 183–202.

[120] Wallström, B. (1966): Congestion studies in telephone systems with overflow facilities. Ericsson Technics, No. 3, 1966, pp. 187–351.

[121] 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