Eötvös Loránd University, Budapest

Traffic Modeling of Communication Networks

Péter Vaderna

PhD dissertation

Doctoral School: Physics
Director: Prof. Zalán Horváth

Doctoral Program: Statistical Physics, Biological Physics and
Physics of Quantum Systems
Director: Prof. Jenő Kürti

Advisor: Prof. Gábor Vattay, D. Sc.

Department of Physics of Complex Systems
Eötvös Loránd University, Budapest

Budapest, Hungary
2008

Contents

Chapter 1

Introduction

The methods of statistical physics have proved to be applicable to many other areas in science where the system is not necessarily governed by physical laws nor any other laws of nature. Extensive interdisciplinary research has been formed on complex systems like sociology, evolution, vehicle transport or finance, to name a few.

The reason to get physicists involved into these research areas is the universality of the phenomena in dynamical systems. The well-known and widely studied concepts in statistical physics such as dynamics of complex systems, stochastic processes, chaotic behaviour, critical phenomena and self-organised criticality appear in the above mentioned research fields. Consequently, the methodology of analysing certain problems in these areas can be very similar to those applied in statistical physics. For example, chaotic dynamical description can be given to a system, phases and phase transitions can be explored, critical exponents can be defined, well-known models can be applied e.g. cellular automata, growth models, etc.

Communication networks [1] has recently become a widely studied research topic of physics [3], [4]. Two main research directions have been developed. The first one deals with the structural properties and topology of communication networks. It is stated that the World Wide Web [30] and the Internet [31] have scale-free property, meaning that the distribution of the number of degrees of the edges in the network follows power-law. These observations are in contrast to the previous random graph model [32], having exponential decay in the degree distribution. Topology issues are not investigated in this dissertation, a thorough summary on this area can be found in [29].

The second direction is the investigation of properties of the data traffic. Statistical characterisation and dynamical modeling of data traffic on certain network links have attracted wide interest in the recent years. Deeper understanding of the dynamics of packet transmission and the statistical description of the data traffic is essential for the design and operation of telecommunication networks. For example, traffic models serve as input for capacity management of certain network elements, topology planning or the design and optimisation of various traffic control mechanisms. Traffic modeling involves statistical description of the traffic where stationary distributions, correlation properties of the main traffic descriptors are given, as well as dynamical description of a particular system based on microscopic models.

In the following, a brief overview is given on the research issues of traffic modeling in communication networks, followed by the state-of-the-art and the scope of my research. Then after introducing the mathematical and technical background of the topic, the detailed results are presented in two parts. The first part of the dissertation focuses on the dynamical characterisation of the traffic in simplistic computer network scenarios, where it is investigated how the basic rules (network protocols) contribute to some macroscopic phenomena. In the second part matrix analytic methods are used to characterise the behaviour of a queuing model and some theoretical results on characterising a set of matrix analytic distributions are shown.

1.1 Evolution of telecommunication traffic modeling

The first report on traffic modeling in communication networks appeared in 1917 [5], where the traditional telephone networks were described. The line was busy during a call and was released after the call (circuit-switched technology). The main parameters (call holding time, frequency of the call initiations) was completely determined by human activity, resulting in random behaviour and Poisson statistics (Poisson call arrivals and exponential call holding times). The load on the telephone exchange centers was an important issue at that time, but based on the Poisson-like statistics and short tailed distributions of the essential parameters, the number of calls in a center was predictable and engineering the capacity of such exchange centers was based on relatively simple calculations. The Internet started to develop in the late 1960s from connected university Local Area Networks (LAN) and soon became worldwide, now it has major role in the business and commercial life. The spreading of data communication changed the paradigm in traffic modeling and engineering. Besides circuit-switching, packet-switched technology has appeared, where the digital information is segmented and transmitted in packets (40-1500 bytes of data).

In packet switching, the address of the source and the destination of a packet is encoded and attached to the packet as a header, separated from payload data. When a packet is sent, central network elements, so-called routers are responsible for forwarding the packet towards the next neighbour hop in the direction of the destination. The decision of which neighbour to forward to, is based on the knowledge of the source and destination information stored in the packet header. In order to achieve inter-operability between remote hosts, a standard set of rules in packet data communication called Internet Protocol (IP) is designed and used in the computers and routers [1], [2].

In contrast to the traditional telephone networks, in packet-switched networks more than one connection can be maintained by the user at a time and not only voice-calls but other applications such as file transfer or video-stream from a remote machine can be initiated. Moreover, although the data transfers themselves are initiated by users, the processes within a data transfer are driven by machines, resulting in side effects in the traffic characteristics.

The load and the capacity of the routers and switches is an important issue in data networks. In IP networks, when the amount of traffic in a router reaches the limit that it can handle, congestion occurs and packets will queue up in buffers. If congestion remains and the buffers are full, the packets arriving at the router can not be handled and will be discarded.

In 1986, the users of computer networks in US universities experienced large drop (by a factor of one thousand) in the performance, the data communication had become very slow [7]. Researchers revealed that if all computers sent their data with maximum rate, the overall performance would highly degrade due to congestion and packet drops. They proposed a solution where the hosts at the end-points were able to adapt their sending rate to the network condition [7] which was later developed further [8], [9], [10], [11], [12], [13] in order to optimise the overall performance. The set of rules describing the methods of adaptation of the rate is implemented as congestion control algorithms in the Transmission Control Protocol (TCP), first proposed in [6].

TCP only works at the end-points and has three main roles: (1) transaction setup, (2) reliable packet sending, and (3) congestion control. The transaction setup is based on specific signal packets exchanged between the end hosts to establish a connection. The reliable packet delivery relies on detecting and re-sending lost packets. It is achieved by sending acknowledgements as a response to the received packets. If a packet is not acknowledged by the receiver, then it is assumed to be lost and its transmission is initiated again by the sender. Congestion control is based on probing the network congestion level and increasing or reducing the sending rate if the load of the network is low or high, respectively. Although the basic mechanism of TCP congestion control has not been changed from 1988, its details are subject to research nowadays. The protocol is fine-tuned all the time due to the fast development and high variability of the physical environments (local area networks, satellite links, mobile networks, etc).

Many kinds of applications use the TCP/IP protocol family for data transfer, like e.g. FTP, WWW, telnet, online streaming, peer-to-peer file exchange tools, e-mail, etc. Their traffic generating mechanism varies a lot from small files (WWW) to large files (FTP, peer-to-peer) and from uniform rate (streaming) to highly variable sending rate (WWW, FTP, peer-to-peer). The heterogeneity and variability of applications and the complexity of the underlying protocols make the traffic characterisation and traffic modeling more challenging than it was in the case of traditional telephone networks.

The Internet today is a world-wide packet switched network connecting networks of universities, companies, wireline/wireless operators. Modeling communication networks is not an easy task due to its heterogeneity. New technologies emerge from time to time, Internet applications are getting more and more various, wideband mobile access is getting more and more widespread, all of these will be new challenges in traffic modeling. The characterisation of the structure and traffic in communication networks can play a major role in engineering the network, where the main goal is to achieve reliability, efficiency and security on this area.

The following section highlights the recent problems and findings in traffic modeling in computer networks.

1.2 Related work

Recent measurements [34], [40], [45] have shown that the traffic volume on a link is highly variable, long-range dependent (LRD, i.e. the autocorrelation function decays slowly) and statistically self-similar (i.e. the series of data has the same statistical properties over several timescales), which is in sharp contrast with the traditional traffic models [5]. Indeed, the Poisson model applied to telephony traffic has lost its validity when applied to the data network [34]. Packet density fluctuations are shown to have self-similar behaviour over long timescales, moreover, long-range spatial correlations of Ethernet/Internet traffic has been observed. More recently, the authors in [59] measured the traffic at the connection point of University LAN and the backbone network and found power-law fluctuations in time.

There are many research activities induced by the results of these experiments in many directions. One direction is revealing the origins and causes of the above phenomena based on statistics related to individual sources. The authors in [42] show that multiplexing large number of on-off packet train sources having long-tailed ”on” periods results in self-similar traffic. Others claim that the long-tailed distribution of file size on the servers has a crucial effect in forming self-similarity [43].

Another direction is to investigate the effect of the network protocols. It is shown that the TCP congestion control mechanism induces the LRD properties of the traffic. Chaotic nature of the deterministic protocol, TCP is demonstrated in [47], and it is pointed out that self-similarity can be generated by TCP's chaotic behaviour. The role of the congestion control algorithm of TCP in the propagation of long-range dependence and self-similarity through the Internet is reported in [48]. It is shown that TCP inherits self-similarity when mixing with self-similar background traffic. It is also demonstrated that TCP flows can pass on self-similarity to multiple hops.

Authors of [35]-[41] apply the mathematical techniques of statistical mechanics to characterise collective dynamics in large computer networks. Analysis of power spectra, correlation functions, phase transitions has been applied and phase transition between 'free-flow' phase and 'congested' phase has been reported.

Self-similarity and long-range dependence is often associated with polynomial decay in the power spectrum of the network load on a given path (that is often referred to as $1/f$ noise). It was first shown in [35] that such $1/f$ fluctuations are observable in the round-trip packet delays in computer networks between two points. Self-Organised Criticality (SOC) was introduced first in [46] to explain $1/f$ noise in systems driven by collective behaviour. A SOC model for computer network traffic is investigated in [44].

Both statistical and dynamical properties of telecommunication network traffic show similarities with highway traffic [53]. The analogies between the traffic of computer networks and highways are investigated in [49]. It has been argued that some of the common phenomena emerging in highways and computer networks can be modelled using similar cellular automaton models.

Model-based approach is often used in order to pinpoint the essential factors contributing to the main statistical properties of the system. Linear chain models [54], [50], two dimensional lattice models (e.g. square lattice models in [57], [58], [36]) and hierarchical trees in [56] are developed to investigate the dynamics of the system. A two dimensional directed model can reproduce the main characteristics of real-world Internet and vehicular traffic flows [55]. Scaling properties are found in the models that make them valid for dynamical models of the Internet and highway traffic as well.

1.3 Research scope

Most quantities in computer networks are characterised by heavy-tailed distributions, arrivals have bursty nature and the traffic rate shows self-similarity and long-range dependence. The main topic addressed by this dissertation is the approximation of heavy traffic behaviour based on Markovian models.

Although the methods in the Markovian approach are not able to catch the tail properties of real networks, Markov modeling is still a relevant method in networking for the following reasons:

- Due to finiteness of time and data, LRD and heavy-tail properties are valid only over a limited range of scales.
- Activities initiated by humans, not computers (e.g. starting of file download or a telnet session) are shown to be well approximated by Poisson statistics.
- Generalised exponential distributions can be fitted to distributions with heavy tails over finite number of timescales.
- Markovian models are easy to handle and analytically tractable.

Buffers and queues play major roles in networking thus, basic elements of queuing theory are often used throughout this dissertation. A typical queuing system is determined by an arrival process, a service time distribution and the number of parallel queues (other parameters such as buffer limitation or prioritisation are not considered here).

This dissertation contains new results in two main areas. The first one is modeling the effects of the feedback control mechanism of TCP on the network. First, back propagating waves of congestion in computer networks are investigated and the bursty nature of TCP is proved to play an important role in this phenomenon. Then, traffic models are set up based on the known sending mechanism of TCP. Most papers studied long-lived TCP connections previously where the steady-state behaviour determined the statistics. The traffic models described in this dissertation relate to short-lived TCP connections (that are typical in Web applications) where transient effects dominate and the startup behaviour of the TCP connections determine the statistical properties of the main traffic descriptors.

The second main area is the application of matrix analytical methods in queuing systems in order to approximate heavy tails and bursty behaviour. The exact transient behaviour of the queue-length moments in an infinite-server queue is computed, where both the arrival process and the service time has slow decay in the distribution for several timescales. The application of matrix analytic methods can be limited by computational capacity if large state space is used in the model. The last part of this dissertation deals with minimising the complexity of the matrix representation of a function of combined exponentials. The solution is general and can be used to optimise computation efficiency in queuing applications.

1.3.1 Traffic modeling

Transmission Control Protocol (TCP) is an end-to-end network protocol responsible for the control mechanism in computer networks in order to avoid long-lasting congestion. Since most data sources use TCP for transfer, it plays an important role in forming the typical data patterns measured on the high-speed links. The main purpose of this part is to investigate the effect of many aggregate TCP flows sharing a link, particularly, how the bursty nature of packet train affect congestion wave formation and the link saturation caused by multiple TCP flows is modelled.

1.3.1.1 The role of TCP in congestion transition

Several observations have been made on the formation and propagation of traffic jams and there is a common agreement that the front of the congestion progresses backwards against the flow of vehicles [51], [52].

There are some phenomena in computer networks indicating propagation of congestion similarly as in car traffic. In [33] the effect of routing policies on the jam transition in computer networks is investigated. The authors in [39] studied the spatiotemporal correlations of the traffic in different network hops along a path and found that congestion can propagate from a heavily loaded node towards its less loaded neighbours. Collective behaviour of network nodes and spatiotemporal forming of congestions are investigated in [44].

It is shown in this dissertation that congestion waves are formed naturally in the data traffic of computer networks. The phenomenon is analyzed in detail and it is derived that the intrinsic properties of the TCP protocol contribute to the formation and the stability of the transition of congestion in the direction opposite to the actual traffic. The large rate variation of TCP sending rate (burst effect) is pinpointed as one of the major contributors of this phenomenon.

This mechanism is checked in a computer network with simple topology realised in a network simulator. Various scenarios are presented with different amount of traffic rate variations and clear dependence is pointed out between congestion transition and burstiness. Nevertheless, this basic mechanism is quite general and can create congestion moving against the direction of the data traffic in more complicated geometries as well. These results are published in [98].

1.3.1.2 Modeling short TCP connections

With the domination of TCP/IP based applications such as Web browsing one can encounter many challenges. In research on network engineering, the description of practical issues like the statistical properties of feedback controlled traffic or network utilisation with Web-like conditions is very important. Most of the studies on traffic preceding my work had considered persistent sources in the case of a fixed number of parallel connections [16], [17], [22].

However, in reality understanding situations where Web users request a file transfer at random inter-arrival times is a very important problem. In such traffic scenario the file transfers are typically short and the transient effects dominate over the whole transaction.

First, I propose a simple model of the TCP connection dynamics. It is shown that the model is capable to describe the steady-state behaviour of the number of parallel connections while the individual connections are in transient state. Short file transfers and low packet loss probability keep the TCP connections in their initial phase (known as slow start) over the download periods. Based on the transient properties the utilisation of a link shared by many TCP flows is computed which then serves as a basis for a Markov model of the number of parallel sessions in the system. This Markov model of parallel short TCP connections can be applied in a more general framework incorporating the network topology as well, as it is shown in [102].

Then I set up a model of transferring a Web page where the files are typically short but one page consists of more than one file. The performance (average transfer rate and download time) is computed in two cases: (1) the short files are downloaded sequentially, each of them opening new TCP connections; (2) one TCP connection remains open for the whole download of the page, that is supported by certain versions of HTTP protocol. In this case, the results are applicable for heavy-tailed file-size distribution as well. The model of short sequential TCP connections is published in [103].

1.3.2 Matrix analytic methods

In real-world data communication systems, the properties of the main traffic parameters such as file size, duration of data transfers, time between packet arrivals do not follow exponential distributions, but they are better described by general distributions, often with heavy tails.

However, some studies suggest to be careful with applying heavy-tails in real networking systems. Authors in [60] argue that in the engineering point of view the waist of the distribution is more important than the tail behaviour. Investigations on the correlation structure of TCP [61] revealed that the TCP protocol generates traffic where self-similarity is valid only for a finite range of timescales. These results seem to suggest that Markovian models can still be appropriate to describe network traffic. Moreover, queuing systems can be much more easily analysed if the service time and the inter-arrival time is exponentially distributed ($M/M/$ type queues). A feasible approach to handle general distributions in queuing systems is to apply approximations using a combination of exponentials.

The class of Phase-type (PH) distributions introduced in [62] and [63] is the generalisation of exponential and Erlang distributions, since it includes all mixtures and convolutions of exponentials. The set of PH distributions is part of matrix-exponential distributions, so calculations with them can be made by using generalised exponential functions with matrices in the exponent. Using PH distributions in queuing theory, complex systems can be handled analytically. Moreover, the set of PH distributions is proved to be dense in the set of nonnegative distributions, implying that any general distributions can be approximated by PH distributions. Certainly, using Markovian approach for modeling general queuing systems has its cost. The higher the accuracy one would like to achieve, the more complex the approximating PH distribution should be and the more computational capacity one needs. Since PH is a combination of exponentials, its tail behaviour will always be exponential. However, by choosing sufficiently large number of components and appropriate parameter settings, the slow decay can be constructed by PH functions for several orders of magnitude, naturally followed by exponential decay in the end.

While the continuous exponential distribution can be generalised by PH (see above), the class of Markov Arrival Processes (MAP) or an even more general class, Batch Markovian Arrival Processes (BMAP) is the generalisation of the Poisson point process. MAPs are useful in modeling burst-like arrivals. Like PH distributions, they can also be handled with matrix-exponential functions. These properties make them useful for approximating LRD behaviour over several scales and also analytically tractable.

Matrix analytic methods can be applied to approximate general models with matrix exponential functions. One possible method for mapping a Markovian model to a general one is to use moment matching algorithms [78]. Another possible approach is fitting the distribution step-by-step, using different exponential components [64]. A method based on Expectation Maximisation (EM) algorithms is used in [69] and [68]. Other models based on Markov Modulated Poisson Process (MMPP) [74] are introduced in e.g. [75], [76], [77]. Fitting tools to model network traffic with PH distributions are used in [65], [66]. MAP fitting methods can be found in [67].

1.3.2.1 Transient behaviour of infinite-server queuing systems

The main objective is to investigate the MAP/PH/∞ queuing model. Numerical methods exist to approximate the moments of the queue-length for the PH/G/∞ system [70] which can be extended to the more general MAP/G/∞ system. However, these solutions rely on the numerical solution of a set of differential equations. Further, authors in [79] derive numerically tractable formulae for the moments of BMAP/PH/∞ system, which still contain elements that can only be obtained approximately.

A different computational method is presented here where the time-dependent moments of the queue length of MAP/PH/∞ system can be obtained exactly. This model can be used to describe transient behaviour of systems with parallel servers, general arrival and general processing times [99], [100].

1.3.2.2 Minimising complexity in matrix analytic functions

The structure of the Phase-type (PH) distributions is analysed here. A PH distribution can be seen as a composition of exponential distributions and also as a special case of matrix-exponential functions that can be represented by matrices. The objective is to minimise the complexity of the matrix representation of a set of PH distributions. The focus is on upper triangular representations i.e. matrices having only upper diagonal elements.

The target set of functions is given as

$$f(t) = \sum_{i=1}^{3} \eta_i \lambda_i e^{-\lambda_i t},$$

where λ_is are different real fixed coefficients in the exponent $(\lambda_1, \lambda_2, \lambda_3 > 0)$ and η_is are parameters. Since $f(t)$ is a probability density function, $\eta_3 = 1 - \eta_1 - \eta_2$ should hold. This set of functions has 2 free parameters (η_1 and η_2) thus, one particular function can be mapped to a point in the 2 dimensional parameter space. The goal is to find the subset of the parameter space where the functions can be represented by 3 dimensional matrices. The method of characterising the PH distributions is based on the invariant polytope approach introduced in [80].

I found that for 3 distinct real coefficients in the exponent, the set of functions that can be represented by 3 dimensional upper triangular matrices is a triangle (polytope with 3 vertices) in the parameter space (η_1, η_2). Based on the construction of the polytope, the representation matrix for the distributions inside the triangle is given. The distributions outside the triangle can not be represented by 3 dimensional matrices. Moreover, a recursive decomposition of the set of the distributions into subsets according to their minimal order upper triangular PH representations can be given.

I generalised the case of 3 exponents to N exponents $(N > 3)$. The polytope with N vertices in the $N - 1$ dimensional space containing the functions represented by N dimensional matrices can be derived from the corresponding case of $N - 1$ exponents. The results on this research area are published in [101].

Chapter 2

Mathematical background

2.1 Basic definitions

In this section definitions and theorems are summarised that are necessary for the analysis later on. Only the most important statements are mentioned here, for more details and the proofs of the theorems see the textbooks on probability theory and queuing theory (e.g. [94], [95]).

A point process (or counting process) contains discrete events at certain times. There is a sequence of non-negative random variables (usually called time variables):

$$0 = t_0 < t_1 < \ldots < \infty$$

Then N_t =max$\{n, t_n \leq t\}$ is a *point process*, that often corresponds to the 'number of arrivals'. The process of inter-arrival times is $T_n = t_n - t_{n-1}$. If the variables T_n are independent and identically distributed (i.i.d.) then $\{t_n\}$ is called *renewal process*. $\{N_t, t \geq 0\}$ is the associated renewal point process.

A widely used point process is the *Poisson process*, that is characterised by the following properties:

- It has stationary and independent increments
- $P(N_{t+\Delta t} - N_t = 1) = \lambda \Delta t + o(\Delta t)$
- $P(N_{t+\Delta t} - N_t \geq 2) = o(\Delta t)$

where $\lambda > 0$ and $o(x)$ is a function with the property that $\lim_{x\to 0} o(x)/x = 0$. The Poisson process has the following important properties:

Theorem 2.1.1. *For a Poisson process, the probability distribution function (p.d.f.) of the number of events within a time interval of length Δt is*

$$P(n \textit{ events in } ([t, t + \Delta t])) = \frac{(\lambda \Delta t)^n}{n!} e^{-\lambda \Delta t}$$

Theorem 2.1.2. *The distribution of the inter-arrival times of Poisson process is exponential.*

There is another type of processes frequently used in probability theory and its applications. Let X be a random variable. Then introducing the auxiliary variable t, $Y_X(t) = f(X, t)$ function of X and t is called a *stochastic process*. In most applications t is the time variable. Continuous-time and discrete-time stochastic processes can be distinguished, depending on that t is real or integer. If the stochastic variable X is fixed to a value x then $Y_x(t) = f(x, t)$ as an ordinary function of t is called the realisation of the process. Given the p.d.f. $P_X(x)$ of the variable X the p.d.f. of $Y_X(t)$ can be written as

$$P_1(y, t) = \int dx \delta[y - f(x, t)] P_X(x)$$

where $\delta(\cdot)$ is the Dirac-delta function. Also the joint p.d.f. that $Y_X(t)$ takes values y_1 at time t_1, y_2 at time t_2, etc, up to n is

$$P_n(y_1, t_1; ...; y_n, t_n) = \int dx \delta[y_1 - Y_X(t_1)] \cdots \delta[y_n - Y_X(t_n)] P_X(x)$$

The index n indicates the number of variables in the joint p.d.f. The stochastic process is *stationary* if the following equation holds for $\tau \in \mathbb{R}$:

$$P_n(y_1, t_1 + \tau; \cdots ; y_n, t_n + \tau) = P_n(y_1, t_1; \cdots ; y_n, t_n)$$

i.e. P_n depends only on the differences of time values. The conditional probability $P_{m|n}(y_{n+1}, t_{n+1}; \cdots ; y_{n+m}, t_{n+m} | y_1, t_1; \cdots ; y_n, t_n)$ is the probability that the process takes values y_{n+1} at $t_{n+1}; \cdots ; y_{n+m}$ at t_{n+m} given that the process has taken y_1 at $t_1; \cdots ; y_n$ at t_n before. It can be calculated by the Bayes' rule:

$$P_{m|n}(y_{n+1}, t_{n+1}; \cdots ; y_{n+m}, t_{n+m}|y_1, t_1; \cdots ; y_n, t_n) = \frac{P_{n+m}(y_1, t_1; \cdots ; y_{n+m}, t_{n+m})}{P_n(y_1, t_1; \cdots ; y_n, t_n)}$$

A very important type of stochastic processes is the Markov-process. A stochastic process is called *Markov-process* if the conditional probabilities satisfy the following equation ($t_1 < t_2 < \cdots < t_n$).

$$P_{1|n-1}(y_n, t_n|y_1, t_1; \cdots ; y_{n-1}, t_{n-1}) = P_{1|1}(y_n, t_n|y_{n-1}, t_{n-1}),$$

i.e. the conditional probability of a Markov process being y_n at t_n provided that it was y_{n-1} at t_{n-1} is completely determined and does not depend on the values at previous times.

Thus, the Markov process is completely determined by $P_1(y, t)$ and the *transition probability* $P_{1|1}(y', t'|y, t)$. The joint distributions $P_n(y_1, t_1; \cdots ; y_n, t_n)$ can be constructed from the initial p.d.f. and the transition probability.

An important equation for the transition probabilities is the Chapman-Kolmogorov equation (the notation $1|1$ is omitted here):

$$P(y_3, t_3|y_1, t_1) = \int dy_2 P(y_3, t_3|y_2, t_2)P(y_2, t_2|y_1, t_1), \quad (t_1 < t_2 < t_3)$$

In the following only those Markov processes are investigated where $X_t \in \mathbb{Z}$. This type of process is called *Markov chain* and the discrete set of X_ts is called state set. In this case the transition probability from state i to state j is

$$P_{ij} = P(j, t_{n+1}|i, t_n), \quad i, j \in \mathbb{Z} \quad n \geq 0$$

The state i is said to be *absorbing*, if $P_{ii} = 1$ and $P_{ij} = 0$ for all j. The *return time* of a state is defined as

$$T_i = inf\,\{n : x_n = i|x_0 = i\}$$

The state i is said to be *transient* if the probability that the return time is infinite is positive:

$$Pr(T_i = \infty) > 0$$

The Markov chain is *positive recurrent*, if all of its states has finite expected return time. The mth transition probability i.e. the probability that starting from state i the system is in state j after m steps can be written as

$$P_{ij}^m = P(j, t_{n+m} | i, t_n), \quad i, j \in \mathbb{Z} \quad n, m \geq 0$$

If the nth transition probability matrix is denoted by

$$P^{(n)} = [P_{ij}^n],$$

then Chapman-Kolmogorov equation can be given in the form of matrix multiplication (it is possible that the matrices are of infinite order):

$$P^{(n+m)} = P^{(n)} P^{(m)}$$

State j is called *reachable* from i if $P_{ij}^{(n)} > 0$ for some n. If i is reachable from j and j is reachable from i ($\forall i, j$) then the Markov chain is *irreducible*.

If the probability that the system is in state j in time t is denoted by $\pi_j(t)$ ($\pi_j(t) = P(j, t)$) then the following balance equation holds (master equation for discrete state set):

$$\frac{d\pi_j(t)}{dt} = \sum_{i(i \neq j)} \pi_i(t) \lambda_{ij} - \pi_j(t) \sum_{i(i \neq j)} \lambda_{ji} \quad j \in \mathbb{Z},$$

where $\lambda_{ij}(i \neq j)$ denotes the transition probability per unit time from state i to state j. Let the matrix $Q = [q_{ij}]$ be defined as

$$q_{ij} = \begin{cases} -\sum_{k(k \neq i)} \lambda_{ik} & \text{if } i = j \\ \lambda_{ij} & \text{if } i \neq j \end{cases}$$

It is called *transition rate matrix* or *infinitesimal generator matrix*. The matrix Q

has the following properties:

- its row sum is zero;
- its diagonal elements are non-positive;
- its off-diagonal elements are non-negative;

With the infinitesimal generator matrix the master equation can be written in a simple form:

$$\frac{d\pi}{dt} = \pi Q$$

The following theorem assures that Markov chains with certain properties are guaranteed to have a unique steady-state solution.

Theorem 2.1.3. *If the system is ergodic (i.e. it consists of one irreducible chain and it is positive recurrent) then* $\pi_j = \lim_{t\to\infty} \pi_j(t)$, $j = 0, 1, \ldots$ *exist.*

The following equation holds for the steady state probabilities:

$$\pi Q = 0 \quad \pi \mathbf{e} = 1,$$

where $\mathbf{e}$ is a column vector of proper dimension whose all elements are one.

Finally, the concept of *queuing* is introduced. Queuing theory is a mathematical theory investigating the behaviour of queues and waiting lines. Typically there is one or more (possibly infinite) service points where demands arrive according to fluctuating process. The service time of the demands at the service points also fluctuates. Usually the demands arriving to a server is either being served or put in a queue if the server is occupied by another demand. Examples for queuing systems are e.g. waiting people standing in lines, waiting cars in highways or buffered data packets in telecommunication networks.

Different queuing disciplines can be used e.g. 'First Come First Serve' (FCFS), 'Last Come First Serve' (LCFS) or 'Service in Random Order' (SIRO). Various operations and limitations can be specified in a queuing model e.g. prioritising demands, limiting buffer sizes, limiting the population of demands, etc. In this dissertation the FCFS queuing discipline is used and no limitations and proiritisations take place.

The main questions regarding the behaviour of a queuing system are the statistical properties of the queue length, the length of time when no demands are being served, the number of running parallel servers etc.

For the classification of the queuing models abbreviations are used known as Kendall-Lee notations. The abbreviations are separated by slashes. The first and the second term corresponds to the arrival and service processes, respectively, while the third term indicates the number of parallel servers. The other terms may indicate the queuing discipline and the capacity and population limitations but these terms are omitted in this study.

The most widely used queuing model is the one with Poisson arrivals and exponential service time, denoted by $M/M/n$, where M indicates Poisson arrivals and exponential service times and n is the number of parallel servers. The general queuing model is indicated by $G/G/n$, where both the arrival process and the service time distribution is general.

2.2 Phase-type distributions

Consider a continuous-time Markov chain with $n+1$ states, where the $(n+1)$st state is absorbing. The transition rate matrix of such a Markov chain can be described by

$$\mathbf{T} = \begin{bmatrix} \mathbf{S} & \mathbf{S}^0 \\ \mathbf{0} & 0 \end{bmatrix},$$

where $\mathbf{S}$ is a non-singular $n \times n$ matrix and $\mathbf{S}^0$ is an n-vector such that $[\mathbf{S}]_{ij} \geq 0$, $[\mathbf{S}]_{ii} < 0$, $[\mathbf{S}^0]_i \geq 0$ for all $i \neq j$, $(1 \leq i, j \leq n)$ and $\mathbf{Se} + \mathbf{S}^0 = \mathbf{0}$, where $\mathbf{e}$ is a row n-vector consisting of ones and $\mathbf{0}$ is a column vector consisting of zeroes. The meaning of $[\mathbf{T}]_{ij}$ $(i \neq j)$ is the transition rate (i.e. the average number of transitions in a time unit) from the ith to jth state. The $(n+1)$st row of $\mathbf{T}$ is zero, indicating that the $(n+1)$st state is absorbing. The initial distribution is determined by the probability vector $[\boldsymbol{\alpha}, \alpha_{n+1}]$, where $0 \leq \alpha_i \leq 1$ for all i

$(1 \leq i \leq n+1)$ and $\boldsymbol{\alpha}\mathbf{e} + \alpha_{n+1} = 1$. The distribution of the time until absorption is called Phase-type (PH) distribution.

For quantitative description of PH distributions, let $p_i(t)$ denote the probability that the process is in state i at time t $(1 \leq i \leq n+1)$. According to the master equation the system is characterised by the following set of differential equations:

$$\frac{d\mathbf{p}(t)}{dt} = \mathbf{p}(t)\mathbf{T}$$

with initial condition $\mathbf{p}(0) = [\boldsymbol{\alpha}, \alpha_{n+1}]$. The equation system has the solution

$$\mathbf{p}(t) = [\boldsymbol{\alpha}, \alpha_{n+1}] \exp(\mathbf{T}t).$$

The probability vector $\mathbf{q}(t)$ that the process is in one of the transient states $1, \ldots, n$ at time t is given by

$$\mathbf{q}(t) = \boldsymbol{\alpha} \exp(\mathbf{S}t).$$

Thus, the probability distribution function of the time until absorption is

$$F(t) = 1 - \boldsymbol{\alpha} \exp(\mathbf{S}t)\mathbf{e}.$$

The corresponding probability density function is

$$f(t) = -\boldsymbol{\alpha}\mathbf{S} \exp(\mathbf{S}t)\mathbf{e}.$$

If $\boldsymbol{\alpha}\mathbf{e} < 1$ (or equivalently $\alpha_{n+1} > 0$) is allowed, then in order to make $f(t)$ a probability density function a weight in 0 has to be added so that $\int_0^\infty f(t)dt = 1$. These probability density functions of PH distributions make a vector space.

The 2-tuple $(\boldsymbol{\alpha}, \mathbf{S})$ is called the representation of the PH distribution. The representation is not unique for a given PH distribution. The less the number of states in a representation the less the complexity of computation will be thus, finding PH representations with minimal number of states is an important issue in applications. The number of states in the underlying Markov chain is called the order of the representation. The order of representation having minimal number of states is called the order of the PH distribution.

PH distributions have the following useful properties:

- The set of PH distributions is dense in the set of nonnegative distributions.
- The mixture of 2 PH distributions is also PH.
- The convolution of 2 PH distributions is also PH.

If the system is assumed to immediately restart after getting into the $(n+1)$st state again according to the probability vector $\boldsymbol{\alpha}$, then the point process associated with absorbings, with PH-distributed variable between arrivals, is called PH renewal process. It is shown in Section 2.3 that PH renewal processes are a special case of Markov Arrival Processes (MAP).

There are various forms of interpretation of the PH distribution, including

- probability density and cumulative density functions,
- matrix-exponential representations,
- Laplace transforms,
- polytopes in the vector space of functions.

2.3 Markov Arrival Processes

Consider a continuous-time Markov-chain $J(t)$ with n states. The transition rate matrix is denoted by $\mathbf{D}$. Let's suppose that the transitions of the Markov-chain are associated with arrivals in a point process. Moreover, there are arrivals associated with staying in certain states where the arrival rate depends on that particular state. The arrival process is thus governed by the states and state transitions of the Markov-chain.

To describe the process, $\mathbf{D}$ is decomposed into 2 matrices as $\mathbf{D} = \mathbf{D}_0 + \mathbf{D}_1$ where $\mathbf{D}_0$ and $\mathbf{D}_1$ are matrices of order n with the following properties: $\mathbf{D}_1 \geq 0$ (i.e. all elements of $\mathbf{D}_1$ are nonnegative), $[\mathbf{D}_0]_{i,j} \geq 0$ for $1 \leq i \neq j \leq n$, $[\mathbf{D}_0]_{i,i} < 0$ for $1 \leq i \leq n$ and the matrix $\mathbf{D} = \mathbf{D}_0 + \mathbf{D}_1$ is stochastic, that is, $\mathbf{De} = \mathbf{0}$. Matrices $\mathbf{D}_0$ and $\mathbf{D}_1$ filter those parts of the Markov process which correspond to non-arrival and arrival transitions, respectively.

- The non-diagonal elements of $\mathbf{D}_1$ refer to all phase transitions that generate arrivals.

- The diagonal elements of $\mathbf{D}_1$ refer to all arrivals without phase transitions.

- The non-diagonal elements of $\mathbf{D}_0$ refer to all phase transitions that do not generate arrivals.

- The diagonal elements of $\mathbf{D}_0$ should be set so that $\mathbf{De} = 0$.

A natural generalisation of MAP is to allow more than one arrivals which requires the definition of $\mathbf{D}_m$ accordingly, for m-sized batches.

A special case of MAP can be derived by letting only the diagonal elements of $\mathbf{D}_1$ be nonzero, which basically means that no state transitions occur at the time of arrivals. This results in the Markov Modulated Poisson Process (MMPP), which is a widely used arrival model [74].

Another special case of MAP is the Phase-type (PH) renewal process. In this case, the arrivals occur according to a renewal process where the time between the arrivals has PH distribution represented by the $(\boldsymbol{\alpha}, \mathbf{S})$ pair (see Section 2.2). The vector $\boldsymbol{\alpha}$ is the initial probability vector of the PH distribution while $\mathbf{S}$ is a non-singular matrix describing the phase transitions until the absorption such that $[\mathbf{S}]_{i,j} \geq 0$ and $[\mathbf{S}]_{i,i} < 0$ for all i ($1 \leq i \neq j \leq n$), and $\mathbf{Se} \leq 0$. In this case the phase process is restarted after an arrival according to the initial probability vector $\boldsymbol{\alpha}$ and the arrival rate that depends on the phase is described by vector $\mathbf{S}^0$:

$$\mathbf{S}^0 = -\mathbf{Se}.$$

The MAP representation in this case is

$$\mathbf{D}_0 = \mathbf{S}, \quad \mathbf{D}_1 = \mathbf{S}^0\boldsymbol{\alpha}.$$

One can consider the counting process and the underlying Markov-chain as a continuous time Markov-chain $\{(N(t), J(t))\}$ $(t \geq 0)$ where $N(t)$ is the number of arrivals in time $(0, t]$. The infinitesimal generator matrix $\mathbf{Q}$ of the Markov-process has the following form:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{D}_0 & \mathbf{D}_1 & \mathbf{0} & \mathbf{0} & \dots \\ \mathbf{0} & \mathbf{D}_0 & \mathbf{D}_1 & \mathbf{0} & \dots \\ \mathbf{0} & \mathbf{0} & \mathbf{D}_0 & \mathbf{D}_1 & \dots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{D}_0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix},$$

The row-blocks indicate the number of arrivals so far and the position inside the block indicates the phase. The Markov-chain described above is divergent since only arrivals are counted. If one would like to involve it in a queuing system, service rates can be included, too. E.g. consider a $MAP/M/\infty$ queuing system where the infinitesimal matrix is the following (using the notations introduced in [62]):

$$\mathbf{Q} = \begin{bmatrix} \mathbf{D}_0 & \mathbf{D}_1 & \mathbf{0} & \mathbf{0} & \dots \\ \mathbf{A}_2 & \mathbf{A}_1 & \mathbf{A}_0 & \mathbf{0} & \dots \\ \mathbf{0} & \mathbf{A}_2 & \mathbf{A}_1 & \mathbf{A}_0 & \dots \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_2 & \mathbf{A}_1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}.$$

Here when the queue is empty, $\mathbf{D}_0$ refers to the phase transitions that do not generate arrivals and $\mathbf{D}_1$ refers to the arrivals as earlier. When the queue is not empty, the matrices $\mathbf{A}_0$, $\mathbf{A}_1$ and $\mathbf{A}_2$ govern the phase transitions, arrivals and departures. The matrix $\mathbf{A}_2 = \mu\mathbf{I}$ refers to the departures ($\mathbf{I}$ is the unity matrix of proper dimension), $\mathbf{A}_0 = \mathbf{D}_1$ refers to the arrivals, and $\mathbf{A}_1 = \mathbf{D}_0 - \mathbf{A}_2$ refers to the phase transitions that do not generate arrivals, nor departures. The transition rate matrix is the sum of the $\mathbf{A}$ matrices: $\mathbf{D} = \mathbf{D}_0 + \mathbf{D}_1 = \mathbf{A}_0 + \mathbf{A}_1 + \mathbf{A}_2$. If $\mathbf{Q}$ is an irreducible Markov chain, then the stationary distribution $\boldsymbol{\pi}$ such that $\boldsymbol{\pi}\mathbf{Q} = 0$ can be found in the form of

$$\boldsymbol{\pi} = [\boldsymbol{\pi}_0\mathbf{R}, \boldsymbol{\pi}_0\mathbf{R}^2, \ldots],$$

where $\mathbf{R}$ is the minimal non-negative solution of the equation

$$\mathbf{R}^2\mathbf{A}_2 + \mathbf{R}\mathbf{A}_1 + \mathbf{A}_0 = 0$$

and $\boldsymbol{\pi}_0$ is the solution of the set of equations

$$\boldsymbol{\pi}_0(\mathbf{D}_0 + \mathbf{R}\mathbf{A}_2) = \mathbf{0}$$

$$\boldsymbol{\pi}_0(\mathbf{I} - \mathbf{R})^{-1}\mathbf{e} = 1$$

More details can be found in [62].

Chapter 3

Preliminaries on telecommunication

3.1 Communication protocols

In packet switched networks the communication between hosts is achieved by protocols. There are various tasks to be solved by protocols such as forwarding the data packets to their destination, assuring security and reliability, providing efficient transfer, etc. In order to satisfy these requirements, different standards have been made defining protocol suites. The protocols are built on each other, forming layers. The dominant layered architecture model is the TCP/IP model [96].

This analysis focuses on the standardised TCP/IP protocol stack. The layers of the TCP/IP standard are the following (in bottom-up direction):

- Link layer or media access layer, providing the physical links and interfaces.
- Network layer, responsible for carrying data from source to destination. In most cases Internet Protocol (IP) is used in the network layer. IP provides a connectionless service, it does not guarantee end-to-end delivery.
- Transport layer, providing end-to-end communication services for applications. The two primary protocols used in transport layer are Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). TCP provides a connection-oriented reliable service and flow control. UDP is a connectionless transport service.

- Application layer, providing application-level communication. Examples for application layer protocols are Telnet for remote login, FTP for file transfer, HTTP for Web browsing or SMTP for mail delivery.

The majority of the applications needs reliable data transfer so they use TCP as transport protocol, therefore TCP has an important role in traffic characteristics. Three main features of TCP are mentioned here:

- It maintains connection between end-hosts that is achieved by exchanging signal packets to establish and to close a connection.
- It provides reliable transfer that is achieved by
  - acknowledging all packets at the receiver side;
  - putting the packets in the right order at the receiver side if they are re-ordered;
  - re-sending lost packets at the sender side;
- It controls network congestion that is achieved by adapting the sending rate to the actual network capacity, meanwhile trying to reach the appropriate rate as soon as possible;

The main results of the first part of the dissertation focus on the congestion control mechanism of TCP. The following sections highlight the most commonly used algorithms of congestion control and also summarise the TCP traffic models.

3.2 TCP congestion control

The aim of TCP congestion control is to adapt the sending rate to the network conditions. It is achieved by increasing the sending rate when the available capacity is sufficient and reduce it when there is traffic congestion in the network. In order to achieve efficiency and fairness towards other competing TCPs, the mechanism of adaptation is based on the AIMD model (Additive Increase, Multiplicative Decrease)[1] introduced first in [97].

[1] The starting phase of TCP (slow start) does not follow the AIMD model.

After establishing connection between two computers over the network data packets start to be delivered. The algorithm implemented in TCP regulates the packet-sending rate in the following way. First a single packet is sent out. Upon receiving that packet the receiver acknowledges the arrival of the packet by sending back a small size acknowledgement packet (ACK). The time elapsed between the sending out of a packet and receiving the corresponding ACK is called round-trip time (RTT). The TCP maintains an internal variable, the congestion window (w), which is used to control the number of packets sent out when the ACK is received. It starts with the initial value $w = 1$ and then it is increased according to the following policy.

In the starting phase, $w \mapsto 2w$ each time an ACK arrives. As a result, the number of unacknowledged packets in the network doubles in a round-trip time. The algorithm of the starting phase is called 'slow start', the term 'slow' refers to the low initial window ($w = 1$) however, w increases exponentially every RTT in this phase. It continues until w reaches a threshold.

After that the window increases as $w \mapsto w + 1/w$ each time an ACK is received. Two new packets are sent out if the congestion window crosses an integer value and only a single packet otherwise. This way the integer part of the window $[w]$ gives the number of sent but not yet acknowledged packets in the network. This process lasts until a packet is lost somewhere in the network, indicating congestion. As a response the packet-sending rate should be decreased. So, the TCP reduces the value of the congestion window $w \mapsto \frac{1}{2}w$ and does not send out any new packets in response to ACKs until the number of still unacknowledged packets decreases to the integer part of the new (reduced) value of the congestion window. After that the packet-sending algorithm returns to the original linear increase phase described above. The second phase of TCP sending is called 'congestion avoidance'.

There is also a possibility for the receiver to set an upper limit of the number of unacknowledged packets kept out in the network, called Advertised window. Setting this parameter aims at protecting the receiver from overload.

There are some other mechanisms in TCP not detailed here, such as timeout indicating that all packets or ACKs are lost, three-way handshake (exchanging signals such as SYN, SYNACK, ACK) preceding the file transfer, closing the connection by FIN packets, etc. These mechanisms are not considered in the analysis because the initial and final signals have small contribution to the traffic and the packet loss is assumed to be small so that the probability of losing w packets is low.

3.3 TCP modeling

Modeling the behaviour of TCP has been a relevant research issue in the last decades. TCP models can be set up in order to evaluate the performance of different implementations, each using different congestion control methods. Usually the main performance measures of interest are the average data rate of a TCP connection (throughput) and the time of the transfer (latency). The parameters determining these performance measures are either constant parameters set by the protocols (e.g. maximum segment size (MSS), the receiver's advertised window (W_m) or variable parameters describing the network properties such as round-trip time (RTT) or packet loss probability (p)). In [16] a simple model is set up assuming long-living TCP transfers, constant RTT, periodic loss and constant p. According to the model the throughput is proportional to MSS and inversely proportional to RTT and the square root of p (K is a constant value):

$$T = \frac{MSS}{RTT}\frac{K}{\sqrt{p}}$$

Several refinements of the above model have been published where other mechanisms are also taken into account such as random loss, loss indication with timeout (in the case when the whole window is lost) [17], [18], [15] and connection establishment and slow start phase [22], [23].

The validation of the models can be performed in different ways. One way is to make measurements in live networks either actively (i.e. generate own traffic and measure its performance) or passively (i.e. measure the traffic by tapping a line without any intervention in the real traffic). Another way of validation is using network simulator tools, that have been developed in order to assist in investigating various network setups. These tools enable one to assemble any computer network configuration, to use the most commonly used packet sending mechanisms in TCP/IP protocol and to simulate its real behaviour without building the system from hardware components. These simulators can imitate the behaviour of hardware elements (computers, routers, lines etc.) accurately so that the results of the simulations are nearly identical with those obtained from measurements. One of the most popular tools is the Berkeley Network Simulator [90], which was used in this study.

Part I

Traffic modeling

Chapter 4

The role of TCP in congestion transition

In this chapter network traffic is studied where it is generated in a unidirectional ring of identical routers connected to each other. Ring topology was chosen to investigate to have some analogy with other granular flow simulations. The ring geometry mimics periodic boundary conditions. This way the propagation of congestion in an isolated, clean setup can be studied, where the effects of inhomogeneity and the complex topology of the real Internet does not interfere with the basic mechanism creating the congestion wave. It is shown that this system drives itself in a self-organised way into a critical congested state, where the system is overloaded and packets are lost for a long period of time. Both the position of the congested router (where packets are dropped) and the profile of the rate of the packet sending activity (number of packets sent by the sources in unit time) at the sites propagate against the direction of the packet flow. The profile of the congestion wave can be reconstructed from the activities of the computers connected to the ring. It is shown that the propagation of congestion is in strong relationship with both congestion control mechanism in the transport protocol (TCP) and bursty nature of the traffic flow coming out of the data sources. The effect of bursts is investigated in different network simulation scenarios.

The speed of the congestion wave is highly dependent on many parameters of the network. This dissertation focuses on the possible reasons of the phenomenon, computation of the propagation speed is out of scope in this document. The observations are verified by network simulations.

4.1 The model

Figure 4.1 shows the network setup. In the model system a ring is formed by N identical routers, which can forward packets in clockwise direction.

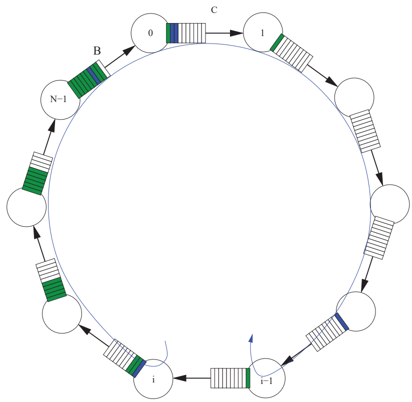

Figure 4.1: The ring structure. In the simulations a set of parameters typical for the real Internet has been set ($C = 10^7$ bit/s, $\tau = .031$ s, $B = 300$ packets, $P = 4416$ bits, $N = 10$).

To each router a terminal computer is attached to generate the traffic. Routers are connected with a line of capacity C (measured in data bits per second) with a constant forwarding delay τ (measured in seconds). Incoming data flow in a router, which is a mixture of packet flows injected by the terminal computer and the background traffic coming from the neighbouring router, can temporarily exceed the capacity of the outgoing line. To avoid data loss in this situation the router contains a buffer of size B (measured in data packets) where packets can be stored. Terminal computers are instructed to send data persistently to their anti-clockwise neighbors, so that the packets traverse the longest possible route in the ring. The traffic studied is ”granular” as computers send data packets of size P (measured in bits). The dynamics of the data traffic of computers is controlled by the TCP protocol. This protocol ensures that the data packet-sending rate is decreased whenever congestion occurs and that it is increased when there is available unused capacity in the system.

4.2 Observed phenomenon

The results of the simulation study carried out with the network simulator are presented. The geometry and parameters of the setup is shown in Figure 4.1. In this simulation scenario only 1 connection is established between one node and its anti-clockwise neighbour, so each terminal sends one flow and receives one flow. Figure 4.2 shows the spatiotemporal diagram of the congestion wave occurring in the network simulator. The horizontal axis is the time (covering 3600 seconds) and the site index (i) is on the vertical axis. In this simulation the number of sites was $N = 10$. Note that the sites $i = 0$ and $i = N - 1$ are neighbors in the ring topology. The shade of the figure represents the buffer size B_i. Dark patches indicate very large buffer size due to high level of congestion. It can be seen that the most congested site propagates in anti-clockwise direction with almost constant speed, while the packet traffic itself is clockwise directed.

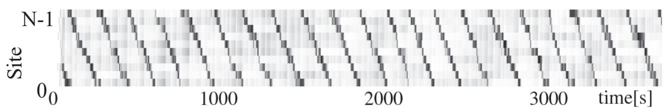

Figure 4.2: Spatiotemporal diagram of congestion propagation (buffer usage).

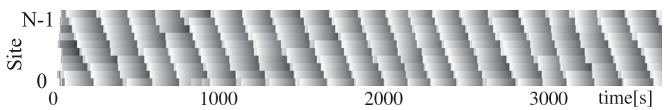

Figure 4.3: Spatiotemporal diagram of congestion propagation (congestion window).

According to the TCP protocol, the sending rate of each individual source is determined by the congestion window (w) maintained for each flow at the senders.

The value of w is the number of packets sent into the network in each round-trip time (RTT). Since RTT is nearly constant in this case, the sending rate is proportional to w. The spatiotemporal diagram of the w values of the individual sources for the same scenario is shown in Figure 4.3.

One can see that in both cases the pattern remains stable and propagates in anti-clockwise direction. In this respect it resembles the congestion propagation in car traffic. The congestion wave is stable, the speed of the congestion wave pattern is almost constant. It is also apparent that the two waves are synchronised.

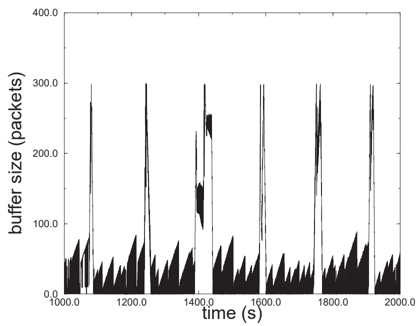

Figure 4.4: Time evolution of the buffer size at one individual link.

Figure 4.4 and Figure 4.5 show the time evolution of the buffer size and the congestion window at one individual node, respectively.

The average speed of the wave can be determined by measuring the average speed of the center of mass of the pattern. This should be carefully defined in the present situation as the system is spatially periodic. Mapping the vertices upon each of the Nth roots of unity in the complex plane weighted by the sending rates gives the complex number indicating the center of mass. The center of mass thus

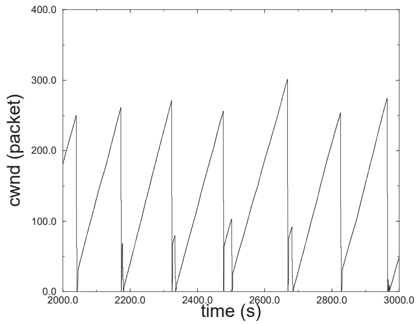

Figure 4.5: Time evolution of the congestion window of one individual TCP connection.

can be obtained by

$$\langle i\rangle(t) = \frac{N}{2\pi}\arg\left(\sum_{j=0}^{N-1} X_j(t)e^{\mathbf{i}(2\pi/N)j}\right),$$

where X_j indicates the sending rate of the jth terminal computer. The speed of the pattern is the time derivative of this quantity.

Once the speed of the pattern is determined the shape of the profile can be analyzed. Since the congestion waves has various shapes in different time instants, it is necessary to take average of the sending rates at the wavefront. Representing the sending rates $X_{i'+[\langle i\rangle]}(t)$ in co-moving coordinates i' relative to the center of mass the shape of the traveling wave pattern is recovered. Averaging the new series in time the profile of the front emerges as in Figure 4.6.

4.3 Analysis

In this section the possible reasons of congestion transition between adjacent sites are investigated. It is shown how packet sending mechanism of TCP contributes

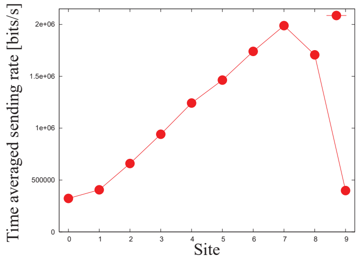

Figure 4.6: The shape of the traveling wave profile. It has been determined by averaging the time series in co-moving spatial coordinates.

to the wave formation. Several traffic properties are pinpointed that are necessary for the development of stable waves.

It is stated that these properties are necessary in the sense that if the system is configured such that one of these properties is not valid, congestion does not propagate (even if it occurs occasionally somewhere).

4.3.1 TCP properties

TCP has many built-in algorithms and methods that enable reliable data delivery and congestion control. Moreover, TCP has many variants that have been developed in the past decade to optimise the data transfer. This makes it difficult to give a general model, however, TCP has some main properties valid for the most frequently used versions. Those properties are highlighted in this section that are common in most cases and contribute to the development of congestion waves between adjacent nodes.

4.3.1.1 Adaptivity

One main property of TCP that plays an important role in the congestion transition is its way to adapt to the network conditions. In this section the traffic pattern of TCP is characterised, based on TCP congestion control described in Section 3.2. In this case long-lasted connections are modelled so it is sufficient to investigate the 'congestion avoidance' phase. In this phase TCP follows the AIMD model that results in a traffic pattern as shown in Figure 4.7.

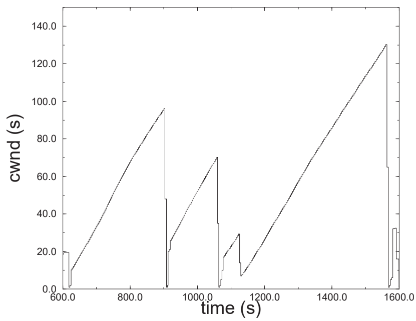

Figure 4.7: Time evolution of TCP congestion window in case of random loss.

Assuming constant RTT during this process, the congestion window and the number of packets out in the network are increased linearly in time. The increase is additive, and it lasts until packets are dropped. Then the congestion window is decreased by halving it. In case of multiple packet loss the congestion window is decreased even further, where it might reach its minimal value.

4.3.1.2 Bursty packet injection

If the waiting time between the packet arrivals is highly variable then a typical traffic pattern has long waiting times without any arrivals and short intervals where many packets arrive. A series of a large number of packets arriving within a short time interval is called burst.

Another property of TCP traffic playing major role in congestion transition is the bursty packet sending from each sources in the network path. Several sources of burstiness can be found in the network, some of them directly relate to the TCP mechanism.

File sending controlled by TCP usually begins with slow start. TCP slow start means that starting with a small window, every acknowledgement generates two other packets. The initial window size is usually 1, but there can also be larger values. With delayed acknowledgement policy, every second packet is acknowledged and every acknowledgement implies three packets, all at once. In case of small file downloads the majority of the file is retrieved in slow start state. This means that a considerable amount of packets are placed in double or triple bursts. In most cases slow start is followed by congestion avoidance phase, where TCP sends a double packet after the whole congestion window is acknowledged, which can cause slightly bursty traffic.

Another source of burstiness can be the fast recovery algorithm of TCP for the detection and correction of packet loss. When a packet is lost and several consecutive packets arrive, the receiver acknowledges only the packet right before the lost packet (duplicate acknowledgements). After the lost packet is resent and successfully received, the receiver may acknowledge several packets at the same time. As a consequence, the number of unacknowledged packets decreases and several packets can be sent into the network in a burst.

There are certain applications that may send larger data packets than the Maximum Segment Size (MSS), e.g. video applications. These packets are fragmented into packets of MSS size on the IP level, resulting in a burst of the same size as the original packet size.

TCP is often in close connection with the application layer (e.g. HTTP) and this can have significant effect on the resulting burst structure. The communication between client and server on the application level begins with a request of a particular file, usually a text file that contains some hyperlinks to other objects and files. This is often achieved by clicking on a given URL. The server sends the file to the client, who is then able to send requests for the embedded files. The request sending policy of the client can be implemented in different ways. Either the client waits for the response before the next request is sent out, or the client sends as many requests as it can allowing more requests to be in the network at the same time without having any response (pipelining). Many servers support the so-called keep-alive connection, where HTTP uses the same TCP connection to get more files. In a persistent connection the congestion window of TCP remains the same when starting a new file download. If pipelining is not set, the initial window size can be as large as 10 or 20 packets, which can cause significant burst effects.

In the network scenario presented here bursts are generated due to the congestion of acknowledgements. If the smaller size acknowledgement packets are queued up one after each other in front of their receiver (that is the same as the sender of the data packets), they are served in relatively small amount of time. In case of long transfer each acknowledgement packet generates another data packet to keep the window open. Due to the short time of the receiving of the the consecutive congested acknowledgements the data packets generated by them arrive in burst.

4.3.2 Wave formation

In this section it is explained how the TCP properties described in 4.3.1 contribute to forming and propagating congestion waves. First a balance equation is derived that expresses the utilisation of each link as the function of the sending rates. Then some properties of the TCP traffic are pinpointed as the major contributors to the wave propagation.

While the continuous equations constitute gross simplification of the original TCP dynamics, the main properties of the traveling wave can be recovered from them with some additional assumption made on the packet loss process as it is shown next.

The utilised bandwidth $C_{i-1}(t)$ on the link connecting nodes $i-1$ and i is the sum of sending rates of TCPs whose traffic flows through that link. In this case the flows of all TCPs traverse that link except the one starting at node i and ending at node $i-1$:

$$C_{i-1}(t) = \sum_{j=0, j\neq i}^{N-1} X_j(t) = \sum_{j=0}^{N-1} X_j(t) - X_i(t), \tag{4.1}$$

where $0 \leq i \leq N$ and site $i = N$ is identified with site $i = 0$ due to periodicity. The traffic of ACK packets emanating in $i-1$ and absorbed in i is low due to their small size and their contribution to the traffic can be neglected.

Due to the *additive increase algorithm* of TCP the rate is increasing monotonously. Congestion and packet loss occur in the system whenever the utilised bandwidth of one of the sites $C_i(t)$ reaches the link capacity C. According to Equation 4.1 the largest link utilisation $C_i(t)$ is at site $i = i^* - 1$ where i^* is the site where the sending rate $X_{i^*}(t)$ is the lowest.

One then has to investigate which TCP flow will lose packet on link $i^* - 1$. In principle all the TCP flows traversing the congested link can lose packets, so only the TCP flow at site i^* is immune. However, the observation is that the TCP flow starting at the actual congested link (with sending rate X_{i^*-1}) experiences the packet loss almost surely. This is due to the fact that TCP sends data packets in batches as it is described in Section 4.3.1. Then obviously the TCP flow that ejects this burst of data packets directly into an almost saturated buffer will lose packets in the process.

The TCP at site $i^* - 1$ suffers packet losses repeatedly and due to the *multiplicative decrease algorithm* of the TCP protocol its sending rate X_{i^*-1} becomes smaller than X_{i^*} after several packet losses. From then on X_{i^*-1} will be the lowest in the system, link utilisation C_{i^*-2} will be the highest after a while and TCP at site $i^* - 2$ suffers the packet losses. This way congestion propagates site by site anti-clockwise in the system. After several rounds of congestion propagation the propagating front of Figure 4.6 emerges.

The explanation of the congestion propagation is summarised in Figure 4.8.

4.4 Avoiding burst effects

The consequence of the bursty nature of the individual flows is that those TCP flows will lose packets that are closest to the loaded buffer. Two simulation scenarios have been installed where the parameters are set so that this effect is avoided and congestion propagation is investigated in these cases. The first simulation setup applies a special queue management algorithm in the routers, while the second one uses a TCP parameter setting to limit the sending rate.

In the first setup a more complicated packet drop scheme is used in the router, called Random Early Detection (RED) introduced in [26]. In spite of the tail-drop algorithm used so far, in this algorithm not only those packets are dropped

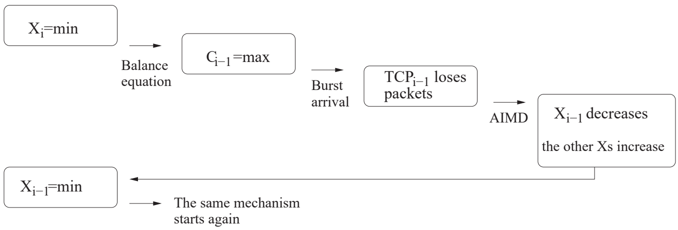

Figure 4.8: Basic mechanism of congestion propagation.

that arrive at full buffer but some randomly chosen packets in case the averaged length of the queue in the buffer reaches a threshold. This way the routers start dropping packets earlier than congestion occurs, thus smoothing out the packet drop process.

After a packet arrival, the average queue size is calculated using an exponential moving average. This calculated average queue size is compared to two thresholds and based on the result a decision is made if the packet is dropped or not. There is a minimum and a maximum threshold (*minthresh*, *maxthresh*). Below *minthresh*, no packets are dropped. Between *minthresh* and *maxthresh* each packet is dropped with probability p where p is a function of the average queue length. If the average queue length exceeds *maxthresh*, all packets are dropped. Figure 4.9 shows the packet drop probability against the calculated average queue length when using RED.

As it is discussed in Section 3.2, there are built-in mechanisms in TCP to provide congestion control at the end points of the network. The main purpose of implementing RED was to recognise and control congestion in the routers as well. Another advantage of RED is that it helps avoiding burst-losses where consecutive packets tend to be dropped, causing global synchronisation of TCP flows and large performance degradation. Evaluations of RED and proposals to improve the

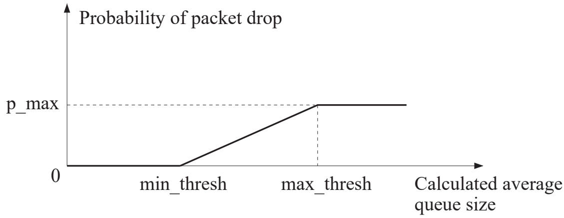

Figure 4.9: Packet drop probability using RED algorithm.

algorithm can be found e.g. in [27], [28].

In the present scenario when RED is applied in a router, not only the computer near the congested router suffers packet loss, which implies that in some randomly chosen cases, other TCP sources will lose packet and decrease their sending rates. This way the basic mechanism illustrated in Figure 4.8 does not work since other computers might have the minimal sending rate and congestion occurs at another router. Moreover, the larger the number of routers RED is applied in, the more computers decrease their rates in advance, resulting that congestion might completely disappear.

Figure 4.10 shows the spatiotemporal diagram of the buffer usage when RED is applied in 1 - 4 routers. In the case of 1 router with RED congestion waves occur but they are not stable. In the case of 2 and 3 routers with RED some congestion can be observed but it does not propagate. When 4 or more routers apply the RED algorithm congestion completely disappears.

Another method to avoid the effect of burstiness is to limit the congestion window. In real systems there is a parameter to maximise the congestion window negotiated between the sender and the receiver to limit the sending rate so as the receiver is not overloaded. This variable is referred to as *Advertised window* in Section 3.2. Indeed, if the limitation is not based on the response to congestion but

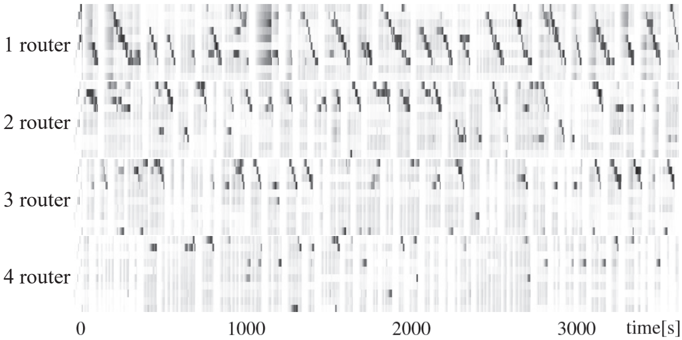

Figure 4.10: Spatiotemporal diagram of congestion propagation. The RED algorithm is applied in 1 - 4 routers. The values of *minthresh* and *maxthresh* were set to 50 and 100, respectively.

it is determined by a built-in constant variable then congestion can be completely eliminated.

Figure 4.11 shows the spatiotemporal diagram when the congestion window was maximised at 100 packets at several TCP flows. The number of limited flows ranges from 1 to 6. It can be observed that the congestion wave tends to be more and more distorted as the number of limited TCP flows increases. In case of 6 limited TCP flows the buffer usage is always small in those routers where the limited flows are connected. In those cases without limitations the system gets congested from time to time and the buffer usage oscillates individually however, wave propagation does not occur.

4.5 Conclusions

In this chapter the forming and propagation of congestion in a simple network scenario is investigated. The phenomenon is analyzed in detail and it is derived that the intrinsic properties of the TCP protocol contribute to the formation and the stability of the congestion waves. The large rate variation of TCP sending (burst

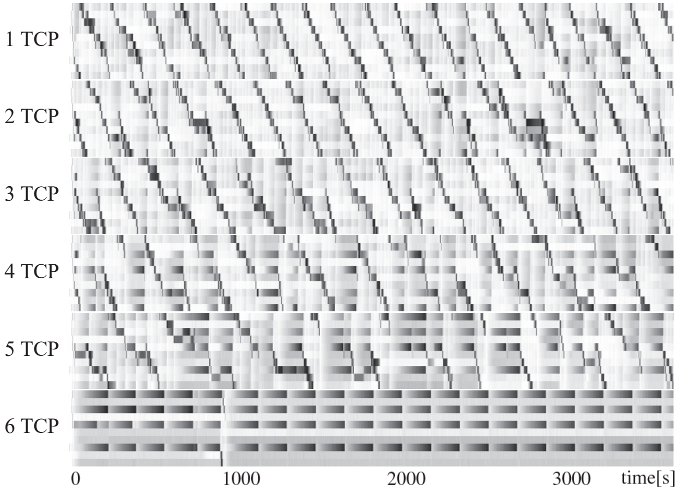

Figure 4.11: Spatiotemporal diagram of congestion propagation. The number of limited TCPs varies from 1 to 6.

effect) is pinpointed as one of the major contributors of this phenomenon. These statements are supported by simulation experiments where the different settings of network parameters and algorithms provide different conditions for congestion propagation.

The microscopic model presented here emphasises some key effects experienced in the current TCP/IP networks such as bursty packet traffic and congestion and sets up a relationship with those effects and network parameters. The model gives a deeper insight into the basic mechanisms of congestion formation and burstiness and the simulation study shows some examples on how the burst effects can be avoided.

Chapter 5

Modeling short TCP connections

Previous TCP models mostly considered infinite data sources, where stationarity of TCP is assumed [14], [15], [20], [21]. In [22] and [23] short data transfers are investigated but the number of parallel connections is limited there. In the first model presented here the TCP connections are in transient phase, moreover, the population of TCP sources is unlimited, which makes it possible to formulate the model in compact way by using a few basic traditional traffic parameters only.

The objective is the description of multiple connections sharing a single link, where the flows are typically short and the traffic rate is decomposed according to the number of parallel TCP flows in the system. Another purpose is to calculate the traffic rate where the files are downloaded sequentially, using different traffic control algorithms.

5.1 Modeling parallel TCP connections

In this section the number of parallel TCP connections sharing a single link is investigated. First the system setup and the main assumptions are shown, then the average utilisation as the function of the number of parallel connections is computed, that is followed by setting up a Markovian model to describe the dynamics of the number of connections.

5.1.1 System setup

The outline of the system model is shown in Figure 5.1. TCP connections arrive randomly from an infinite population according to a Poisson process. Each connection initiates a file transfer. The model focuses on short files where the tail of the file size distribution is short (exponential decay). Low packet loss is assumed so that the transfers seldom leave the initial slow start phase.

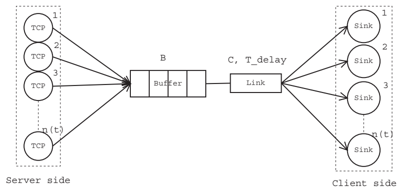

Figure 5.1: System topology. In the simulations zero packet loss and fixed delay T_d varying between $5ms$ and $160ms$ has been assumed. The buffer size is considered as infinite and the bandwidth is $C = 10^7$ bps.

Generally, the teletraffic systems can not be characterised by Poisson arrivals and exponential file sizes. The reason of the choice of these simple models instead is that:

- The file requests arrive from a large population of users, often resulting in Poisson statistics. Packet arrival statistics within connections are different from Poisson.
- The model concentrates on WWW browsing where small files dominate. The transmission of large files can be treated separately, based on the well-known persistent TCP models.

5.1.2 Description of aggregated traffic

A computation is shown that can be used to obtain the distribution of the congestion window (*cwnd*) sizes when multiple different connections are present. The calculation leads to a formula describing the utilisation of the link. The utilisation can be considered as the probability that at a randomly chosen time the buffer is serving. The computation method is based upon the independence of the parallel TCP connections. The packet loss is neglected and the distribution of file sizes is exponential. It is assumed that the sources send a certain amount of data (based on *cwnd*) in every round. The value of the round-trip time is not needed in the calculations. Although TCP measures and updates its *cwnd* in terms of bytes it is easier to count the number of packets out in the network unacknowledged by the receiver. The probability that a file consists of N_p packets can be written as

$$\rho_p(N_p) = P((N_p-1)S_p < S < N_pS_p) = \int\limits_{(N_p-1)S_p}^{N_pS_p} \rho(S)dS = e^{-\sigma(N_p-1)S_p} - e^{-\sigma N_p S_p}, \tag{5.1}$$

where S_p is the size of the IP packets, N_p is the length of file measured in packets and $\rho(S)$ is the probability density function of the file sizes ($\sigma > 0$):

$$\rho(S) = \begin{cases} \sigma e^{-\sigma S}, & \text{if } S > 0 \\ 0, & \text{if } S \le 0. \end{cases}$$

A file consisting of S bytes can be divided into $N_p = [S/S_p] + 1$ packets where $[\cdot]$ denotes the lower integer part. The file sizes have been modelled as real numbers so far, however the number of packets in a file is always an integer number. The discretisation of the exponential distribution gives geometric distribution with parameter $p = e^{-\sigma S_p}$.

$$P(N_p - 1 = k) = p^k(1-p) \tag{5.2}$$

that is the random variable $[S/S_p]$ follows geometric distribution with parameter p. If the number of round-trips needed for the file to be downloaded is N_r, the number of packets that have been sent out in the last window is denoted by $\hat{S}$ and assuming that the TCP is in slow start phase in the whole download period, one can write

$$N_p = \sum_{i=0}^{N_r-2} 2^i + \hat{S}.$$

Given the file size N_p, the N_r and $\hat{S}$ can be calculated by

$$\begin{aligned} N_r(N_p) &= [\log_2(N_p)] + 1 \\ \hat{S}(N_p) &= N_p - 2^{[\log_2(N_p)]} + 1, \end{aligned}$$

where $N_p \geq 1$.

Let $\rho(N_p, w)$ denote the probability that a TCP transferring a file of length N_p keeps w packets in the network at an arbitrary moment. Given $N_r(N_p)$ states of the system and assuming that the probability of all states is equal $(1/N_r(N_p))$ the following equation holds:

$$\rho(N_p, w) = \frac{1}{N_r(N_p)} \sum_{i=0}^{N_r(N_p)-2} \delta(w - 2^i) + \frac{1}{N_r(N_p)} \delta(w - \hat{S}(N_p)) \tag{5.3}$$

where $\delta(.)$ is the delta function giving

$$\delta(x) = \begin{cases} 1, & \text{if } x = 0 \\ 0, & \text{otherwise.} \end{cases}$$

Using the discrete probability distributions (5.1) and (5.3) the distribution of the congestion window size when one TCP connection is present in the system can be computed as the following:

$$\rho_{TCP}^{(1)}(w) = \sum_{N_p=1}^{\infty} \rho(N_p, w)\rho_p(N_p) =$$

$$= \sum_{N_p=1}^{\infty} \frac{e^{-\sigma(N_p-1)S_p} - e^{-\sigma N_p S_p}}{[\log_2(N_p)] + 1} \left[\delta(w - \hat{S}(N_p)) + \sum_{i=0}^{[\log_2(N_p)]-1} \delta(w - 2^i) \right]. \quad (5.4)$$

If more than one TCPs are allowed to run in the system then the sum of the *cwnd* sizes is the quantity characterising the network load. Convolving two distribution functions like in Equation (5.4) one can get the result.

$$\rho^{(2)}_{TCP}(w) = \sum_{k=1}^{w} \rho^{(1)}_{TCP}(k) \rho^{(1)}_{TCP}(w-k),$$

Following this method, the distribution of the sum of *cwnd*s in case of n connections can also be achieved by taking the convolution of the distribution functions in case of 1 TCP and $n-1$ TCPs.

$$\rho^{(n)}_{TCP}(w) = \sum_{k=1}^{w} \rho^{(1)}_{TCP}(k) \rho^{(n-1)}_{TCP}(w-k) \quad (5.5)$$

Knowing these results the probability of queuing can be obtained. The sum of the *cwnd*s is the number of all segments that have been already sent out by the TCP sources but have not yet been acknowledged. Since in this case there is only one buffer in the path all unacknowledged packets are near that link, either being served or waiting. The maximal number of packets on the link is determined by the product of the bandwidth and the delay of the link (often referred to as 'pipe size'). This is the number of segments that can be transmitted over the link without suffering any queuing delay. If the sum of the *cwnd* sizes (w) is larger than the bandwidth-delay product measured in packets (CT_d/S_p) then buffering will certainly occur. If $w < CT_d/S_p$ then the probability of buffering is the ratio of the sum of *cwnd*s to the 'pipe size':

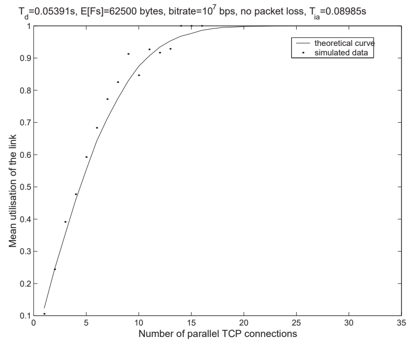

Figure 5.2: The utilisation of the buffer. The data points are the simulation results and the solid line represents the numerical evaluation of the analytical model.

$$p_q(w) = \begin{cases} \frac{wS_p}{CT_d}, & \text{if} \quad \frac{CT_d}{S_p} > w \\ 1, & \text{otherwise.} \end{cases}$$

Using the distribution of the congestion window sizes (5.5) the link utilisation as the function of the number of parallel TCPs can be obtained.

$$r_n = \sum_{w=1}^{\infty} \rho_{TCP}^{(n)}(w) p_q(w). \tag{5.6}$$

Some useful descriptors of the network have been obtained in a theoretical way. The Formula 5.6 is evaluated numerically by iterating Equation 5.5 using the first step of the iteration (Equation 5.4). Figure 5.2 shows the solution compared to the simulation results. After a linear increase the function r_n apparently goes to 1 as n grows. The next step is to model the number of parallel flows and connect it to Equation 5.6.

5.1.3 Markovian model of the number of flows

A simple Markov model is introduced to describe the system. The states of the Markov chain are the number of parallel TCP connections n.

It is assumed, that the length of the buffer is infinite, no packet loss occur and the delay is fixed. Since connection departure can occur only in packet departure instants, the previously developed results can be applied to obtain the distribution of the number of parallel TCP connections.

Using the exponential file size, let μ denote the rate of connection departure, given that the server is fully utilised. Then

$$\mu = \frac{C}{E(S)} = \frac{C}{S_p\, E(N_p)}.$$

If there are i connections in the system and the utilisation is r_i then the rate of departure is $r_i\mu$. The connections arrive randomly according to a Poisson process with rate λ. The state-diagram of the Markov chain is depicted in Figure 5.3.

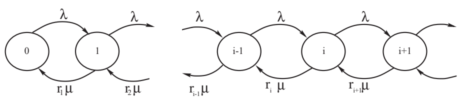

Figure 5.3: State diagram of the Markov chain describing the system model.

Figure 5.4 shows the utilisation computed in Section 5.1.2. The function r_n is partitioned to a linear part and a constant part (that is equal to 1). The linear part corresponds to the case when the bandwidth-delay product is large enough, queuing does not occur and the TCPs are independent from each other. The constant part means that the pipe is full, TCP packets have to wait in the queue and the connections have to share the available bandwidth. A threshold value is introduced to separate the two scenarios.

Using the linear approximation $r_n = \frac{n}{k^*}$ this threshold will be at k^* as it is shown in Figure 5.4. The value of k^* can be calculated as the inverse slope of this linear approximation. The simplified function of r_n can then be written as

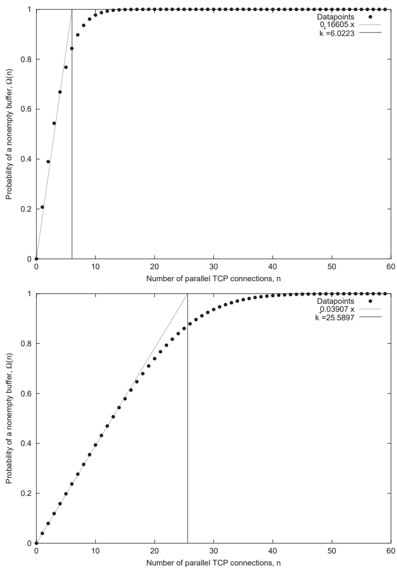

Figure 5.4: The utilisation of the buffer. The mean value of the file size is set to 62500 bytes and the packet size is 512 bytes. Two cases are presented, T_d =26.95ms, 161.72ms.

$$r_n = \begin{cases} \frac{n}{k^*}, & \text{if } n \leqslant [k^*] \\ 1, & \text{if } n \geqslant [k^*]. \end{cases}$$

Using the approximated values of r_n the simplified Markov-chain is shown in Figure 5.5. Using standard techniques [95] the steady-state probabilities of the simplified Markov-chain can be evaluated. The distribution is Poissonian below k^* with parameter $\varrho = \frac{\lambda}{\mu/k^*}$ and geometric with parameter $\varrho^* = \lambda/\mu$ above k^*:

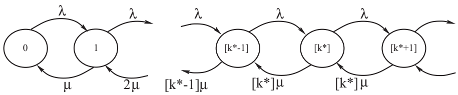

Figure 5.5: Simplified state diagram of the Markov chain describing the system model.

$$p_n^* = \begin{cases} p_0^* \frac{\varrho^n}{n!}, & \text{if } n \leqslant [k^*] \\ p_0^* \frac{\varrho^{[k^*]}}{[k^*]!} (\varrho^*)^{n-[k^*]}, & \text{if } n \geqslant [k^*]. \end{cases} \tag{5.7}$$

The value of p_0^* can be determined by normalisation and is given by

$$\frac{1}{p_0^*} = \sum_{j=0}^{[k^*]} \frac{\varrho^j}{j!} + \frac{\varrho^{[k^*]}}{[k^*]!(1-\varrho^*)} \tag{5.8}$$

The above formulae have been derived also in [19] for M/G/1 processor sharing model. An important consequence of this relation is that the parameter of the geometric distribution can be expressed with the parameter of Poisson distribution and the threshold. This relation enables one to interpret Equation 5.7 as a generalisation of Erlang's formula [5] for TCP traffic. The parameter ϱ can be calculated from the classic traffic parameters λ and μ – as in Erlang's formula –, and recall that the relation between ϱ and ϱ^* is $\varrho^* = \varrho/k^*$.

Once the number of connections is modelled, traffic descriptors such as download time can be calculated from Little's law.

5.1.4 Validation of the model

Simulations has been performed in order to validate the Markov model. The `ns-2b` simulator [90] with TCP Reno version was used. In the simulations the random packet loss has been neglected and the buffer size was set to an extremely large value. The link speed was set to $C = 10^7$ bps. The average file size was $1/\sigma = 62500$ Bytes that is about 122 IP packets. Simulations of 15000 file downloads have been made at different link delays. In the simulations $\varrho^* = 0.6$ and $\lambda = 11.14$ $1/s$ were fixed.

Then the parameter values ϱ and $[k^*]$ have been estimated from the observed histograms. The histograms and the model distribution with the estimated parameters are depicted in Figure 5.6. Both linear and logarithmic scales are presented. It can be seen that the model distribution follows the histogram and the error remains bounded both in the main part and in the tail. The estimation of the model parameters was done by the weighted least squares method.

T_d[ms]	ϱ	ϱ^*	ϱ/ϱ^*	k^*
5.39	1.67	0.6	2.78	2.87
26.95	3.43	0.6	5.72	6.02
53.91	5.67	0.6	9.45	9.26
107.81	9.83	0.6	16.38	17.38
161.72	14.10	0.6	23.50	25.59

Table 5.1: Parameter values for different simulation scenarios.

In Table 5.1 the fitted values of ϱ and ϱ/ϱ^* are shown. The variance of ϱ decreased with the increasing number of samples. It was found that for 15000 downloads the relative error of ϱ was around 1.5%. In the last column the computed k^* values are shown.

Comparing the fitted ϱ/ϱ^* and the calculated k^* values one can see that the anticipated relation $k^* = \varrho/\varrho^*$ holds over the whole parameter range with some minor deviations. Moreover, the k^* values calculated analytically from the model (last column) is close to the data fitted to the simulations.

5.2 Modeling sequential TCP connections

In this section short TCP connections following each other sequentially are analysed, which is typical in Web browsing. An analytical model is introduced to compute the speed of the Web page download in case of different mechanisms of handling the consecutive TCPs.

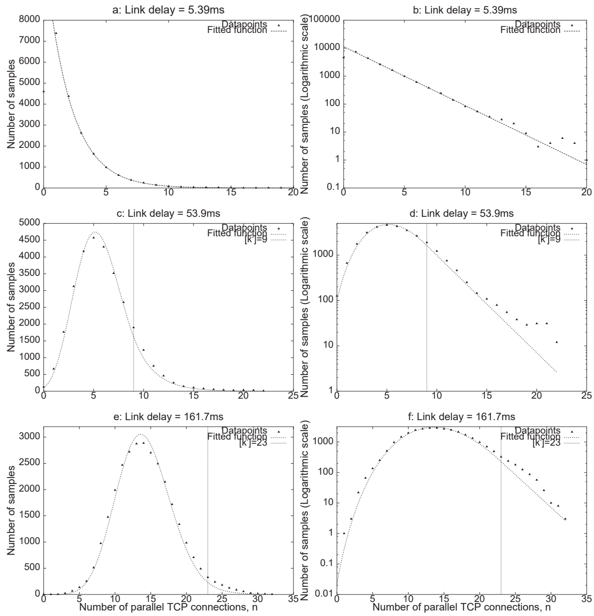

Figure 5.6: The distribution of the number of TCP connections. Three typical cases are presented on linear and logarithmic scale.

5.2.1 Packet transmission in WWW applications

Although various network applications can be found in the Internet, Web applications have significant share in the total traffic volume.

The Hypertext Transfer Protocol (HTTP) is responsible for sending and receiving the contents of Web pages. HTTP is an application layer protocol that uses TCP on transport layer to control the transmission.

The communication between client and server on the application level begins with a request of a particular file, usually a text file that contains some hyperlinks to other objects and files. This is often achieved by clicking on a given URL. The server sends the file to the client, who is then able to send requests automatically for the embedded files. The request sending policy of the client can be implemented in different ways. Either the client waits for the response before the next request is sent out, or the client sends as many requests as it can allowing more requests to be in the network at the same time without having any response. The later version is called *pipelining* and it is an important functionality of HTTP. It can improve the performance, since all files belonging to a Web page is downloaded without waiting for any file transfer to be finished.

5.2.2 Analytical model

Different HTTP models and different file size distributions are considered. The effect of TCP is taken into account by the results presented in [22]. This model needs only the round-trip time RTT, the file size in packets, the packet loss probability and some TCP-related parameters as input and it generates the download time as output.

In case of HTTP with pipelining, the download time and the average congestion window ($cwnd$) can be calculated by considering the transfer of the Web page as a continuous data flow[1], while in HTTP without pipelining the download is sometimes interrupted at end-of-file events. This behaviour results in additional RTT in the download time and the average $cwnd$ (W_{npl}) is therefore smaller than that of HTTP with pipelining (W_{pl}). Analytical formulae can be developed in both cases.

The model is based on the formula for expected time of data transfer given in [22]. With that formula the download time of a given file with known size can be computed. In the following, the distribution of the file size and the number of objects is assumed to be known. Considering a given file size distribution, the average download time can be calculated by taking the probabilities of different file size occurrences. Let the file size be a continuous variable and $f(x)$ be the probability density function of the file size distribution. In case of more than one embedded object, the page size (PS) is the sum of the embedded object sizes.

[1] In practice, first the client should wait for the arrival of the base-page and then follows the sending of the requests for the embedded objects that the base-page contains references for.

The probability density function of the sum of i variables from the distribution characterized by f can be written as the ith convolution of f. If we define $g_1(x) = f(x)$ the following recursive formula can be written:

$$g_i(x) = \sum_{y=1}^{x} f(y) g_{i-1}(x+1-y)$$

This is the probability density that one page contains x packets in the case of i embedded objects. If h denotes the distribution of the number of embedded objects on one page, the probability that one page contains x packets can be described by

$$g(x) = \sum_{i=1}^{\infty} g_i(x) h(i).$$

The expected download time of one page is then

$$E(T_{dl}) = \sum_{x=1}^{\infty} T_{dl}(x) g(x),$$

where $T_{dl}(x)$ is the expected time for data transfer according to [22]. The expected page size $E(PS)$ can be calculated by taking the average over the $g(x)$ distribution:

$$E(PS) = \sum_{x=1}^{\infty} x g(x)$$

The approximate calculation of the throughput is given by dividing the page size with the download time for one page and the average $cwnd$ is the throughput multiplied by RTT:

$$W_{pl} = \frac{E(PS)}{E(T_{dl})} RTT \tag{5.9}$$

The average $cwnd$ may depend on the file size distribution. The effect of the tail of distribution is investigated by numerical computations. Table 5.2 shows that comparing typically short-tailed (exponential) and heavy-tailed (Pareto) file sizes, the final result does not differ much, the deviation remains below 4 %.

	W_{pl}, Exp	W_{pl}, Par	Dev [%]
$F = 10, p = 0.01$	4.257206	4.164887	2.22
$F = 10, p = 0.05$	2.284299	2.243360	1.82
$F = 10, p = 0.10$	0.918700	0.909426	1.02
$F = 50, p = 0.01$	17.18964	16.53288	3.97
$F = 50, p = 0.05$	9.074864	8.733306	3.91
$F = 50, p = 0.10$	2.822689	2.758703	2.32

Table 5.2: Deviation in average $cwnd$ between exponentially distributed and Pareto distributed file size, for different combinations of average file size (F) and packet loss (p).

The formula for W_{pl} is checked for exponentially distributed file size and verified by simulations presented in Section 5.2.3.

In HTTP without pipelining the average congestion window is, as mentioned earlier, smaller than the average congestion window of HTTP with pipelining, for which the calculations shown above are appropriate for attaining W_{pl}. This result can be used for deriving W_{npl}, taking into consideration that sometimes the last segments of a particular file do not fill the available space allowed by the congestion window. Since the files are transmitted one by one, the relation between W_{pl} and W_{npl} can be written as

$$W_{npl} = \frac{F}{\hat{R}(W_{pl})} \tag{5.10}$$

where F is the average file size and $\hat{R}(W_{pl})$ is the average number of round-trips needed to download one file. On average, in case of HTTP with pipelining, the system works as if W_{pl} packets were transmitted in every round-trip. All files are finite, so in the last round-trip the sender does not always send W_{pl} packets. Consequently, the transfer of a file of size S is expected to be completed in $\lceil \frac{S}{W_{pl}} \rceil$ round-trips, where $\lceil x \rceil$ means the ceiling function, i.e. the smallest integer larger than or equal to x. Taking the average over the file size distribution, the expression for $\hat{R}(W_{pl})$ can be written as

$$\hat{R}(W_{pl}) = \int_{0}^{\infty} \left\lceil \frac{S}{W_{pl}} \right\rceil f(S)dS \tag{5.11}$$

When considering exponentially distributed file sizes with average F, this expression can be simplified by evaluating the integral as follows:

$$\hat{R}(W_{pl}) = \sum_{i=0}^{\infty} \int_{iW_{pl}}^{(i+1)W_{pl}} \left\lceil \frac{S}{W_{pl}} \right\rceil f(S)dS = \sum_{i=0}^{\infty} e^{-\frac{iW_{pl}}{F}} = \frac{1}{1 - e^{-\frac{W_{pl}}{F}}} \tag{5.12}$$

This formula is appropriate only for exponential distribution. With heavy-tailed distributed file size (e.g. Pareto) an explicit formula is not this easily derived. However, numerical evaluations of Equation 5.11 show that the results when using Pareto distributed file size do not differ much from the case when using exponentially distributed file size. In Table 5.3 the results are shown where the difference in $\hat{R}(W_{pl})$ between Pareto and exponential distribution is investigated by varying W_{pl} and the average file size.

	$\hat{R}(W_{pl})$, Exp	$\hat{R}(W_{pl})$, Par	Dev [%]
$F = 10, W_{pl} = 1$	10.508	10.513	0.05
$F = 10, W_{pl} = 4$	3.0332	3.0717	1.25
$F = 10, W_{pl} = 8$	1.8160	1.8821	3.51
$F = 50, W_{pl} = 1$	50.502	50.408	0.19
$F = 50, W_{pl} = 6$	8.8433	8.8519	0.10
$F = 50, W_{pl} = 12$	4.6866	4.7099	0.50

Table 5.3: Deviation in download time between exponentially distributed and Pareto distributed file size, for different combinations of average file size (F) and average $cwnd$.

Since the mean values of the file sizes are the same, the influence of the ceiling function in Equation 5.11 is large when the value of F/W_{pl} is small, but in all cases it remains within 4 %.

Finally, combining Equations 5.10 and 5.12 gives us the relation between the average congestion window sizes using HTTP with and without pipelining:

$$W_{npl} = F(1 - e^{-\frac{W_{pl}}{F}}) \tag{5.13}$$

The simulation results also confirm that the distribution of file size does not have much influence on the performance therefore, the computations and simulations are based on the exponential case.

5.2.3 Validation of the model

The main results of the analytical model stated in Section 5.2.2 are compared to simulations and measurements of the corresponding scenarios.

The calculations are carried out for different parameters of the Web site (average file size and average number of embedded objects on a Web page) and different values of packet loss in the network (from 0.01 to 0.1). Constant round-trip time is assumed in all cases and the pipelined and non-pipelined versions are both considered. The main metric of interest is the average congestion window size, from which the average download time and the average offered load can be calculated. For testing the method, a simple network of a Web client, a Web server and a link connecting them is used.

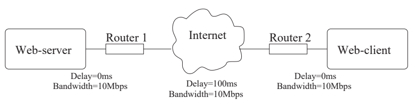

Figure 5.7: System topology. The delay and the bandwidth are fixed, the packet loss varies between 0.01 and 0.1.

Figure 5.7 shows the investigated scenario. The fix delay in the core network is 0.1s and the bandwidth values of the links are 10Mbps, which corresponds to a high bandwidth-delay product network. The upload delay is 0s, so acknowledgements and requests from the client side can reach their destination immediately.

Since the TCP segment size is 1000 bytes (plus 40 bytes header size), the RTT (which includes the fix delay and the packet service time) has a constant value of 0.1025s.

The simulations were performed by `ns-2.1b6` simulator tool [90]. This version of `ns` does not contain any HTTP-related objects, so the application part of the simulator needed to be implemented. An `ns`-based HTTP-simulator written in Tcl found at [25] served as the underlying tool of the simulations, where HTTP with and without pipelining is implemented.

Figure 5.8 shows the simulation and computation results in different scenarios. The average $cwnd$ of TCP is presented as the function of packet loss probability.

In the simulations packets were dropped randomly at a given rate. The distribution of the file size and the number of objects inside a Web page were both exponential and the simulations stopped when 2000 pages were retrieved. In order to view the difference between the pipelined and non-pipelined connections, the two cases with the same parameter settings are plotted on the same figure. Six cases are distinguished depending on the average file size ($F = 10kB$, $50kB$), average number of embedded objects ($N = 5$, 10), and the maximum $cwnd$ of TCP advertised by the client ($W_{max} = 10$, 50). The usage of Nagle algorithm [2] was switched off, no delayed acknowledgement was set and the initial $cwnd$ was 1.

Several conclusions can be drawn looking at the plots more closely. The analytical model fits well to the simulations up to 5% packet loss, but for larger packet losses they become more separated. The pipelined HTTP and the corresponding data generated by HTTP without pipelining converge, both in simulation and computation cases. This means that if the packet loss rate is high, then the effect of pipelining is small. A similar statement can be declared concerning the file size. If the average file length is large, the influence of the pipelining is small. A larger maximal congestion window results, of course, in larger throughput, and this effect is more relevant at smaller packet losses.

In order to validate the model with passive measurements, the same setup was

[2] The Nagle algorithm is an optional method in TCP to collect those consecutive packets smaller than the maximum segment size and concatenate them into one segment in order to decrease the overhead. In case of file transfers typically the last packet is delayed due to this method.

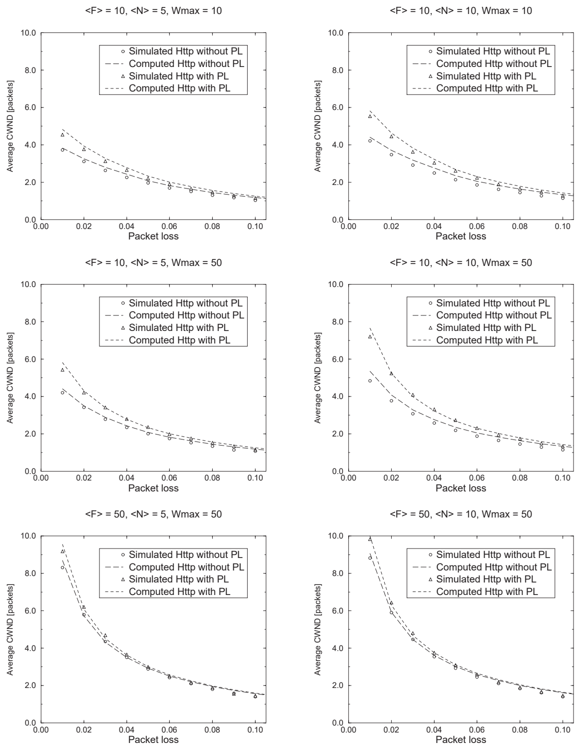

Figure 5.8: Average congestion window, simulation vs computation results.

used except for several parameters. There was a narrow link at the client side, a serial line with 115.2 Kbps. The packet size was 1500 bytes and an extra 0.5s delay was set in the router at the client side in order to decrease the variation of the RTT. The delay was included by the help of `NIST-Net` network emulator tool [91]. Several Web pages were downloaded from a public Web server and the packets at the server side were traced by the `tcpdump` packet capture tool. The measured packet loss was 0.5 %, the average file size was 16.5 Kbytes, the average number of embedded objects was 19, the maximum $cwnd$ advertised by the client was 32120 bytes, which corresponds to 21 packets and the measured RTT was 565 ms. The pipelining was switched on and off in the browser according to the investigated scenario.

The $cwnd$ of the TCP at the server side was calculated by counting the packets sent to the network between the departure of a particular packet and the arrival of the corresponding acknowledgement. Table 5.4 shows the measured and computed $cwnd$.

	Measurement		Computation	
	Pipelined	Non-pipelined	Pipelined	Non-pipelined
$cwnd$	13.38	6.44	11.73	6.56

Table 5.4: Comparison of measured and computed data.

For testing pipelining, `opera6.03` for linux was used, where the maximum number of parallel TCP sessions can be set to 1, but the usage of pipelining can not be disabled. For non-pipelined requests, `mozilla5.0` was applied, where using and not using pipelining can be chosen, but the maximal number of connections can not be set and at least two TCP sessions are running most of the time. The computed values should be estimated with the assumption that in case of two parallel connections one TCP retrieves half of the page with the same file-size distribution.

5.3 Conclusions

In this chapter file transfers are considered where the files are small, the packet loss probability is low and the RTT is constant. In the first case parallel file transfers sharing a single link are investigated and the utilisation of the link is computed as the function of the number of parallel files. Packet level dynamics of TCP is used in the calculations. The values of the link utilisation are then used in the Markovian model of the number of parallel TCP flows on the link. A simplified version of the Markov model is solved and verified. From the number of parallel connections the statistical properties of useful traffic descriptors such as download time and throughput can be calculated.

In the second case the download performance of small files contained on a Web page is analyzed. The difference between the average throughputs and latencies are calculated in case when the files are concatenated into one object (pipelining) and in case when regular file downloads follow each other. From the computational results it can be deduced that the difference between the two cases is larger if the packet loss is smaller and if the average file size is smaller. The throughput calculation is robust in the sense it is insensitive to the distribution of the file size.

Part II

Matrix analytic methods

Chapter 6

Transient behaviour of infinite-server queuing systems

In [71] some explicit formulae for the queuing system with phase-type (PH) arrivals, infinite-server queues and general service-time distributions ($PH/G/\infty$ queue) are derived. A basic system of differential equations is obtained for the queue-length moment generating function. Although the equation system can be solved explicitly only in special cases, certain statements on the moments of the queue length can be made. Additionally, the same statements concerning the basic equation system and the generation of the factorial moments are valid when MAP is applied instead of PH arrival process, resulting in a $MAP/G/\infty$ system. The exact solution of this queuing model is not presented in [71], only numerical solution of the generated set of differential equations is obtained. However, if the service time distribution is restricted to PH, an exact solution can be obtained for the moments of the number of sessions. In [73] the time-dependent generalisation of the queuing system $MAP_t/PH_t/\infty$ is considered and numerical evaluation of the basic system of differential equation is presented. This chapter provides the exact time-dependent solution for the moments of the number of elements being served in $MAP/PH/\infty$ queuing system.

6.1 Moments of an infinite-server queuing system

In the following an infinite-server queuing system is introduced with MAP arrivals and PH service time distribution. The moments of the queue length are computed, where the queue length stands for the parallel demands being served in the system.

6.1.1 Equations for the moments

Let $X(t)$ denote the queue length and $J(t)$ the phase of the arrival process at time t and let $\boldsymbol{\mu}^{(K)}(t)$ denote the M-vector whose ith element is $\mu_i^{(K)}(t)$ $(K \geq 1)$, where

$$\mu_i^{(K)}(t) = E[X^{(K)}(t)|X(0) = 0, J(0) = i]$$

$$K \geq 1, 1 \leq i \leq M$$

$X^{(K)}(t)$ denotes the factorial product $X(t)[X(t)-1]\cdots[X(t)-K+1]$ and $\mu_i^{(K)}(t)$ denotes the Kth factorial moments of the number of demands being served if the system is started from state i with zero queue length.

The main purpose is to calculate the time evolution of the moments of the above queuing system. According to the calculations in [71], the following system of differential equations can be written for each factorial moments of the queue length:

$$\begin{aligned} \frac{d}{dt}\boldsymbol{\mu}^{(1)}(t) &= \mathbf{D}\boldsymbol{\mu}^{(1)}(t) + \{1 - H(t)\}\mathbf{D_1 e} \\ \boldsymbol{\mu}^{(1)}(0) &= \mathbf{0} \end{aligned} \tag{6.1}$$

and for $K \geq 2$

$$\begin{aligned} \frac{d}{dt}\boldsymbol{\mu}^{(K)}(t) &= \mathbf{D}\boldsymbol{\mu}^{(K)}(t) + K\{1 - H(t)\}\mathbf{D_1}\boldsymbol{\mu}^{(K-1)}(t) \\ \boldsymbol{\mu}^{(K)}(0) &= \mathbf{0}. \end{aligned} \tag{6.2}$$

$H(t)$ denotes the cumulative distribution function (c.d.f.) of the service time and $\mathbf{e}$ is the M-vector whose each elements are 1.

According to [71] the following theorem holds for the asymptotic nature of the factorial moments of the queue length:

Theorem 6.1.1. *If the mean service time is assumed to be finite ($\mu = \int_0^\infty \{1 - H(t)\}dt < \infty$), then for any $K \geq 1$, the vector of the factorial moments $\boldsymbol{\mu}^{(K)}(t)$ converges as $t \to \infty$ to a finite vector all of whose components are equal.*

Proof. The proof can be found in [71] in detail. □

The authors in [71] use numerical approach to solve Equations 6.1 and 6.2. However, if the service time distribution is restricted to be PH distribution, the exact solution can be found.

Starting from solving Equation 6.1, the time dependence of each moment of the queue length can be iteratively calculated by Equation 6.2 by using the preceding moment. The solution method is shown next.

6.1.2 Solution for the first moment

For the solution of the transient behaviour, the method introduced here is based on solving the above set of differential equations with service time distributions generated by mixing Erlang type distributions.

The sum of L independent, identically distributed random variables from exponential distribution with parameter δ gives the Erlang distribution with parameters L and δ, and it is denoted by $E(L, \delta)$. The c.d.f. of the Erlang distribution with parameters L and δ is

$$H_e(t) = 1 - \sum_{i=0}^{L-1} \frac{e^{-\delta t} \delta^i t^i}{i!}.$$

The general formula for the c.d.f. of the service time distribution consisting of mixed Erlang distributions is then

$$H(t) = 1 - \sum_{i=1}^{I} e^{-\beta_i t} \sum_{j=0}^{J_i} \alpha_{ij} t^j, \tag{6.3}$$

where β_is and α_{ij}s are coefficients, I is the number of different exponents and J_i is the maximal t-power belonging to the ith exponent. Since $H(t)$ is a c.d.f., $\beta_i > 0$ for all $1 \leq i \leq I$. Formula 6.3 is one representation of a PH distribution function (for details see e.g. [72]).

The solution method is based on standard techniques of solving first order linear inhomogeneous ordinary sets of differential equations with constant coefficients, which can be found e.g. in [92]. The solution for the factorial moments can be expressed as the sum of the general solution of the homogeneous part and a particular solution of the inhomogeneous part of the set of differential equations. Since $\mathbf{D_1}$ and $\mathbf{D_0}$ matrices are constant, the solution of the homogeneous part remains the same for all moments. For $K \geq 2$, the solution of the inhomogeneous part of the set of differential equations for the Kth moment depends on the solution of the set of differential equations corresponding to the $(K-1)$st moment. The first step is computing $\boldsymbol{\mu}^{(1)}(t)$, then the same method can be used for the higher moments, accordingly.

The first step of solving Equation 6.1 is to obtain the solution of the homogeneous part of the equation:

$$\frac{d}{dt}\boldsymbol{\mu}^{(1)(H)}(t) = \mathbf{D}\boldsymbol{\mu}^{(1)(H)}(t) \tag{6.4}$$

The general solution is the linear combination of exponentials:

$$\mu_n^{(1)(H)}(t) = \sum_{m=1}^{M} V_{nm} c_m e^{r_m t} \tag{6.5}$$

The columns of the $M \times M$ matrix $\mathbf{V}$ are the eigenvectors of the constant coefficient matrix $\mathbf{D}$, r_ms are the corresponding eigenvalues, c_ms are unknown variables.

Theorem 6.1.2. *The matrix* $\mathbf{D}$ *has an eigenvalue equal to* 0*, and the real part of all other eigenvalues are negative.*

Proof. The theorem immediately follows from the fact that $\mathbf{D}$ is a stochastic matrix i.e. $\mathbf{De} = 0$. □

Remark 6.1.3. *Since one of the exponents is* 0*, the above solution of the homogeneous part of the set of differential equations has a term independent of* t*. This term plays an important role, when taking the limit* $t \to \infty$*.* $r_1 = 0$ *can be chosen without breaking the generality.*

Remark 6.1.4. *In case of complex eigenvalues, complex conjugate pairs occur in the set of roots of the characteristic polynomial of* $\mathbf{D}$ *and also in the exponents, resulting in real numbers in the solution.*

Remark 6.1.5. *In case of multiple eigenvalues, extra polynomials should be taken into account but the basic solution methodology remains the same. The details of solving Equation 6.4 taking into account multiple eigenvalues are shown later.*

For the general solution of the set of differential equations, one should also get one of the particular solutions. For this purpose, the method of undetermined coefficients can be applied, i.e. c_m $(1 \leq m \leq M)$ is considered as a function of t.

$$\sum_{m=1}^{M} V_{km} \dot{c_m}(t) e^{r_m t} = f_k(t), \tag{6.6}$$

where

$$f_k(t) = (1 - H(t)) \sum_{i=1}^{M} D_{1ki}$$

and it has the following general form:

$$f_k(t) = \sum_{i=1}^{I} e^{-\beta_{ik} t} \sum_{j=0}^{J_i} \alpha_{ij}^{(k)} t^j \tag{6.7}$$

The derivatives of the $c_m(t)$ functions $(1 \leq m \leq M)$ can be expressed by $f_k(t)$ functions and the inverse of $\mathbf{V}$. Let the $M \times M$ matrix $\mathbf{A}$ be the inverse of $\mathbf{V}$, $\mathbf{A} = \mathbf{V}^{-1}$. Solving Equation 6.6 for $\dot{\mathbf{c}}$, the following equations hold for the derivatives of the coefficients:

$$\dot{c_m}(t) = \sum_{k=1}^{M} A_{mk} f_k(t) e^{-r_m t} = \sum_{k=1}^{M} A_{mk} \sum_{i=1}^{I} e^{-(\beta_{ik}+r_m)t} \sum_{j=0}^{J_i} \alpha_{ij}^{(k)} t^j,$$

$$1 \leq m \leq M \tag{6.8}$$

For a particular term the integration can be performed as follows:

$$\int e^{-\delta t} t^j dt = -\frac{\Gamma(j+1)}{\delta^{(j+1)}} \sum_{l=0}^{j} \frac{e^{-\delta t}\delta^l t^l}{l!} + C$$

if $\delta \neq 0$ and in case of $\delta = 0$

$$\int t^j dt = \frac{t^{j+1}}{j+1} + C.$$

After the integration of both sides of Equation 6.8, the following formula for c_m is obtained (the constant part of the integration is taken as 0 for simplicity because a particular solution is sufficient):

$$c_m^{\diamond}(t) e^{r_m t} = \sum_{(\diamond)k=1}^{M} \sum_{(\diamond)i=1}^{I} \sum_{j=0}^{J_i} \sum_{l=0}^{j} A_{mk} \alpha_{ij}^{(k)} \frac{\Gamma(j+1)}{\delta_{ik}^{(m)\,(j+1)}} \frac{e^{-\beta_{ik} t} \delta_{ik}^{(m)\,l} t^l}{l!},$$

$$1 \leq m \leq M$$

where $\delta_{ik}^{(m)} = \beta_{ik} + r_m$. The diamond sign indicates that the above formula is evaluated only for those i and k indexes for that $\delta_{ik}^{(m)} \neq 0$. If $\delta_{ik}^{(m)} = 0$, those i and k indexes should be treated separately:

$$c_m^*(t)e^{r_m t} = \sum_{(*)k=1}^{M} \sum_{(*)i=1}^{I} \sum_{j=0}^{J_i} A_{mk} \alpha_{ij}^{(k)} \frac{t^{j+1}}{j+1} e^{r_m t}$$

where the star sign indicates that the summation is performed for only those i and k indexes for that $\delta_{ik}^{(m)} = 0$. The nth component of the particular solution can be written as

$$\mu_n^{\diamond(1)(P)}(t) = \sum_{m=1}^{M} V_{nm} c_m^{\diamond}(t) e^{r_m t} =$$

$$\sum_{(\diamond)m=1}^{M} \sum_{(\diamond)k=1}^{M} \sum_{(\diamond)i=1}^{I} \sum_{j=0}^{J_i} \sum_{l=0}^{j} V_{nm} A_{mk} \alpha_{ij}^{(k)} \frac{\Gamma(j+1)}{\delta_{ik}^{(m)\,(j+1)}} \frac{e^{-\beta_{ik} t} {\delta_{ik}^{(m)}}^{l} t^l}{l!}, \qquad 1 \leq n \leq M \tag{6.9}$$

where only those i, k and m indexes are taken into account for that $\delta_{ik}^{(m)} \neq 0$. In case of $\delta_{ik}^{(m)} = 0$, the following formula holds:

$$\mu_n^{*(1)(P)}(t) = \sum_{m=1}^{M} V_{nm} c_m^*(t) e^{r_m t} = \sum_{(*)m=1}^{M} \sum_{(*)k=1}^{M} \sum_{(*)i=1}^{I} \sum_{j=0}^{J_i} V_{nm} A_{mk} \alpha_{ij}^{(k)} \frac{t^{j+1}}{j+1} e^{r_m t}$$

$$1 \leq n \leq M \tag{6.10}$$

The particular solution of the Equation system 6.1 is the sum of the two different cases.

$$\mu_n^{(1)(P)}(t) = \mu_n^{\diamond(1)(P)}(t) + \mu_n^{*(1)(P)}(t) \tag{6.11}$$

Note that both $\mu_n^{\diamond(1)(P)}(t)$ and $\mu_n^{*(1)(P)}(t)$ converge with $t \to \infty$ to zero since $\beta_{ik} > 0$ for all i and k in the corresponding interval, $r_m > 0$ for all $2 \leq m \leq M$ and $r_1 = 0$ can not occur in $\mu_n^{*(1)(P)}(t)$ since in that case $\delta_{ik}^{(m)} = 0$ does not hold.

The general solution of the first factorial moment can then be derived by replacing Equations 6.5 and 6.11 in the following formula:

$$\mu_n^{(1)}(t) = \mu_n^{(1)(H)}(t) + \mu_n^{(1)(P)}(t) \tag{6.12}$$

In order to calculate c_m coefficients in Equation 6.5 the initial condition of Equation 6.1 can be used, thus

$$\mu_n^{(1)}(0) = \mu_n^{(1)(P)}(0) + \sum_{m=1}^{M} V_{nm} c_m = 0$$

where c_m is considered as constant. Using the notation $\mathbf{A} = \mathbf{V}^{-1}$ again, the vector of coefficients can be calculated by

$$\mathbf{c} = -\mathbf{A} * \boldsymbol{\mu}^{(1)(P)}(0) \tag{6.13}$$

The time-dependent solution of Equation 6.1 is thus described in Equation 6.12 where $\mu_n^{(1)(H)}(t)$ is specified in Equation 6.5 where the c_m expressed in Equation 6.13 should be replaced and $\mu_n^{(1)(P)}(t)$ is specified in Equation 6.11.

At last, the stationary solution for $\boldsymbol{\mu}^{(1)}(t)$ can be achieved by taking the limitation $t \to \infty$ in Equation 6.12. Since $\beta_{ik} > 0$ $(1 \leq i \leq I$,$1 \leq k \leq M)$, all terms of $\mu_n^{(1)(P)}(t)$ in Equation 6.11 goes to 0 if $t \to \infty$. Since $r_m < 0$ $(2 \leq m \leq M)$ and $r_1 = 0$ according to Theorem 6.1.2 and Remark 6.1.3, the asymptotic value of $\mu_n^{(1)}(t)$ is $V_{n1}c_1$ which is equal for all $1 \leq n \leq M$ according to Theorem 6.1.1.

6.1.3 Handling multiple eigenvalues

If there exists one or more eigenvalues of the $M \times M$ matrix $\mathbf{D}$ with multiplicity larger than one, the solution of Equation 6.4 can not be written as in Equation 6.5. However, the solution in this case can be achieved by letting the coefficient matrix elements depend on t as polynomials.

$$\mu_n^{(1)(H)}(t) = \sum_{(\Box)m=1}^{M} V_{nm}(t) c_m e^{r_m t}$$

where the square at the bottom of the term means that the eigenvalue with multiplicity larger than one is considered only once in the sum. If e.g. $r_1 \neq r_2 = r_3 = r_4 \neq r_5$ meaning that r_2 has multiplicity 3, then $m = 3$ and $m = 4$ is not taken into account. Assuming that one particular eigenvalue r_m has multiplicity Q, $V_{nm}(t)$ can be written as a polynomial with degree Q:

$$V_{nm}(t) = \sum_{j=1}^{Q} p_{nj} t^{j-1} \qquad 0 \leq n \leq M$$

The $M \times Q$ constant coefficient matrix $\mathbf{p}$ can then be achieved by replacing the particular term $V_{nm}(t) c_m e^{r_m t}$ into the original homogeneous differential equation and dividing by $c_m e^{r_m t}$. If the matrix elements of $\mathbf{p}$ are written in a column-vector $\hat{\mathbf{p}}$ such that

$$\hat{\mathbf{p}} = [[p_{11} \dots p_{M1}] \dots [p_{1Q} \dots p_{MQ}]]^T,$$

the following equation system is obtained for the p_{nj} variables:

$$\begin{bmatrix} \mathbf{F}_m & -\mathbf{E} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_m & -2\mathbf{E} & \mathbf{0} & \vdots \\ \mathbf{0} & \mathbf{0} & \ddots & \ddots & \mathbf{0} \\ \vdots & \vdots & & & -(Q-1)\mathbf{E} \\ \mathbf{0} & \mathbf{0} & \dots & & \mathbf{F}_m \end{bmatrix} [\hat{\mathbf{p}}] = [\mathbf{0}]$$

where $\mathbf{F}_m = \mathbf{D} - r_m\mathbf{E}$ and $\mathbf{E}$ stands for the $M \times M$ unity matrix. The solution of the above equation system can be expressed by exactly Q free parameters chosen from the different columns of the $\mathbf{p}$ matrix.

The solution of the homogeneous system achieved this way differs from the solution shown in Equation 6.5 in that it may have polynomial coefficients. It does not cause any problems afterwards, since it falls in the function class described by Equation 6.7. Additionally, obtaining the particular solution of the inhomogeneous differential equation by using the well-known method of undetermined coefficients can be done in the same way as in the calculations before. Some of the coefficients are those of the polynomials in this case.

6.1.4 Solution for the higher moments

Since the general form of the basic equation system does not change during the derivation of the particular factorial moments, the solution of the equations can be obtained iteratively by updating the second term of the right-hand side of Equation 6.2 by Formula 6.3. This way $\mu_n^{(K)}(t)$ can be computed for any $K \geq 1$ and $1 \leq n \leq M$. The asymptotic behaviour of the system can then be obtained by taking the limit $t \to \infty$ in each step. From the factorial moments, the moments of the queue length can be obtained by simple computations. Practically the iteration steps can be performed by using e.g. Matlab, appropriately defining the initial β vector and α matrix and updating them according to the above method.

As a summary, the sets of differential equations 6.1 and 6.2 can be solved exactly if the distribution of the service time (whose c.d.f. is $H(t)$) is a mix of exponentials. A practical limitation is that due to the complexity, only the solutions for the first few factorial moments can be achieved in a considerable amount of computational time. A numerical application and evaluation of the method can be found in the following example.

6.2 Application example

In this section the usage of the model is illustrated by a technical application. The system under investigation consists of processing modules of a Web-based content provider (e.g. a news service using Multimedia Messaging Service, MMS). The service operates via a central file-server containing the news items and different multimedia objects (pictures, videos, animations). The server can identify the type of browser a certain request is sent from. The appearance of the article depends on the terminal type so the server has to optimise it according to the type of the browser. The aim of the operator is to send the messages in a format that appears in as good quality as possible (e.g. the size and resolution of the pictures or the rendering of the text). Web servers may adapt to the limited capabilities of mobile equipments as well in order to improve the performance and the quality of browsing.

The content provider consists of a central server, converter units and a storage unit. The operation steps of the service is illustrated in Figure 6.1. The timing sequence of the events is represented by the numbers next to the arrows.

1. When a new article arrives from the news agency at the server a basic version is generated.

2. A message arrives from a user requesting for the article.

3. The server converts the basic version according to the browser type.

4. Since there may be another user with the same browser type requesting this article the converted version is stored temporarily so that the conversion need not be performed once again.

5. The storage unit sends the requested version of the article to the server.

6. The server forwards the article to the user.

When a request arrives from a browser of type that has been processed so far, the server can turn to the storage unit. For a new browser type the conversion has to be made. Note that the converters are not necessarily separate processing units, they can be different program threads on the same processor.

The purpose is to investigate the functionality of the conversion from the publication of a new article by the agency to the state where the article is converted to all possible forms. In order for the fast service more than one conversion can

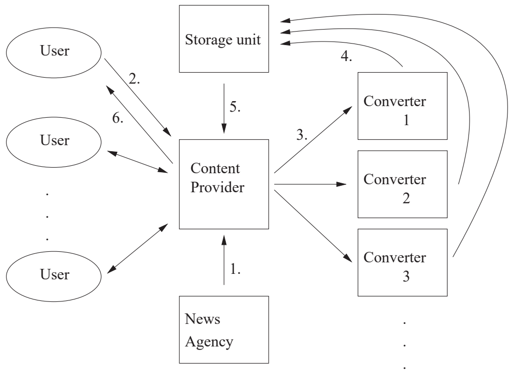

Figure 6.1: Outline of the news service.

be processed at the same time. If the number of converters is not large enough all the converters may be occupied resulting in rejection of a request coming from a terminal which has a new type. Increasing the number of converters may meet financial limits. The task is to fix an upper limit for the processing converters where the probability of being all of them occupied is very small.

6.2.1 Mathematical modeling

Matrix analytic methods can be applied to the above model in the following way. If there are n types of browser, matrices $\mathbf{D}$, $\mathbf{D}_0$ and $\mathbf{D}_1$ can be built from the relative frequency of the occurrence of each browser types and the arrival intensity of the requests from the users. Table 6.1 shows the relative frequencies of n browser types.

Type	1	2	...	n
Frequency	F_1	F_2	...	F_n

Table 6.1: Share of the different types of browsers (frequency).

Since F_is are relative frequencies, $\sum_{i=1}^{n} F_i = 1$. If all F_is are different, the number of states in the MAP is $S = 2^n$ thus, the $S \times S$ matrix $\mathbf{D}$ describing the model becomes rather large. However, the size of the matrix can be reasonably decreased by letting the frequency of some types the same so as n_i browser types have the same frequency. Table 6.2 shows such a scenario, here $\sum_{i=1}^{k} n_i F_i = 1$. The resulting number of states is $S = \prod_{i=1}^{k} n_k$.

Type	1	2	...	n_1	n_1+1	n_1+2	...	n_1+n_2	...	$n_1+...+n_k$
Frequency	F_1	F_1	...	F_1	F_2	F_2	...	F_2	...	F_k

Table 6.2: Share of the different types of browsers with reduced number of different frequencies.

The states represent the number of messages of certain browser types that have arrived so far. State transitions may only occur only if a new type of request arrives. If the requests have Poisson arrival with rate λ, the elements of the MAP representation matrix $\mathbf{D}$ can be built from λ multiplied by the proper frequency values. For details see Section 6.2.2.

6.2.2 Numerical example

An example is shown where the time-dependent moments of the queue-length of a $MAP/M/\infty$ queuing system is computed. Let's assume that the requests of the users arrive according to Poisson process with intensity 8 requests per sec. In the example 10 different types of terminals are known with different converting procedures. The time of conversion is exponentially distributed with average 5 seconds (though the model can handle more complex distributions).

$$H(t) = 1 - e^{-\frac{t}{5}}.$$

The share of the 10 different types and their average number of requests in one second is summarised in Table 6.3. Note that the sum of the *Frequencies* is 100% and the sum of the *Intensities* is 8 1/s which corresponds to the above assumption on the request intensity.

Type	Frequency	Intensity
1. type	21%	1.68
2. type	20%	1.6
3. type	13%	1.04
4. type	13%	1.04
5. type	13%	1.04
6. type	4%	0.32
7. type	4%	0.32
8. type	4%	0.32
9. type	4%	0.32
10. type	4%	0.32

Table 6.3: The share of the different types of browsers (frequency) and the number of requests in a second generated by them (intensity).

Frequency	Types	Intensity
21%	1.	1.68
20%	2.	1.6
13%	3., 4., 5.	1.04
4%	6., 7., 8., 9., 10.	0.32

Table 6.4: Different types of browsers sorted by their intensities.

One can see from Table 6.4 that the state space of the underlying MAP can be described by 4-tuples. These vectors point out which terminal types have a properly converted version of the latest article in the storage unit. The meaning of the elements of the 4-tuples is:

1: If the article is not converted for type 1 then its value is 0 otherwise 1.

2: If the article is not converted for type 2 then its value is 0 otherwise 1.

3: Conversions of types 3-5 are counted here. Its value can be between 0 and 3. The arrival order does not matter since types 3-5 have the same frequency.

4: Conversions of types 6-10 are counted here. Its value can be between 0 and 5. The arrival order does not matter since types 6-10 have the same frequency.

The number of states is $2 \cdot 2 \cdot 4 \cdot 6 = 96$. State transition is allowed only between states whose 4-vector representation differs in only one digit. The initial state is $(0, 0, 0, 0)$, i.e. the storage unit is still empty. The state transitions are given in a 96×96 matrix $\mathbf{D_1}$ in the following way (see Figure 6.2).

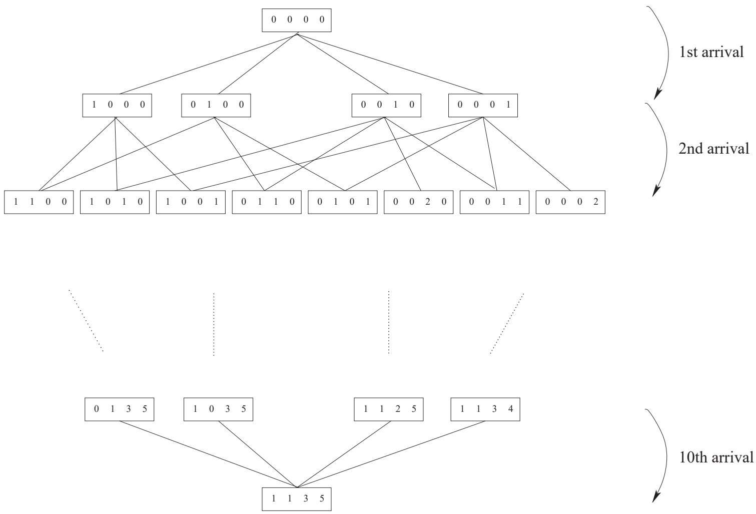

Figure 6.2: State transition upon arrivals.

If the state transition corresponds to the conversion of terminal type 1 then the value of the matrix element is $8\ 1/s \cdot 0.21 = 1.68\ 1/s$ as it can be seen in Table 6.3 in column "Intensity". The matrix element of the state transition corresponding the conversion of terminal type 2 is $1.6\ 1/s$.

In case of types 3-5 the corresponding state transition changes the 3rd vector element. In this case the number of conversions needs to be maintained as well since the first request is expected to arrive with intensity $3 \cdot 1.04\ 1/s = 3.12\ 1/s$

from one of types 3-5. However, after the first request, only the remaining two types can generate new requests so the intensity decreases to $2 \cdot 1.04\ 1/s = 2.08\ 1/s$. If 2 types are already processed from types 3-5, the arrival intensity of the remaining request is $1.04\ 1/s$.

The case of types 6-10 is similar to the above case of types 3-5. Here the initial intensity is $5 \cdot 0.32\ 1/s = 1.6\ 1/s$ and it decreases to $0.32\ 1/s$ if one type is left to be processed.

From the publishing of the article the requests arrive continuously to the server. After the first types are processed, the number of needed conversions is smaller. If 10 converters are available, all demands can be served. However, it will be shown that the service can be completed with a smaller number of converters.

Since every request from a new type generates a state transition, $\mathbf{D_0}$ has only diagonal elements, each of them assigned so that $\mathbf{D} = \mathbf{D}_0 + \mathbf{D}_1$ is stochastic, i.e. the sum of the elements in a row is 0. By replacing the $\mathbf{D}$ and $\mathbf{D_1}$ matrices and the $H(t)$ function into Equations 6.1 and 6.2 and solving them following the steps shown before, the time-dependent moments can be obtained. The formulae are evaluated by Matlab.

In Figure 6.3 the time evolution of the average number of busy converters is depicted. It can be seen clearly that the largest number of working converters is expected to operate 0.75 seconds after the news publishing.

In order to find a reasonable limit for the number of available converters where the probability of saturation is small at the maximum utilisation, the moments of the number of busy converters are evaluated 0.75 seconds after the publishing of the new article. Assuming that unlimited number of available converters exists, Table 6.5 contains the results.

Since the number of conversions is at most 10, the distribution of the number of parallel conversions can be computed from the moments with the help of a Vandermonde-type matrix [93]. Table 6.6 shows the probability of the number of parallel conversions (N) exceeding a given limit (n).

If the system was designed so that the probability of reaching the capacity limit should be less than 0.1%, from Table 6.6 it can be deduced that instead of 10 converters, 5 converter units or processing capacity corresponding to 5 parallel processing threads would be sufficient.

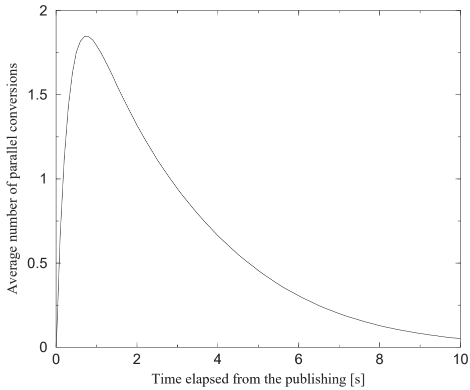

Figure 6.3: Time evolution of the average number of busy converters after the publishing of a new article.

	Factorial moment	Moment
1.	1.848	1.848
2.	3.063	4.911
3.	4.499	15.537
4.	5.763	56.048
5.	6.307	224.208
6.	5.734	976.641
7.	4.157	4573.540
8.	2.253	22805.388
9.	0.812	120181.397
10.	0.146	665327.954

Table 6.5: Moments of the number of parallel conversions 0.75 seconds after the publishing of the new article.

n	$P(N > n)$
0	0.871
1	0.577
2	0.276
3	0.095
4	0.023
5	0.004
6	$5 \cdot 10^{-4}$
7	$4 \cdot 10^{-5}$
8	$2 \cdot 10^{-6}$
9	$3 \cdot 10^{-8}$
10	0

Table 6.6: Distribution function of the number of parallel conversions 0.75 seconds after the publishing of the new article.

6.3 Conclusions

In this chapter a mathematical model is introduced and a possible application of the model is shown. The transient behaviour of the first moment of a MAP/PH/∞ queuing system is determined exactly by setting up and solving an inhomogeneous linear set of differential equations. The solution is iterated several times to obtain the higher moments. This model can be used to describe transient behaviour of systems with parallel servers, general arrival and general processing times. The applicability of the computational method is illustrated by solving a dimensioning problem of content and multimedia servers.

Chapter 7

Minimising complexity in matrix analytic functions

Matrix analytic representations play important role in queuing analysis. The purpose of reducing the number of states in PH representations is to minimise the complexity of numerical methods.

A special type of the PH-representations is the triangular representation, where the elements of the generator matrix are non-zero only at the diagonal elements and above (upper triangular representation) or only at the diagonal elements and below (lower triangular representation). A special case of triangular representations is the bi-diagonal representation where only the bi-diagonal elements of the generator matrix are non-zero. A bi-diagonal representation is also called Coxian representation [83]. The Coxian representation is called ordered Coxian, if the diagonal entries of the generator matrix are decreasing.

Triangular representations and Coxian representations are sparse and less complex than the general ones. However, the triangular order of a PH distribution (i.e. the minimal number of states the distribution can be represented with triangular matrix) is generally higher than the order. Several statements have been presented on the Coxian representations, triangular representations and triangular order of the PH-distributions. It was shown in [85] that any upper triangular PH representation has an equivalent ordered Coxian representation of the same or smaller order.

There are attempts to give lower bounds for the number of states needed in a PH representation based on some knowledge of the distribution. For example, the main theorem of [84] shows that the order n Erlang distribution has the smallest coefficient of variation among the order n PH distributions, that is, the coefficient of variation can be used to calculate a lower bound for the order of the PH distributions. Another related result, which gives such bound in the case of complex conjugate poles, is Theorem 3.1 in [81]. The authors in [82] characterise the minimal order of upper triangular PH representations for PH distributions with one real pole of multiplicity at most 3. Besides, in [88] bounds on the PH order of PH distributions subject to certain conditions are collected. An important goal in [87] is to find a smaller PH representation given an existing one. It is proved there that any PH representation with order 3 with only real eigenvalues has an ordered Coxian representation of order 4 or smaller order.

The objective of this thesis is to find a method to construct upper triangular representations to PH distributions using as few states as possible. The target set of functions is the absolutely continuous PH distributions (i.e. no weight at 0) with 3 distinct real poles in their Laplace-transform.

A method is shown how to decompose this set into subsets containing distributions possessing order 3 and higher order upper triangular PH representations. The decomposition is based on the concept of invariant polytopes defined in [80]. Moreover, it is shown how to build the PH representation from an invariant polytope.

7.1 Definitions and basic theorems

A short summary of the most important notations and definitions that need to be introduced to make the relevant statements is given below. Basic lemmas and theorems are also proved.

Definition 7.1.1. *$\mathcal{ME}_{\lambda_1,\lambda_2,\lambda_3}$ is the convex set of matrix-exponential distributions with distinct real poles $-\lambda_1, -\lambda_2, -\lambda_3$ ($\lambda_1, \lambda_2, \lambda_3 > 0$). $\mathcal{ME}^{\delta}_{\lambda_1,\lambda_2,\lambda_3}$ is the exten-*

sion of this set with the Dirac-delta function.

$$\mathcal{ME}_{\lambda_1,\lambda_2,\lambda_3} = \big\{ f(t),\ t \in \mathbb{R}_0^+ : f(t) \in \mathbb{R}_0^+,$$

$$\exists \mathbf{a} \in \mathbb{C}^3, \exists \mathbf{M} \in \mathbb{C}^{3\times 3},\ \mathrm{sp}(\mathbf{M}) = \{-\lambda_1, -\lambda_2, -\lambda_3\}, f(t) = \mathbf{a}^T e^{\mathbf{M}\cdot t}\mathbf{e}\big\}.$$

$$\mathcal{ME}^{\delta}_{\lambda_1,\lambda_2,\lambda_3} = \mathrm{co}\big\{\{\delta_0\}, \mathcal{ME}_{\lambda_1,\lambda_2,\lambda_3}\big\},$$

where $\mathrm{co}\{\dots\}$ *denotes the convex hull of the union of the sets listed.*

That is, for each $f(t) \in \mathcal{ME}_{\lambda_1,\lambda_2,\lambda_3}$ there is a representation $(\mathbf{a}, \mathbf{M})$, where $\mathbf{M}$ has eigenvalues $-\lambda_1, -\lambda_2, -\lambda_3$. The function represented by $(\mathbf{a}, \mathbf{M})$ is denoted by $f_{(\mathbf{a},\mathbf{M})}$.

The algebraical form of a density function of the Phase-type distribution $f(t)$ with three distinct real poles is

$$f(t) = \alpha_0\delta_0(t) + \sum_{i=1}^{3} \alpha_i e^{-\lambda_i\cdot t}, \quad \alpha_0, \alpha_i \in \mathbb{R}, \lambda_i \in \mathbb{R}_+, i = 1, 2, 3$$

Following the path of [80] two linear operators are defined on $\mathcal{ME}^{\delta}_{\lambda_1,\lambda_2,\lambda_3}$.

Definition 7.1.2. *Let a linear operator* $\mathbf{R}_t : \mathcal{ME}^{\delta}_{\lambda_1,\lambda_2,\lambda_3} \to \mathcal{ME}^{\delta}_{\lambda_1,\lambda_2,\lambda_3}$ *be*

$$\mathbf{R}_t\left(\alpha_0\delta_0(u) + f_{(\mathbf{a},\mathbf{M})}(u)\right) = \beta_0\delta_0(u) + f_{(\mathbf{b},\mathbf{M})}(u),$$

where $\beta_0 = \alpha_0 + \int_0^t f_{(\mathbf{a},\mathbf{M})}(u)du$ *and* $\mathbf{b} = \mathbf{a}^T e^{\mathbf{M}\cdot t}$.

Let $\mathbf{\Gamma} : \mathcal{ME}^{\delta}_{\lambda_1,\lambda_2,\lambda_3} \to \mathcal{ME}^{\delta}_{\lambda_1,\lambda_2,\lambda_3}$ *be*

$$\mathbf{\Gamma}\left(\alpha_0\delta_0(u) + f_{(\mathbf{a},\mathbf{M})}(u)\right) =$$

$$\lim_{t\to 0} \frac{\mathbf{R}_t\left(\alpha_0\delta_0(u) + f_{(\mathbf{a},\mathbf{M})}(u)\right) - \mathbf{R}_0\left(\alpha_0\delta_0(u) + f_{(\mathbf{a},\mathbf{M})}(u)\right)}{t}$$

The operator $\mathbf{R}_t$ shifts the continuous part of the distribution to the left with t and that part getting into $(-\infty, 0)$ is transformed to the mass at zero. The following lemma is also adopted from [80].

Lemma 7.1.3. *Let $f(u) \in \mathcal{ME}^{\delta}_{\lambda_1,\lambda_2,\lambda_3}$ be written in the following form*

$$f(u) = \alpha_0\delta_0(u) + \sum_{i=1}^{3} \alpha_i e^{-\lambda_i \cdot u},$$

then

$$\mathbf{R}_t\left(f(u)\right) = \left(\alpha_0 + \sum_{i=1}^{3} \frac{\alpha_i}{\lambda_i}(1 - e^{-\lambda_i \cdot t})\right)\delta_0(u) + \sum_{i=1}^{3} \alpha_i e^{-\lambda_i \cdot (t+u)}$$

and

$$\mathbf{\Gamma}\left(f(u)\right) = \sum_{i=1}^{3} \alpha_i \delta_0(u) - \sum_{i=1}^{3} \alpha_i \lambda_i e^{-\lambda_i \cdot u}.$$

The $\mathcal{ME}^{\delta}_{\lambda_1,\lambda_2,\lambda_3}$ convex set is a subset of a 4 dimensional vector space of functions $\mathbb{V}$. The following distributions form a basis of the vector space:

$$\delta_0, \lambda_1 e^{-\lambda_1 \cdot t}, \lambda_2 e^{-\lambda_2 \cdot t}, \lambda_3 e^{-\lambda_3 \cdot t}$$

In the following, all vectors will be expressed with coordinates in this basis, which shall later be referred to as canonical basis. These vectors can be written in the form of $(\eta_0; \eta_1, \eta_2, \eta_3)$. The effect of $\mathbf{R}_t$ and $\mathbf{\Gamma}$ can be expressed in the canonical basis as

$$\mathbf{R}_t\left((\eta_0; \eta_1, \eta_2, \eta_3)\right) = \left(\eta_0 + \sum_{i=1}^{3} \eta_i (1 - e^{-\lambda_i \cdot t}); \eta_1 e^{-\lambda_1 \cdot t}, \eta_2 e^{-\lambda_2 \cdot t}, \eta_3 e^{-\lambda_3 \cdot t}\right) =$$

$$= \left(1 - \sum_{i=1}^{3} \eta_i e^{-\lambda_i \cdot t}; \eta_1 e^{-\lambda_1 \cdot t}, \eta_2 e^{-\lambda_2 \cdot t}, \eta_3 e^{-\lambda_3 \cdot t}\right). \tag{7.1}$$

$$\mathbf{\Gamma}\left((\eta_0;\eta_1,\eta_2,\eta_3)\right) = \left(\sum_{i=1}^{3}\eta_i\lambda_i; -\eta_1\lambda_1, -\eta_2\lambda_2, -\eta_3\lambda_3\right). \tag{7.2}$$

Definition 7.1.4. *Let $f(\cdot) \in \mathcal{ME}^{\delta}_{\lambda_1,\lambda_2,\lambda_3}$ given by $(\eta_0;\eta_1,\eta_2,\eta_3)$, where*

$$f(t) = \eta_0\delta_0(t) + \sum_{i=1}^{3}\eta_i\lambda_i e^{-\lambda_i\cdot t},$$

and $\eta_0+\eta_1+\eta_2+\eta_3 = 1$. The non-linear operator $\mathbf{L}$ is defined as

$$\mathbf{L} : \mathcal{ME}^{\delta}_{\lambda_1,\lambda_2,\lambda_3} \to \mathcal{ME}^{\delta}_{\lambda_1,\lambda_2,\lambda_3}$$

$$\mathbf{L}\left((\eta_0;\eta_1,\eta_2,\eta_3)\right) = \left(0; \frac{\eta_1}{\sum_{i=1}^{3}\eta_i}, \frac{\eta_2}{\sum_{i=1}^{3}\eta_i}, \frac{\eta_3}{\sum_{i=1}^{3}\eta_i}\right). \tag{7.3}$$

In this thesis representations are investigated for distributions with absolutely continuous density functions given as

$$f(t) = \sum_{i=1}^{3}\eta_i\lambda_i e^{-\lambda_i\cdot t}, \tag{7.4}$$

for which $\lambda_3 > \lambda_2 > \lambda_1 > 0$. Since $f(t)$ is a probability density function, $\sum_{i=1}^{3}\eta_i = 1$ should hold. It is obvious that absolutely continuous distribution functions are within a 2 dimensional subspace in $\mathbb{V}$. Thus, it is possible to define a bijection between this affine plane and $\mathbb{R}^2$.

Definition 7.1.5. *Let $f(\cdot) \in \mathcal{ME}_{\lambda_1,\lambda_2,\lambda_3}$ be an absolutely continuous distribution, which can be expressed in the canonical basis as*

$$f(\cdot) = (0;\eta_1,\eta_2,\eta_3)$$

and $\eta_1+\eta_2+\eta_3 = 1$, that is $\mathbf{L}(f(\cdot)) = f(\cdot)$. Define the operator $\mathbf{T}$ as

$$\mathbf{T} : \mathbb{V} \to \mathbb{R}^2 \quad \mathbf{T}\left((0;\eta_1,\eta_2,\eta_3)\right) = (\eta_1,\eta_2).$$

The operator $\mathbf{T} \circ \mathbf{L}$ maps $\mathcal{ME}^{\delta}_{\lambda_1,\lambda_2,\lambda_3}$ to $\mathbb{R}^2$. The operators corresponding to $\mathbf{R}_t$ and $\mathbf{\Gamma}$ can also be defined on $\mathbb{R}^2$.

Definition 7.1.6. *Let the $\mathbf{W}_t : \mathbb{R}^2 \to \mathbb{R}^2$ operator be defined as*

$$\mathbf{W}_t(\eta_1, \eta_2) = \left(\frac{\eta_1 e^{-\lambda_1 \cdot t}}{\sum_{i=1}^{3} \eta_i e^{-\lambda_i \cdot t}}, \frac{\eta_2 e^{-\lambda_2 \cdot t}}{\sum_{i=1}^{3} \eta_i e^{-\lambda_i \cdot t}} \right),$$

where $\eta_3 = 1 - \eta_1 - \eta_2$. Let the $\hat{\mathbf{\Theta}} : \mathbb{R}^2 \to \mathbb{R}^2$ operator be defined as

$$\hat{\mathbf{\Theta}}(\eta_1, \eta_2) = \lim_{t \to 0} \frac{\mathbf{W}_t(\eta_1, \eta_2) - \mathbf{W}_0(\eta_1, \eta_2)}{t}$$

A direct expression for $\hat{\mathbf{\Theta}}$ is the following

$$\begin{aligned}\hat{\mathbf{\Theta}}(\eta_1, \eta_2) = \Big(&\eta_1(\eta_1(\lambda_1 - \lambda_3) + \eta_2(\lambda_2 - \lambda_3) + \lambda_3 - \lambda_1), \\ &\eta_2(\eta_1(\lambda_1 - \lambda_3) + \eta_2(\lambda_2 - \lambda_3) + \lambda_3 - \lambda_2)\Big). \end{aligned} \tag{7.5}$$

Note that the operators $\mathbf{W}_t$ and $\hat{\mathbf{\Theta}}$ in $\mathbb{R}^2$ correspond to $\mathbf{R}_t$ and $\mathbf{\Gamma}$ in $\mathcal{ME}^{\delta}_{\lambda_1,\lambda_2,\lambda_3}$, respectively. The claim of the following lemma is that $\mathbf{W}_t$ is consistent to $\mathbf{R}_t$ in the desired way.

Lemma 7.1.7. *Let $(\eta_1, \eta_2) \in \mathbb{R}^2$, $\eta_3 = 1 - \eta_1 - \eta_2$ and $f(t) = \sum_{i=1}^{3} \eta_i e^{-\lambda_i t}$, i.e. $f(\cdot) \in \mathcal{ME}^{\delta}_{\lambda_1,\lambda_2,\lambda_3}$ and the integral of $f(\cdot)$ is 1. Then*

$$\forall t \in \mathbb{R}_0^+ : \mathbf{W}_t(\eta_1, \eta_2) = \mathbf{W}_t \circ \mathbf{T}(f(\cdot)) = \mathbf{T} \circ \mathbf{L} \circ \mathbf{R}_t(f(\cdot)).$$

The proof is straightforward using the definitions. The following lemma highlights an important property of $\mathbf{R}_t$.

Lemma 7.1.8. *Let $f(\cdot) \in \mathcal{ME}_{\lambda_1,\lambda_2,\lambda_3}$ be an absolutely continuous distribution, which can be expressed in the canonical basis as $f(\cdot) = (0; \eta_1, \eta_2, \eta_3)$, ($\eta_1 + \eta_2 + \eta_3 = 1$).*

Then for all such $f(\cdot) \in \mathcal{ME}_{\lambda_1,\lambda_2,\lambda_3}$ and $t \geq 0$ ($t \in \mathbb{R}$): $\exists c \in [0, 1]$ such that

$$\mathbf{R}_t(f(\cdot)) = \delta_0 + c\left(\mathbf{L}(\mathbf{R}_t(f(\cdot))) - \delta_0\right).$$

In other words, $\mathbf{R}_t(f(\cdot))$ is the convex combination of δ_0 and $\mathbf{L}(\mathbf{R}_t(f(\cdot))$.

Proof. Let c be chosen such that $c = \sum_{i=1}^{3} \eta_i e^{-\lambda_i \cdot t}$.

$$\mathbf{L}(\mathbf{R}_t(f(\cdot))) = \left(0; \frac{\eta_1 e^{-\lambda_1 \cdot t}}{\sum_{i=1}^{3} \eta_i e^{-\lambda_i \cdot t}}, \frac{\eta_2 e^{-\lambda_2 \cdot t}}{\sum_{i=1}^{3} \eta_i e^{-\lambda_i \cdot t}}, \frac{\eta_3 e^{-\lambda_3 \cdot t}}{\sum_{i=1}^{3} \eta_i e^{-\lambda_i \cdot t}}\right),$$

$$\delta_0 + c\left(\mathbf{L}(\mathbf{R}_t(f(\cdot))) - \delta_0\right) = \left(1 - \sum_{i=1}^{3} \eta_i e^{-\lambda_i \cdot t}; \eta_1 e^{-\lambda_1 \cdot t}, \eta_2 e^{-\lambda_2 \cdot t}, \eta_3 e^{-\lambda_3 \cdot t}\right).$$

If $f(t) = \sum_{i=1}^{3} \eta_i \lambda_i e^{-\lambda_i \cdot t}$ is a probability density function of an absolutely continuous distribution, then the cumulative density function is $F(t) = 1 - \sum_{i=1}^{3} \eta_i e^{-\lambda_i \cdot t}$. Since $F(t) \in [0,1]$, the constant in Lemma 7.1.8 is $c \in [0,1]$. That is, $\mathbf{R}_t(f(\cdot))$ is the convex combination of δ_0 and $\mathbf{L}(\mathbf{R}_t(f(\cdot)))$. $\square$

Definition 7.1.9. *Let $f_1(\cdot), \ldots, f_n(\cdot) \in \mathcal{ME}^{\delta}_{\lambda_1,\lambda_2,\lambda_3}$ be arbitrary vectors. Denote by $\mathrm{co}\{f_1(\cdot), \ldots, f_n(\cdot)\}$ the convex hull of these vectors, forming a polytope in $\mathbb{V}$.*

The $\mathrm{co}\{f_1(\cdot), \ldots, f_n(\cdot)\}$ set is said to be $\mathbf{R}_t$-invariant if

$$\forall t \in \mathbb{R}_0^+, \forall f(\cdot) \in \mathrm{co}\{f_1(\cdot), \ldots, f_n(\cdot)\} \;:\; \mathbf{R}_t\left(f(\cdot)\right) \in \mathrm{co}\{f_1(\cdot), \ldots, f_n(\cdot)\}.$$

It is apparent that if $\mathrm{co}\{f_1(\cdot), \ldots, f_n(\cdot)\}$ is $\mathbf{R}_t$-invariant then it contains the whole trajectory of $f(\cdot)$ starting from inside the $\mathbf{R}_t$-invariant set and vice versa. From Lemma 7.1.8 the following theorem follows:

Theorem 7.1.10. *If one finds a polytope in the 2 dimensional affine plane entirely containing the orbit $\mathbf{L}(\mathbf{R}_t(f(\cdot)))$, then the convex hull of this polytope and δ_0 entirely contains $\mathbf{R}_t(f(\cdot))$ given that $f(\cdot)$ is an absolutely continuous distribution.*

In Lemma 3.4 in [80] a nice and simple necessary and sufficient condition is given for a polytope $\mathrm{co}\{\delta_0(\cdot), f_1(\cdot), \ldots, f_n(\cdot)\}$ to be $\mathbf{R}_t$-invariant. For this condition, the concept of "pointing inward" is defined.

Definition 7.1.11. *Let $\mathrm{co}\{\delta_0(\cdot), f_1(\cdot), \ldots, f_n(\cdot)\}$ be a polytope in $\mathcal{ME}^{\delta}_{\lambda_1,\lambda_2,\lambda_3}$. Then $\mathbf{\Gamma}(f_i(\cdot))$ points inward to the polytope if*

$$\exists \epsilon > 0, \forall \delta \in [0, \epsilon) : f_i(\cdot) + \delta \mathbf{\Gamma}(f_i(\cdot)) \in \mathrm{co}\{\delta_0(\cdot), f_1(\cdot), \ldots, f_n(\cdot)\}.$$

According to Lemma 3.4 in [80], the following theorem holds:

Theorem 7.1.12. *A polytope* $\mathrm{co}\{\delta_0(\cdot), f_1(\cdot), \dots, f_n(\cdot)\}$ *is* $\mathbf{R}_t$*-invariant if and only if* $\forall i = 1, \dots, n$ *the vector* $\mathbf{\Gamma}(f_i(\cdot))$ *"points inward" to the polytope.*

The following theorem states that $\mathbf{R}_t$-invariance of a polytope in $\mathcal{ME}^{\delta}_{\lambda_1,\lambda_2,\lambda_3}$ is related to the "point inward" property of the set transformed to $\mathbb{R}^2$.

Theorem 7.1.13. *Let* $f_1(\cdot), f_2(\cdot), \dots, f_n(\cdot) \in \mathcal{ME}_{\lambda_1,\lambda_2,\lambda_3}$*. Then the* $\mathrm{co}\{\delta_0(\cdot), f_1(\cdot), \dots, f_n(\cdot)\}$ *polytope is* $\mathbf{R}_t$*-invariant if and only if* $\hat{\mathbf{\Theta}} \circ \mathbf{T}(f_i(\cdot))$ *points inward to the* $\mathrm{co}\{\mathbf{T}(f_1(\cdot)), \dots, \mathbf{T}(f_n(\cdot))\}$ *polytope* $\forall i = 1, \dots, n$*.*

Proof. Assuming that $\mathrm{co}\{\delta_0(\cdot), f_1(\cdot), \dots, f_n(\cdot)\}$ is $\mathbf{R}_t$-invariant

$$\exists \epsilon > 0, \forall \delta \in [0, \epsilon), \forall i = 1, \dots, n \;:\; \mathbf{R}_\delta(f_i(\cdot)) \in \mathrm{co}\{\delta_0(\cdot), f_1(\cdot), \dots, f_n(\cdot)\}.$$

From this it follows that

$$\exists \epsilon > 0, \forall \delta \in [0, \epsilon), \forall i = 1, \dots, n \;:\; \mathbf{W}_\delta \circ \mathbf{T}(f_i(\cdot)) \in \mathrm{co}\{\mathbf{T}(f_1(\cdot)), \dots, \mathbf{T}(f_n(\cdot))\}.$$

Then, according to Definition 7.1.6,

$$\exists \epsilon' > 0, \forall \delta' \in [0, \epsilon'), \forall i = 1, \dots, n \;:$$

$$\mathbf{W}_0 \circ \mathbf{T}(f_i(\cdot)) + \delta' \hat{\mathbf{\Theta}} \circ \mathbf{T}(f_i(\cdot)) \in \mathrm{co}\{\mathbf{T}(f_1(\cdot)), \dots, \mathbf{T}(f_n(\cdot))\},$$

that is, $\hat{\mathbf{\Theta}} \circ \mathbf{T}(f_i(\cdot))$ points inward to $\mathrm{co}\{\mathbf{T}(f_1(\cdot)), \dots, \mathbf{T}(f_n(\cdot))\}$ ($1 \le i \le n$).

Now assume that $\forall i = 1, \dots, n$ the $\hat{\mathbf{\Theta}} \circ \mathbf{T}(f_i(\cdot))$ vector points inward to $\mathrm{co}\{\mathbf{T}(f_1(\cdot)), \dots, \mathbf{T}(f_n(\cdot))\}$. Then

$$\exists \epsilon > 0, \forall \delta \in [0, \epsilon), \forall i = 1, \dots, n \;:$$

$$\mathbf{T}(f_i(\cdot)) + \delta \hat{\mathbf{\Theta}} \circ \mathbf{T}(f_i(\cdot)) \in \mathrm{co}\{\mathbf{T}(f_1(\cdot)), \dots, \mathbf{T}(f_n(\cdot))\}.$$

According to Definition 7.1.6,

$$\exists \epsilon' > 0, \forall \delta' \in [0, \epsilon'), \forall i = 1, \ldots, n \ :$$

$$\mathbf{W}_{\delta'} \circ \mathbf{T}(f_i(\cdot)) \in \mathrm{co}\{\mathbf{T}(f_1(\cdot)), \ldots, \mathbf{T}(f_n(\cdot))\}.$$

In this case $\mathbf{W}_{\delta'} \circ \mathbf{T}(f_i(\cdot))$ can be obtained as a convex combination:

$$\mathbf{W}_{\delta'} \circ \mathbf{T}(f_i(\cdot)) = \sum_{j=1}^{n} \alpha_j \mathbf{T}(f_j(\cdot)), \quad \sum_{j=1}^{n} \alpha_j = 1, \quad \alpha_j \geq 0 \quad \forall j$$

Since $\mathbf{T}$ is a linear map,

$$\mathbf{W}'_{\delta} \circ \mathbf{T}(f_i(\cdot)) = \mathbf{T}\left(\sum_{j=1}^{n} \alpha_j f_j(\cdot)\right),$$

otherwise, according to Lemma 7.1.7,

$$\mathbf{W}_{\delta'} \circ \mathbf{T}(f_i(\cdot)) = \mathbf{T} \circ \mathbf{L} \circ \mathbf{R}_{\delta'}(f_i(\cdot)).$$

Consequently the following equation holds:

$$\mathbf{T} \circ \mathbf{L} \circ \mathbf{R}_{\delta'}(f_i(\cdot)) = \mathbf{T}\left(\sum_{j=1}^{n} \alpha_j f_j(\cdot)\right).$$

Since $\mathbf{T}$ is a bijection,

$$\mathbf{L} \circ \mathbf{R}_{\delta'}(f_i(\cdot)) = \sum_{j=1}^{n} \alpha_j f_j(\cdot).$$

According to Lemma 7.1.8, since $\mathbf{R}_{\delta'}(f_i(\cdot))$ is a convex combination of δ_0 and $\mathbf{L} \circ \mathbf{R}_{\delta'}(f_i(\cdot))$, it follows that $\mathbf{R}_{\delta'}(f_i(\cdot)) \in \mathrm{co}\{\delta_0(\cdot), f_1(\cdot), \ldots, f_n(\cdot)\}$. $\square$

An important consequence of Theorem 7.1.13 is that if a polytope in $\mathcal{ME}^{\delta}_{\lambda_1,\lambda_2,\lambda_3}$

is $\mathbf{R}_t$-invariant, then the corresponding polytope in $\mathbb{R}^2$ is $\mathbf{W}_t$-invariant and vice versa.

The next goal is the classification of the set of absolutely continuous probability distributions according to the minimal order, at which a distribution can be represented by an upper triangular PH generator matrix.

7.2 Classification of PH distributions

In this section a recursive method is shown to find PH distributions with 3 distinct real poles that can be represented by matrices of 1,2,3 or more dimensions. First the mapping between $\mathcal{ME}^{\delta}_{\lambda_1,\lambda_2,\lambda_3}$ and $\mathbb{R}^2$ is illustrated. It is shown where the PH distributions are mapped and how the $\mathbf{W}_t$ and $\hat{\mathbf{\Theta}}$ operators work on the mapped points. Then a method is shown how to identify the set of functions that can be represented in n dimensions but can not be represented in $n+1$ dimensions. The methodology is demonstrated up to 3 dimensions, the investigation of higher dimensions is out of scope in this dissertation.

Figure 7.1 shows the set of $\mathcal{ME}_{\lambda_1,\lambda_2,\lambda_3}$ distributions and the main structure of the $\hat{\mathbf{\Theta}}$ vector field where some of the trajectories, which are in fact the $\mathbf{W}_t(\eta_1,\eta_2)$ orbits, of the vector field are plotted. The starting point of a $\mathbf{W}_t$-orbit represents a Phase-type distribution. The direction of the $\hat{\mathbf{\Theta}}$ corresponding to (η_1,η_2) is the tangential of the $\mathbf{W}_t$-orbit at this point.

According to [80], for each PH representation there is a corresponding $\mathbf{W}_t$-invariant polytope in the vector field. Visually, the $\mathbf{W}_t$-invariance means that a $\mathbf{W}_t(\eta_1,\eta_2)$-orbit starting inside the polytope remains inside the polytope for all t.

All $\mathbf{W}_t$-orbits (except for $\eta_1 = 0$) converge to $(1,0)$ since the distribution

$$f(u) = \eta_1\lambda_1 e^{-\lambda_1 u} + \eta_2\lambda_2 e^{-\lambda_2 u} + \eta_3\lambda_3 e^{-\lambda_3 u}, \quad \lambda_3 > \lambda_2 > \lambda_1 > 0$$

($\eta_1 > 0$) has exponential asymptotic with rate parameter λ_1. As $t > 0$ grows the absolutely continuous part of the distribution $\mathbf{R}_t(f(\cdot))$ becomes dominated by the $\lambda_1 e^{-\lambda_1 t}$ distribution and consequently the $\mathbf{T} \circ \mathbf{L}(\mathbf{R}_t(f(\cdot)))$ vector has increasingly dominant first coordinate.

Figure 7.1 also shows many $\mathbf{W}_t$-orbits asymptotically approaching the $\eta_1 +$

$\eta_2 = 1$ line. The reason for this is that the $e^{-\lambda_3 t}$ component in the absolutely continuous part of $\mathbf{R}_t(f(\cdot))$ vanishes faster therefore the other two components become dominant over the third component.

There are two curves shown by dashed lines in Figure 7.1 indicating the border of the convex set corresponding to the $\mathcal{ME}_{\lambda_1,\lambda_2,\lambda_3}$ distributions. One border is the line corresponding to the $f(0) = 0$ distributions. This is a one dimensional affine plane since the (η_1, η_2) points of this curve satisfy the following linear equation:

$$\eta_1\lambda_1 + \eta_2\lambda_2 + (1 - \eta_1 - \eta_2)\lambda_3 = 0. \tag{7.6}$$

The other curve contains points corresponding to distributions for which $f(t) = 0$ for a certain $t > 0$. This curve is a solution of a parametric equation on η_1 and η_2:

$$\eta_1\lambda_1 e^{-\lambda_1 \cdot t} + \eta_2\lambda_2 e^{-\lambda_2 \cdot t} + (1 - \eta_1 - \eta_2)\lambda_3 e^{-\lambda_3 \cdot t} = 0.$$

Note that this curve is also developed in Theorem 1 in [86].

The purpose of the following investigations is to find minimal upper triangular phase-type (TPH) representation with order 1, 2 and 3. The order of the minimal TPH representation is the triangular order. The triangular order is sometimes higher than the order.

The sets in $\mathbb{R}^2$ corresponding to the sets of absolutely continuous Phase-type distributions in $\mathcal{ME}_{\lambda_1,\lambda_2,\lambda_3}$ are constructed with upper triangular PH representation of order $1, 2$ and 3. Theorem 6.2 in [82] provides the basis for this analysis:

Theorem 7.2.1. *A triangular PH-distribution μ is of triangular order $k \geq 1$ if and only if, for some $\epsilon > 0$, $\mu + \epsilon\Gamma\mu$ is of triangular order $k - 1$.*

A formal definition for this relation is established by introducing the following notion of "look at".

Definition 7.2.2. *The distribution $g(\cdot) \in \mathcal{ME}_{\lambda_1,\lambda_2,\lambda_3}$, "looks at" the distribution $f(\cdot) \in \mathcal{ME}_{\lambda_1,\lambda_2,\lambda_3}$ if there exists a $c > 0$ ($c \in \mathbb{R}$), such that*

$$\hat{\mathbf{\Theta}} \circ \mathbf{T}(g(\cdot))) = c\left(\mathbf{T}(f(\cdot)) - \mathbf{T}(g(\cdot))\right).$$

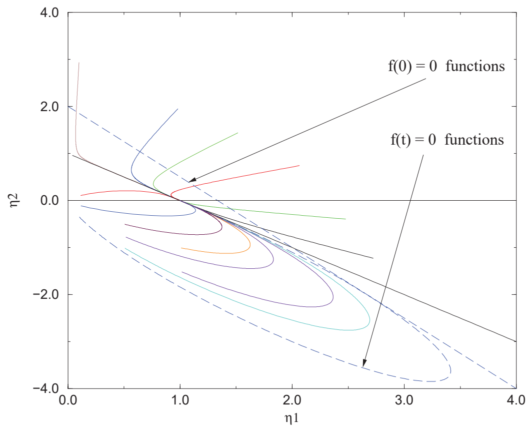

Figure 7.1: The structure of the vector field

Remark 7.2.3. *The constant in Definition 7.2.2 is defined to be positive for a* $\mathbf{T}(g(\cdot))$ *vertex of a convex polytope in order to point inward to the polytope. Nevertheless, the sign of the constant is not always considered in the intermediate calculations. If a negative constant occurs in the later equations, then this yields to pseudo-solutions. These will be sorted out in the last step, when a particular representation is found.*

First, distributions that can be represented in 1 dimension are found. In fact, these are the exponential distributions with parameter λ_1, λ_2 or λ_3. For this the set S_1 is defined.

Definition 7.2.4. *Let* $S_1 \subset \mathbb{R}^2$ *be as follows:*

$$S_1 = \{(0,0), (0,1), (1,0)\}.$$

Indeed, the vectors in S_1 correspond to the desired exponential distributions, since

$$(0,0) = \mathbf{T}\left((0;0,0,1)\right):\ \lambda_3 e^{-\lambda_3 \cdot t},$$

$$(0,1) = \mathbf{T}\left((0;0,1,0)\right):\ \lambda_2 e^{-\lambda_2 \cdot t},$$

$$(1,0) = \mathbf{T}\left((0;1,0,0)\right):\ \lambda_1 e^{-\lambda_1 \cdot t}.$$

Then S_2 can be defined formally as the set of those points "looking at" a point in S_1. The following definition of S_n shows how to distinguish the different sets of functions that have order-n upper triangular PH representations using the concept of Definition 7.2.2.

Definition 7.2.5. *S_n consists of points "looking at" points in S_{n-1}, but themselves are outside of S_{n-1}. Formally*

$$S_n = \{\overrightarrow{\eta}_n \in \mathbb{R}^2 : \overrightarrow{\eta}_n \notin S_{n-1}, \exists \overrightarrow{\eta}_{n-1} \in S_{n-1}, \overrightarrow{\eta}_n \text{“\textit{looks at}”} \overrightarrow{\eta}_{n-1}\}.$$

Based on Theorem 7.2.1, the set of distributions with triangular order n is S_n. The set S_2 is the union of three sets.

1. $\mathrm{co}\{(0,0),(0,\frac{\lambda_3}{\lambda_3-\lambda_2})\}$: $f(t) = \eta_2\lambda_2 e^{-\lambda_2\cdot t} + \eta_3\lambda_3 e^{-\lambda_3\cdot t}$, $\eta_2+\eta_3 = 1$,
2. $\mathrm{co}\{(0,0),(\frac{\lambda_3}{\lambda_3-\lambda_1},0)\}$: $f(t) = \eta_1\lambda_1 e^{-\lambda_1\cdot t} + \eta_3\lambda_3 e^{-\lambda_3\cdot t}$, $\eta_1+\eta_3 = 1$,
3. $\mathrm{co}\{(0,1),(\frac{\lambda_2}{\lambda_2-\lambda_1},\frac{-\lambda_1}{\lambda_2-\lambda_1})\}$: $f(t) = \eta_1\lambda_1 e^{-\lambda_1\cdot t} + \eta_2\lambda_2 e^{-\lambda_2\cdot t}$, $\eta_1+\eta_2 = 1$.

The set S_3 consists of distributions $\mathbf{T}(f_3(\cdot)) = (\eta_{3,1},\eta_{3,2})$ with the property that the $\hat{\mathbf{\Theta}}(\eta_{3,1},\eta_{3,2})$ vector points from $(\eta_{3,1},\eta_{3,2})$ towards a point $\mathbf{T}(f_2(\cdot)) = (\eta_{2,1},\eta_{2,2}) \in S_2$.

The $\mathcal{ME}_{\lambda_1,\lambda_2,\lambda_3}$ set can be divided into four main subsets according to the structure of the $\mathbf{W}_t$-orbits as it is shown in Figure 7.2:

$$C_1 \cup C_2 \cup C_3 \cup C_4 = \{\mathbf{T}(f(\cdot)) \ :\ f(\cdot) \in \mathcal{ME}_{\lambda_1,\lambda_2,\lambda_3}\},$$

where

$$C_1 = \{(\eta_1, \eta_2) : \eta_1 \geq 0, \eta_2 \geq 0, 1 - \eta_1 \leq \eta_2\} \cap \{\mathbf{T}(f(\cdot)) \ : \ f(\cdot) \in \mathcal{ME}_{\lambda_1,\lambda_2,\lambda_3}\},$$

$$C_2 = \{(\eta_1, \eta_2) : \eta_1 \geq 0, \eta_2 \geq 0, 1 - \eta_1 \geq \eta_2\} \cap \{\mathbf{T}(f(\cdot)) \ : \ f(\cdot) \in \mathcal{ME}_{\lambda_1,\lambda_2,\lambda_3}\},$$

$$C_3 = \{(\eta_1, \eta_2) : \eta_1 \geq 0, \eta_2 \leq 0, 1 - \eta_1 \leq \eta_2\} \cap \{\mathbf{T}(f(\cdot)) \ : \ f(\cdot) \in \mathcal{ME}_{\lambda_1,\lambda_2,\lambda_3}\},$$

$$C_4 = \{(\eta_1, \eta_2) : \eta_1 \geq 0, \eta_2 \leq 0, 1 - \eta_1 \geq \eta_2\} \cap \{\mathbf{T}(f(\cdot)) \ : \ f(\cdot) \in \mathcal{ME}_{\lambda_1,\lambda_2,\lambda_3}\}.$$

The $\mathbf{W}_t$-orbits starting from the points in C_1 approach the $\eta_1 + \eta_2 = 1$ line from above, while starting in C_2 the $\mathbf{W}_t$-orbits does the same from below. In both cases, the orbits stay above the horizontal axis. The trajectories starting in C_3 approach the $\eta_1 + \eta_2 = 1$ line from above, but the trajectories remain below the horizontal axis in this case. The structure of the trajectories starting in C_4 is more complex. It is bounded by the $f(t) = 0$ curve from below and 3 straight lines from above:

$$\eta_2 = 0$$

$$\eta_1 + \eta_2 = 1$$

$$f(0) = 0$$

Here, the distributions in C_1, C_2 and C_3 are part of S_3 since any of their points "looks at" a point in S_2, particularly on the one dimensional affine plane of $\eta_1 + \eta_2 = 1$.

Since not all points of C_4 "look at" a point in S_2, the set is determined whose points "look at" an extremal point of S_2. This extremal point is the intersection of the two lines given by $\eta_1 + \eta_2 = 1$ and $f(0) = 0$:

$$\left(\frac{\lambda_2}{\lambda_2 - \lambda_1}, \frac{-\lambda_1}{\lambda_2 - \lambda_1}\right) \in S_2.$$

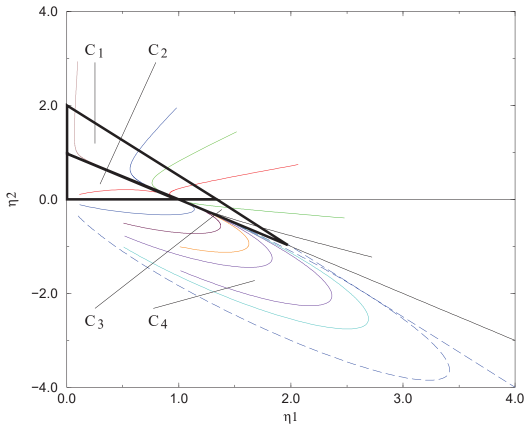

Figure 7.2: The C_i sets

The border of S_3 which splits the set C_4 into two, is the following.

Definition 7.2.6.

$$L_3 = \Big\{ (\eta_1, \eta_2) \in \mathbb{R}^2 \ : \ g(\cdot) \in \mathcal{ME}_{\lambda_1,\lambda_2,\lambda_3},\ (\eta_1, \eta_2) = \mathbf{T}(g(\cdot)),$$

$$\hat{\mathbf{\Theta}}((\eta_1, \eta_2)) = c\left((\eta_1, \eta_2) - \left(\frac{\lambda_2}{\lambda_2 - \lambda_1}, \frac{-\lambda_1}{\lambda_2 - \lambda_1}\right)\right) \Big\}.$$

L_3 is the set of points “looking at” the extremal point of S_2.

Theorem 7.2.7. *The set L_3 is a segment of a 1 dimensional affine plane given by*

$$(\lambda_3 - \lambda_1)\lambda_1\eta_1 + (\lambda_3 - \lambda_2)\lambda_2\eta_2 = 0.$$

Proof. The theorem is the special case of a more general theorem stated and proved later in Section 7.3. □

Now all information is available to obtain S_3.

Theorem 7.2.8. *S_3 is the triangle bordered by the following lines:*

$$\eta_1 = 0$$

$$f(0) = 0$$

$$L_3$$

Proof. All vertices of the triangle "look at" a point in S_2 so S_3 contains the triangle. The sets C_1, C_2 and C_3 are part of the triangle so S_3 contains C_1, C_2 and C_3. The question is that which part of C_4 "looks at" a point in S_2 and which does not. L_3 is defined as the set of points "looking at" the extremal point of S_2.

With the current parametrisation the signed curvature of a $\mathbf{W}_t$-orbit is positive if the tangent vector rotates anti-clockwise and negative if it rotates clockwise direction.

It is easy to see that if the signed curvature of the $\mathbf{W}_t$-orbits is positive for all starting points in this region then the points above L_3 "look at" a point in S_2 and none of the points below L_3 "look at" a point in S_2. In other words, if the signed curvature is positive then L_3 is the lower border of S_3.

Starting from (η_1, η_2) the $\mathbf{W}_t$-orbit is described as

$$(x(t), y(t)) = \left(\frac{\eta_1 e^{-\lambda_1 t}}{\sum_{i=1}^{3} \eta_i e^{-\lambda_i t}}, \frac{\eta_2 e^{-\lambda_2 t}}{\sum_{i=1}^{3} \eta_i e^{-\lambda_i t}} \right)$$

If x', x'', y', y'' denote the first and second derivatives at $t = 0$ of $x(t)$ and

$y(t)$, respectively then the signed curvature of the $\mathbf{W}_t$-orbit at the starting point is defined as

$$\kappa = \frac{x'y'' - y'x''}{(x'^2 + y'^2)^{3/2}}$$

After evaluating the numerator of κ, taking the derivatives at $t = 0$ (every point can be the starting point) and simplifying the expression one gets

$$\kappa = \frac{\eta_1\eta_2\eta_3(\lambda_2 - \lambda_1)(\lambda_1 - \lambda_3)(\lambda_3 - \lambda_2)}{(x'^2 + y'^2)^{3/2}}.$$

Using the facts that $\lambda_3 > \lambda_2 > \lambda_1$ and in C_4 $\eta_1 > 0$, $\eta_2 < 0$ and $\eta_3 > 0$, it can be deduced that $\kappa > 0$ and the proof is completed. □

Figure 7.2 illustrates that the signed curvature is positive for the orbits starting in C_4.

The above tedious construction of S_2 and in particular S_3 enables one to illustrate the sequential construction of the subsets whose corresponding distributions have upper triangular PH representation of certain order. The idea of sequential construction has been proposed in [82].

The construction presented here is based on the following idea. The distributions, which "look at" the set of distributions possessing upper triangular representations of minimal order $k - 1$, have upper triangular representations of minimal order k . According to this, the points of S_3 "look at" S_2 whose points "look at" S_1 as it was shown above.

This method can be used for higher dimensions as well. If L_n $(n > 3)$ is the set of points "looking at" the extremal points of S_{n-1} and L_n is computed then S_n is bordered by L_n and L_{n-1}. However, finding the extremal points of S_n for $n > 3$ is not trivial since S_n is not a polytope in this case. The method for finding the sets S_n $(n > 3)$ is not detailed here, it can be found in [101].

Figure 7.3 shows the sets of distributions with minimal triangular PH representations in $3, 4, 5$ and 6 dimensions. The remaining part of the $\mathcal{ME}_{\lambda_1,\lambda_2,\lambda_3}$ set contains distributions that do not have triangular PH representations up to order 6.

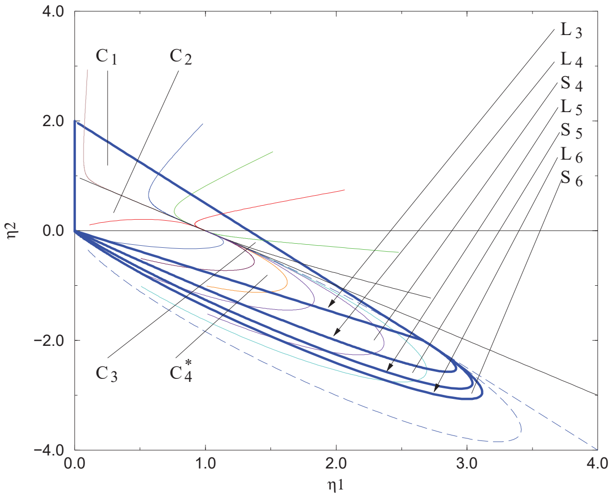

Figure 7.3: The borders of the subsets

7.3 Generalisation to m poles

Some statements can be generalised to functions with arbitrary number of distinct real poles in their Laplace transform. The form of these functions can be written as

$$f(t) = \sum_{i=1}^{m} \eta_i \lambda_i e^{-\lambda_i t},$$

for which

$$\sum_{i=1}^{m} \eta_i = 1, \quad \lambda_m > \ldots > \lambda_1 > 0$$

Some of the definitions are repeated in a more general setting.

Definition 7.3.1. *Let $f(\cdot) \in \mathcal{ME}_{\lambda_1,\ldots,\lambda_m}$ be an absolutely continuous function, which can be expressed in the canonical basis as*

$$f(\cdot) = (0; \eta_1, \ldots, \eta_m)$$

and $\eta_1 + \ldots + \eta_m = 1$, that is $\mathbf{L}(f(\cdot)) = f(\cdot)$. Define the operator $\mathbf{T}$ from the above affine plane of absolutely continuous functions to $\mathbb{R}^{m-1}$ as

$$\mathbf{T}\left((0; \eta_1, \ldots, \eta_m)\right) = (\eta_1, \ldots, \eta_{m-1}).$$

Definition 7.3.2. *Let the $\mathbf{W}_t : \mathbb{R}^{m-1} \to \mathbb{R}^{m-1}$ operator ($t \in \mathbb{R}$, $t > 0$) be defined as*

$$\mathbf{W}_t(\eta_1, \ldots, \eta_{m-1}) = \left(\frac{\eta_1 e^{-\lambda_1 \cdot t}}{\sum_{i=1}^{m-1} \eta_i e^{-\lambda_i \cdot t} + (1 - \sum_{i=1}^{m-1} \eta_i) e^{-\lambda_m \cdot t}}, \ldots, \right.$$

$$\left. \frac{\eta_{m-1} e^{-\lambda_{m-1} \cdot t}}{\sum_{i=1}^{m-1} \eta_i e^{-\lambda_i \cdot t} + (1 - \sum_{i=1}^{m-1} \eta_i) e^{-\lambda_m \cdot t}} \right).$$

Let the $\hat{\mathbf{\Theta}} : \mathbb{R}^{m-1} \to \mathbb{R}^{m-1}$ operator be defined as

$$\hat{\mathbf{\Theta}}(\eta_1, \ldots, \eta_{m-1}) = \lim_{t \to 0} \frac{\mathbf{W}_t(\eta_1, \ldots, \eta_{m-1}) - \mathbf{W}_0(\eta_1, \ldots, \eta_{m-1})}{t}$$

Definition 7.3.3. *The distribution $g(\cdot) \in \mathcal{ME}_{\lambda_1,\ldots,\lambda_m}$ where $g(\cdot) \geq 0$ and $\int_0^\infty g(t)dt = 1$ "looks at" the subset $D \subset \mathcal{ME}_{\lambda_1,\ldots,\lambda_m}$ if $\exists f(\cdot) \in D$ and $\exists c > 0$ ($c \in \mathbb{R}$), such that*

$$\hat{\mathbf{\Theta}} \circ \mathbf{T}(g(\cdot))) = c\left(\mathbf{T}(f(\cdot)) - \mathbf{T}(g(\cdot))\right).$$

Using the above definitions, the following theorem holds:

Theorem 7.3.4. *Introduce the* $m-2$ *dimensional affine planes* $D_1, D_2 \subset \mathbb{R}^N$ *as*

$$D_1 = \{(\eta_1, \dots, \eta_{m-1}) : \sum_{i=1}^{m-1} \eta_i = 1\}$$

$$D_2 = \{(\eta_1, \dots, \eta_{m-1}) : \sum_{i=1}^{m-1} \lambda_i \eta_i = 0\}$$

where $N \geq m-1$*,* $m \geq 3$*,* $\lambda_i \in \mathbb{R}$ $i = 1, \dots, m-1$ *are constant. The functions that "look at" the* $m-3$ *dimensional affine plane* $D_1 \cap D_2$*, form an* $m-2$ *dimensional affine plane given by the following equation:*

$$\sum_{i=1}^{m-1} (\lambda_i - \lambda_m) c_i \eta_i = 0.$$

Proof. The aim is to determine the set of distributions $g(\cdot)$ that "look at" $D_1 \cap D_2$. Let's introduce η_i, η_i', $i = 1, \dots, m-1$ where $\mathbf{T}(f(\cdot)) = (\eta_1, \dots, \eta_{m-1})$, $f(\cdot) \in D_1 \cap D_2$ and $\mathbf{T}(g(\cdot)) = (\eta_1', \dots, \eta_{m-1}')$. Then the following equation holds,

$$\left[\hat{\mathbf{\Theta}} \circ \mathbf{T}(g(\cdot))\right]_i = c(\eta_i' - \eta_i),$$

where

$$\left[\hat{\mathbf{\Theta}} \circ \mathbf{T}(g(\cdot))\right]_i = -\lambda_i \eta_i' + \eta_i' \sum_{j=1}^{m} \lambda_j \eta_j'.$$

Thus, η_i can be expressed by λ_is and η_i's.

$$\eta_i = \eta_i' + \frac{1}{c} \lambda_i \eta_i' - \frac{1}{c} \eta_i' \sum_{j=1}^{m} \lambda_j \eta_j'$$

Note that $\eta_m' = 1 - \sum_{j=1}^{m-1} \eta_j'$. Since η_is are components of a distribution in

$D_1 \cap D_2$, they satisfy the following equations,

$$\sum_{i=1}^{m-1} \eta_i = 1$$

and

$$\sum_{i=1}^{m-1} \lambda_i \eta_i = 0.$$

By replacing η_i to the above equations and eliminating c one gets

$$\left(1 - \sum_{i=1}^{m-1} \eta_i'\right) \left(\lambda_m \sum_{i=1}^{m-1} \lambda_i \eta_i' - \sum_{i=1}^{m-1} \lambda_i^2 \eta_i'\right) = 0$$

In fact, this equation defines two affine planes. For all the points of D_1 the expression inside the first parentheses is 0. The other affine plane is defined by the expression inside the second parentheses when it is set to 0. This is a linear equation for η_i's, that is, this affine plane contains the origin. □

Specifying Theorem 7.3.4 with $m = 3$ one gets Theorem 7.2.7 as indicated before.

7.4 Finding a representation

A method is shown in this section for finding upper triangular representations to distributions within invariant polytopes. The method is based on Lemma 3.1 in [80].

Theorem 7.4.1. *1. Let $f_i(\cdot) \in \mathcal{ME}_{\lambda_1,\lambda_2,\lambda_3}$, $(i = 1, \ldots, n)$ be distributions where*

$$\mathrm{co}\{f_1(\cdot), \ldots, f_n(\cdot)\}$$

is $\mathbf{R}_t$-invariant. Then there exists an $n \times n$ generator matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$, *for which*

$$\mathbf{\Gamma}(f_i) = \sum_{j=1}^{n} \mathbf{M}_{ij} f_j$$

2. *Introduce* $\mathbf{S}, \mathbf{G} \in \mathbb{R}^{n\times 3}$ *matrices containing the coordinates of distributions* $f_i(\cdot)$ *and* $\mathbf{\Gamma}(f_i(\cdot))$ *in the*

$$\lambda_1 e^{-\lambda_1 t}, \lambda_2 e^{-\lambda_2 t}, \lambda_3 e^{-\lambda_3 t}$$

canonical basis, that is,

$$f_i(t) = \sum_{k=1}^{3} \mathbf{S}_{ik} \lambda_k e^{-\lambda_k t}, \text{ where } i = 1, \ldots, n$$

and

$$\mathbf{\Gamma}(f_i(t)) = \sum_{k=1}^{3} \mathbf{G}_{ik} \lambda_k e^{-\lambda_k t}, \text{ where } i = 1, \ldots, n.$$

Then the following holds,

$$\mathbf{G} = \mathbf{M}\mathbf{S}.$$

Proof. 1. The proof of the first part can be found in [80] as part of the proof of the "Invariant polytope lemma".

2. The ith row of $\mathbf{G}$ can be written as

$$\mathbf{\Gamma}(f_i(t)) = \sum_{j=1}^{n} \mathbf{M}_{ij} f_j = \sum_{j=1}^{n} \mathbf{M}_{ij} \left(\sum_{k=1}^{3} \mathbf{S}_{jk} e^{-\lambda_k t} \right) =$$

$$\sum_{k=1}^{3} e^{-\lambda_k t} \left(\sum_{j=1}^{n} \mathbf{M}_{ij} \mathbf{S}_{jk} \right) = \sum_{k=1}^{3} [\mathbf{M}\mathbf{S}]_{ik} e^{-\lambda_k t},$$

that is, $\mathbf{G}_{ik} = [\mathbf{MS}]_{ik}$ and the Theorem follows.

□

Corollary 7.4.2. *If the polytope in Theorem 7.4.1 has 3 extreme points, that is, $n = 3$ and $\mathbf{S} \in \mathrm{R}^{3\times 3}$ is invertible, then*

$$\mathbf{M} = \mathbf{GS}^{-1}.$$

If the $\mathbf{S} \in \mathbb{R}^{3\times 3}$ matrix is not invertible or if one intends to find a PH representation of order more than 3 (in which case $\mathbf{S}$ is not a square matrix therefore $\mathbf{S} \in \mathbb{R}^{n\times 3}$ cannot be invertible) then the result of Corollary 7.4.2 can be generalised to the case of non-invertible $\mathbf{S}$, where the generalised inverse (pseudoinverse) of $\mathbf{S}$ is used. For details regarding the generalised inverse refer to [89].

Corollary 7.4.3. *If $\mathbf{S} \in \mathbb{R}^{3\times 3}$ is not invertible, or if the polytope in Theorem 7.4.1 has more than 3 extreme points, that is, $n > 3$, then there exists a matrix $\mathbf{K} \in \mathbb{R}^{n\times n}$ such that*

$$\mathbf{M} = \mathbf{GS}^{\#} + \mathbf{K},$$

where $\mathbf{KSS}^{\#} = \mathbf{0}$ and $\mathbf{S}^{\#}$ denotes the pseudoinverse of $\mathbf{S}$.

Proof. Since $\mathbf{SS}^{\#}$ is a projection, it decomposes the vector space $\mathbb{R}^n$ into a direct sum of two subspaces. One is $\mathrm{Ker}(\mathbf{SS}^{\#})$ the kernel of $\mathbf{SS}^{\#}$ and the other one is $\mathrm{Im}(\mathbf{SS}^{\#})$ the image of $\mathbf{SS}^{\#}$, that is,

$$\mathbb{R}^n = \mathrm{Ker}(\mathbf{SS}^{\#}) + \mathrm{Im}(\mathbf{SS}^{\#}).$$

where $\mathbf{SS}^{\#}$ is identical on $\mathrm{Im}(\mathbf{SS}^{\#})$.

Denote the rows of $\mathbf{M}$ by $\overrightarrow{m}_i$:

$$\mathbf{M} = \begin{bmatrix} \overrightarrow{m}_1 \\ \vdots \\ \overrightarrow{m}_n \end{bmatrix}.$$

Let $\overrightarrow{k}_i \in \mathrm{Ker}(\mathbf{SS}^{\#})$ be such that $\overrightarrow{m}_i - \overrightarrow{k}_i \in \mathrm{Im}(\mathbf{SS}^{\#})$. Denote $\mathbf{K}$ the matrix composed of $\overrightarrow{k}_i$ vectors:

$$\mathbf{K} = \begin{bmatrix} \overrightarrow{k}_1 \\ \vdots \\ \overrightarrow{k}_n \end{bmatrix}.$$

From Theorem 7.4.1 one has

$$\mathbf{G} = \mathbf{MS}.$$

Then

$$\mathbf{GS}^{\#} = \mathbf{MSS}^{\#} = (\mathbf{M} - \mathbf{K})\, \mathbf{SS}^{\#} = (\mathbf{M} - \mathbf{K}).$$

It was used in the above steps that $\mathbf{K}$ is composed of kernel vectors and $(\mathbf{M} - \mathbf{K})$ is composed of image vectors. $\square$

In practice, there is a certain degree of freedom in choosing the $\overrightarrow{k}_i$ kernel vector in $\mathbf{K}$ when the ith row of $\mathbf{M}$ is calculated. It should be done in such a way that $\mathbf{M}$ would become a proper generator matrix. This also supports the fact that a PH representation is not always unique.

7.5 Numerical example

First the concept of "points inward" is demonstrated with the following simple example. Let $\lambda_1 = 0.5$, $\lambda_2 = 1$ and $\lambda_3 = 2$ and consider the following three points in $\mathbb{R}^2$:

$$\overrightarrow{\eta}_1 = (0, 0),\ \overrightarrow{\eta}_2 = (2.667, -2),\ \overrightarrow{\eta}_3 = (0, 2).$$

The $\mathrm{co}\{\overrightarrow{\eta}_1, \overrightarrow{\eta}_2, \overrightarrow{\eta}_3\}$ triangle is $\mathbf{W}_t$-invariant because $\hat{\mathbf{\Theta}}(\overrightarrow{\eta}_i)$ points inward to the triangle for $i = 1, 2, 3$. In order to see this, $\hat{\mathbf{\Theta}}(\overrightarrow{\eta}_i)$ can be calculated using

Definition 7.1.6. First, the $\mathbf{W}_t$ functions are calculated:

$$\mathbf{W}_t(\overrightarrow{\eta}_1) = (0,0),$$
$$\mathbf{W}_t(\overrightarrow{\eta}_2) = \left(\frac{2.667e^{-0.5t}}{2.667e^{-0.5t}-2e^{-t}+0.333e^{-2t}}, \frac{-2e^{-t}}{2.667e^{-0.5t}-2e^{-t}+0.333e^{-2t}}\right),$$
$$\mathbf{W}_t(\overrightarrow{\eta}_3) = \left(0, \frac{2e^{-t}}{2e^{-t}-e^{-2t}}\right).$$

Next the $\hat{\mathbf{\Theta}}$ vectors can be calculated

$$\hat{\mathbf{\Theta}}((0,0)) = (0,0), \quad \hat{\mathbf{\Theta}}((2.667,-2)) = (-1.333,2), \quad \hat{\mathbf{\Theta}}((0,2)) = (0,-2).$$

It is obvious for the $\hat{\mathbf{\Theta}}(\overrightarrow{\eta}_1)$ vector that it points inward to the triangle because this is the $\overrightarrow{0}$ vector. $\hat{\mathbf{\Theta}}(\overrightarrow{\eta}_3)$ points inward to the triangle because this vector points from $\overrightarrow{\eta}_3$ towards $\overrightarrow{\eta}_1$:

$$\overrightarrow{\eta}_1 - \overrightarrow{\eta}_3 = (0,0) - (0,2) = (0,-2) = \hat{\mathbf{\Theta}}((0,2)).$$

Finally, $\hat{\mathbf{\Theta}}(\overrightarrow{\eta}_2)$ also points inward to the triangle, because this vector points from $\overrightarrow{\eta}_2$ towards $\overrightarrow{\eta}_3$:

$$\overrightarrow{\eta}_3 - \overrightarrow{\eta}_2 = (0,2) - (2.667,-2) = (-2.667,4) = 2\hat{\mathbf{\Theta}}((2.667,-2)).$$

Note that η_1, η_2 and η_3 are the vertices of S_3 with the given λ_i values.

The second example demonstrates that the triangular order is generally higher than the order. An order 3 PH generator matrix containing a cycle with eigenvalues $-\lambda_1, -\lambda_2, -\lambda_3$ is shown in Equation 7.7.

$$\mathbf{P} = \begin{bmatrix} -2.0421 & 0.1982 & 0 \\ 0 & -0.6365 & 0.6365 \\ 0.5363 & 0 & -0.8214 \end{bmatrix} \tag{7.7}$$

The invariant triangle corresponding to this representation is different from the triangles above in the sense that the vector field is non-zero in all the 3 vertices of the triangle. The 1st vertex “looks at” the 2nd one. The 2nd “looks at” the 3rd vertex. Moreover, the vector at the 3rd vertex points towards the 1st vertex.

Since this construction is fundamentally different it is not trivial how the set of PH distributions possessing such representations is related to S_3.

The vertices of $\mathbf{P}$ are $([0,0,1],P)$, $([0,1,0],P)$ and $([1,0,0],P)$:

$$\begin{aligned}\overrightarrow{\nu}_1 &= (0.5200, \quad 0.9349),\\ \overrightarrow{\nu}_2 &= (2.4247, \quad -1.6371),\\ \overrightarrow{\nu}_3 &= (0.3116, \quad -0.3113).\end{aligned}$$

It is not difficult to see that $\overrightarrow{\nu}_1, \overrightarrow{\nu}_2 \in S_3$. However, $\overrightarrow{\nu}_3$ is not in S_3 because $(0.3116, -0.2337) \in L_3$ and the second coordinate of $\overrightarrow{\nu}_3$ is smaller than -0.2337. Consequently, there is no equivalent upper triangular PH representation of order 3 for $([1,0,0],\mathbf{P})$.

7.6 Conclusions

The structure of the Phase-type distributions whose Laplace transform have 3 distinct real poles is investigated. A recursive decomposition of the set of such distributions into subsets according to their minimal order upper triangular PH representations is provided. This is done by mapping the set of distributions into a 2 dimensional vector space. In order to use the invariant polytope approach, a parametric linear mapping and a corresponding vector field on this vector space is defined.

This analysis provides a basis for finding those functions with minimal triangular order higher than 3. Also a generalisation is given for finding n dimensional triangular PH representations in case of n distinct real poles. A method is shown to obtain the representation matrix of the functions inside an invariant polytope.

A possible generalisation of the results is the representation of PH distributions with more than 3 distinct real poles and the case of non-distinct real poles through the special order 3 case, which is already developed in [82]. Further generalisation can be the case of complex poles.

Summary

The objective of my research was to analyse and model the traffic behaviour in computer networks. The results presented in this dissertation are based on capturing the essential properties of the underlying network protocols on one hand and on the application and optimisation of Markovian models and matrix analytic methods on the other hand.

The first part of the dissertation focuses on traffic characteristics on a certain link and the behaviour of the communication protocols are modelled. First it is shown that congestion can propagate in TCP/IP networks in a natural way. It is explained how the feedback-based end-to-end protocol, TCP contributes to burst effects in the network and how the burst effect causes the propagation of congestion from one router to the other. Then traffic models are set up for file downloads where the average file size is small. In the first model parallel transfers sharing a link are investigated and formulae are derived for the link utilisation in deterministic case and for the number of parallel connections where the connection arrival and departure is random. The second model determines the download performance for a Web page when the objects are retrieved sequentially.

The second part of the dissertation contains results on solving a queuing problem with matrix analytic methods. The time-dependent moments of an infinite server queuing model is obtained exactly and it is illustrated how the solution can be used in modeling and engineering of a telecommunication server. Then a new formalism is introduced to investigate the structure of phase-type distributions. The distribution functions are mapped to a vector space where the phase-type distributions were classified based on complexity i.e. the size of their representation matrices. The statements are declared on 3 dimensions and some theorems are proved for n dimensions which can contribute to the solution of the generalised problem in the future.

Összefoglalás

Kutatómunkám célja az volt, hogy elemezzem és modellezzem a forgalom viselkedését számítógéphálózatokban. Az ismertetett eredmények egyrészt a hálózatban működő protokollok alaptulajdonságaira, másrészt Markov modellek és mátrix-analitikus módszerek alkalmazására és azok optimalizálására épülnek.

A disszertáció első részében a hálózat egy pontján mérhető forgalmi jellemzők leírására és a kommunikációs protokollok modellezésére helyeztem a hangsúlyt. Először megállapítottam, hogy a torlódások természetes módon terjednek a TCP/IP hálózatokban. Megmutattam, hogy a visszacsatoláson alapuló, a hálózat végpontjain működő TCP protokoll hogyan járul hozzá a ”burst”-ös csomagérkezésekhez és ezen keresztül a torlódás terjedéséhez egyik routertől a másikig. Majd olyan fájlletöltések forgalmát modelleztem, ahol az átlagos fájlméret kicsi. Az első modellben párhuzamos letöltéseket vizsgáltam, melyek ugyanazon a vonalon osztoznak, és levezettem egy formulát a vonal kihasználtságának jellemzésére determinisztikus esetben és a párhuzamos TCP kapcsolatok számának leírására abban az esetben, amikor az érkezés és a kiszolgálási idő véletlenszerű. A második modellben Web-oldalak letöltésének teljesítménymutatóit határoztam meg, ahol az oldalon levő objektumok egymás után töltődnek le.

A disszertáció második része egy sorbanállási probléma megoldását mutatja be mátrix analitikus módszerek segítségével. Egzakt megoldást mutattam be a sorhossz momentumainak időbeli változására egy olyan sorbanállási rendszerben, ahol végtelen számú kiszolgáló van, az érkezések és a kiszolgálás pedig mátrix analitikus függvényekkel adott, továbbá demonstráltam, hogyan alkalmazható közvetlenül a kapott eredmény távközlésben használt szerverek tervezésénél. Majd egy új formalizmust vezettem be a fázis-típusú eloszlások jellemzésére. Az eloszlásfüggvényeket egy vektortérre képeztem le, ahol bonyolultság szerint osztályoztam az eloszlásokat, vagyis aszerint, hogy milyen méretű mátrixokkal reprezentálhatók. Az állításokat három dimenzióra fogalmaztam meg, de bizonyos tételeket több dimenzióra is beláttam, ami az általános probléma megoldásához vezethet.

Bibliography

[1] A. Tanenbaum, ”Computer Networks,” *3rd edition (Prentice Hall, Englewood Cliffs)*, 1996.

[2] W. R. Stevens, ”TCP/IP Illustrated,” *Volume 1, Addison-Wesley*, 1994.

[3] R. Pastor-Storras and A. Vespignani, ”Evolution and Structure of the Internet,” *Cambridge U.P., Cambridge*, 2004.

[4] L. Kocarev and G. Vattay, ”Complex Dynamics in Communication Networks,” *Springer-Verlag*, Berlin, 2005.

[5] A. K. Erlang, ”Solution of some problems in the theory of probabilities of significance in automatic telephone exchanges,” *Elektrotkeknikeren*, 13, 5:13, 1917.

[6] J. Postel, ”Transmission Control Protocol,” *RFC 793*, September 1981.

[7] V. Jacobson, ”Congestion avoidance and control,” in *Proc. of SIGCOMM’88*, 314-329, August 1988.

[8] V. Jacobson, R. Braden and D. Borman, ”TCP Extensions for High Performance,” *RFC 1323*, May 1992.

[9] M. Mathis, J. Madavi, S. Floyd and A. Romanow, ”TCP Selective Acknowledgement Options,” *RFC 2018*, October 1996.

[10] J. Nagle ”Congestion Control in IP/TCP Internetworks,” *RFC 896*, January 1984.

[11] M. Allman, S. Floyd and C. Partridge, ”Increasing TCP’s Initial Window,” *RFC 2414*, September 1998.

[12] M. Allman, V. Paxson and W. Stevens, ”TCP Congestion Control,” *RFC 2581*, April 1999.

[13] S. Floyd, ”TCP and Explicit Congestion Notification,” *ACM Computer Communication Review*, Vol. 24, No. 5, October 1994.

[14] T. Lakshman and U. Madhow, ”The performance of TCP/IP for networks with high bandwidth-delay products and random loss,” *IEEE/ACM Transactions on Networking*, 5(3) 336-350, July 1997.

[15] T. J. Ott, J. H. B. Kemperman and M. Mathis, ”The Stationary Behavior of Ideal TCP Congestion Avoidance,” in *Proc. of IEEE INFOCOM’99*, (New York), 1999.

[16] M. Mathis, J. Semke and J. Mahdavi, ”The macroscopic behavior of the TCP congestion avoidance algorithm,” *Computer Communications Review*, 27(3), 1997.

[17] J. Padhye, V. Firoiu, D. Towsley and J. Krusoe, ”Modeling TCP throughput: A simple model and its empirical validation,” in *Proc. of SIGCOMM*, 303-314, 1998.

[18] E. Altman, ”A stochastic model of TCP/IP with stationary random losses,” *ACM Computer Communication Review*, 30(4), 2000.

[19] S. Ben Fredj, T. Bonald, A. Proutiere, G. Regnie and J. Roberts ”Statistical Bandwidth Sharing: A Study of Congestion at Flow Level,” in *Proc. of ACM SIGCOMM*, 111-122, 2001.

[20] D. P. Heyman, T. V. Lakshman and A. L. Neidhart, ”A new method for analyzing feedback-based protocols with applications to engineering web traffic over the internet,” in *Proc. of SIGMETRICS*, 24-38, (Seatle, WA, USA), 1997.

[21] L. Zhang, S. Shenker and D. D. Clark, ”Observations on the dynamics of a congestion control algorithm: The effects of two-way traffic,” in *Proc. of SIGCOMM*, 133-147, 1991.

[22] N. Cardwell, S. Savage and T. Anderson ”Modeling TCP latency,” in *Proc. of INFOCOM*, (3) 1742-1751, 2000.

[23] N. Cardwell, S. Savage and T. Anderson, ”Modeling the performance of short TCP connections,” *Technical Report*, Computer Science Department, Washington University, 1998.

[24] R. Chakravorty and I. Pratt, ”WWW Performance over GPRS,” in *Proc. of 4th IEEE International conference in Mobile and Wireless Communications Networks (IEEE MWCN 2002)*, (Stockholm, Sweden), September 2002.

[25] U. Fiedler, ”Provisioning Networks for Web Services,” *http://www.tik.ee.ethz.ch/ fiedler/provisioning.html*

[26] S. Floyd and V. Jacobson, ”Random Early Detection Gateways for Congestion Avoidance,” *ACM Transactions on Networking*, 1(4), 397-413, 1993.

[27] D. Lin and R. Morris, ”Dynamics of Random Early Detection,” in *Proc. of ACM SIGCOMM Computer Communication Review*, 27(4), 127-137, 1997.

[28] M. Christiansen, K. Jeffay, D. Ott and F.D. Smith, ”Tuning RED for Web Traffic,” in *Proc. of ACM SIGCOMM*, 139-150 2000.

[29] R. Albert and A.L. Barabási, ”Statistical Mechanics of Complex Networks,” *Rev. Mod. Phys.* **74**, 47, 2002.

[30] A. L. Barabási, R. Albert and H. Jeong, ”Scale-free characteristics of random networks: the topology of the World Wide Web,” *Physica A***281**, 69-77, 2000.

[31] M. Faloutsos, P. Faloutsos and C. Faloutsos, ”On power-law relationships of the Internet topology,” *Computer Communication Review*, 29, 251-263, 1999.

[32] P. Erdős and A. Rényi, ”On Random Graphs,” *Publ. Math.* (Debrecen), 6, 290, 1959.

[33] P. Echenique, J. Gomez-Gardenes and Y. Moreno, ”Dynamics of jamming transitions in complex networks,” *Europhys. Lett.* 71, 325-331, 2005.

[34] V. Paxson and S. Floyd, ”Wide-Area Traffic: The Failure of Poisson Modeling,” *IEEE/ACM Transactions on Networking*, **3**, 226, 1995.

[35] I. Csabai, ”1/f Noise in Computer Network Traffic,” *J. Phys A* **27**, L417-L421, 1994.

[36] T. Ohira and R. Sawatari, ”Phase transition in computer network traffic model,” *Phys. Rev E* **58**, 193-195, 1998.

[37] M. Takayasu, H. Takayasu and T. Sato, ”Critical behaviors and 1/f noise in information traffic,” *Physica A* **233**, 824-834, 1996.

[38] M.Takayasu, H.Takayasu, and K.Fukuda, ”Dynamic Phase Transition Observed in the Internet Traffic Flow,” *Physica A* **277**, 248-260, 2000.

[39] K. Fukuda, M. Takayasu and H. Takayasu, ”Spatial and temporal behavior of congestion in Internet traffic,” *Fractals* **7**, 23-31, 1999.

[40] K. Fukuda, H. Takayasu and M. Takayasu, ”Origin of Critical Behavior in Ethernet Traffic,” *Physica A* **287**, 289-301, 2000.

[41] A. Y. Tretyakov, H. Takayasu and M. Takayasu, ”Phase Transition in a computer network model,” *Physica A* **253**, 315, 1998.

[42] M.S. Taqqu, W. Willinger and R. Sherman, ”Proof of a fundamental result in self-similar traffic modelling,” *Computer Comm. Rev.* 27, 5-23, 1997.

[43] M. E. Crovella and A. Bestavros, ”Self-similarity in World Wide Web traffic. Evidence and possible causes,” *IEEE/ACM Trans. Network* **5**, 835, 1997.

[44] J. Yuan, Y. Ren and Xiuming Shan, ”Self-organized criticality in a computer network model,” *Phys. Rev. E* **61**, 1067, 1999.

[45] W. E. Leland, M. S. Taqq, W. Willinger and D. V. Wilson, ”On the Self-similar Nature of Ethernet Traffic,” in *Proc. of ACM SIGCOMM*, 183-193, (San Francisco, California), 1993.

[46] P. Bak, C. Tang and K. Weisenfeld, ”Self-Organized Criticality: An Explanation of 1/f Noise,” *Phys. Rev. Lett.* **59**, 381, 1987.

[47] A. Veres and M. Boda, ”The chaotic nature of TCP congestion control,” in *Proc. of IEEE Infocom'2000* 1715-1723, 2000.

[48] A. Veres, Zs. Kenesi, S. Molnár and G. Vattay, ”On the propagation of long-range dependence in the Internet,” in *Proc. of ACM Sigcomm'2000*, (Stockholm, Sweden), August 2000.

[49] G. Simon and I. Csabai, ”The analogies of highway and computer network traffic,” *Physica A* **307**, 516 - 526, 2002.

[50] K. Nagel and M. Paczuski, ”Emergent traffic jams,” *Phys. Rev. E* **51**, 2909, 1995.

[51] M. Bando, K. Hasebe, A. Nakayama, A. Shibata and Y. Sugiyama, ”Dynamical model of traffic congestion and numerical simulation,” *Phys. Rev. E* 51(2), 1035-1042, 1995.

[52] A. Schadschneider, ”Statistical physics of traffic flow,” *Physica A* **285**, 101, 2000.

[53] K. Nagel, ”High-speed microsimulations of traffic flow,” *PhD thesis*, 2005.

[54] T. Huisinga, R. Barlovic, W. Knospe, A. Schadschneider and M. Schrackenberg, ”A microscopic model for packet transport in the Internet,” *Physica A* **294**, 249, 2001.

[55] G. Mukherjee and S. S. Manna, ”Phase transition in a directed traffic flow network,” *Physical Review E*, vol. 71, Issue 6, id. 066108

[56] A. Arenas, A. Diaz-Guilera and R. Guimera, ”Communication in networks with hierarchical branching,” *Phys. Rev. Lett.* **86**, 3196, 2001.

[57] R. V. Sole and S. Valverde, ”Information transfer and phase transitions in a model of internet traffic,” *Physica A* **289**, 595, 2001.

[58] S. Valverde and R. V. Sole, ”Self-organized critical traffic in parallel computer networks,” *Physica A* **312**, 636, 2002.

[59] S. Tadaki, ”Long-term power-law fluctuation in Internet traffic,” *Journal of the Physical Society of Japan* Vol. 76, No. 4, 044001, 2007.

[60] W. Gong, Y. Liu, V. Misra, and D. Towsley, ”On the Tails of Web File Size Distributions,” in *Proc. of the 39th Annual Allerton Conference on Communication, Control and Computing*, October 2001.

[61] D. R. Figueiredo, B. Liu, V. Misra, and D. Towsley, ”On the Autocorrelation Structure of TCP Traffic,” *Computer Networks*, 40(3), 339-361, 2002.

[62] M. F. Neuts, ”Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach,” *Johns Hopkins University Press*, 1981.

[63] M. F. Neuts, ”Probability distributions of phase type,” *In Liber Amicorum Professor Emeritus H. Florin*, (University of Louvain, Belgium), 173-206, 1975.

[64] A. Feldmann and W. Whitt, ”Fitting mixtures of exponentials to long-tail distributions to analyze network performance models,” *Performance Evaluation*, 32(4), 245-279, 1998.

[65] A. Horváth and M. Telek, ”Approximating heavy tailed behavior with phase type distributions,” *In Advances in Matrix-Analytic Methods for Stochastic Models* 191-214, Notable Publications, July 2000.

[66] A. Horváth and M. Telek , ”PhFit: A general phase-type Fitting tool,” in *Proc. of Performance TOOLS 2002*, 82-91, April 2002.

[67] A. Horváth, G.I. Rozsa and M. Telek, ”A MAP fitting method to approximate real traffic behavior,” in *Proc. of 8th IFIP Workshop on Performance Modelling of ATM & IP Networks* 33, 1-12, (Illkey, UK) 2001.

[68] R. E. A. Khayari, R. Sadre and B. Haverkort ”Fitting World-Wide Web request traces with the EM-algorithm,” *Performance Evaluation*, 52(3), 175-191, 2003.

[69] A. Riska, V. Diev, and E. Smirni, ”An EM-based technique for approximating long-tailed data sets with PH distributions,” *Performance Evaluation*, 55(1-2), 147-164, 2004.

[70] G. Latouche and V. Ramaswami, ”Introduction to Matrix Analytic Methods in Stochastic Modeling,” *ASA-SIAM series on statistics and applied probability*, 1999.

[71] V. Ramaswami and M. F. Neuts, ”Some explicit formulas and computational methods for infinite-server queues with phase-type arrivals,” *J. Appl. Prob. 17*, 498-514, 1980.

[72] A. Pfening and M. Telek, ”Evaluation of task completion time of phase type work requirement,” in *Proc. of Relectronic'95*, (Budapest), 73-78, 1995.

[73] B. L. Nelson and M. R. Taaffe, ”The $Ph_t/Ph_t/\infty$ Queueing System: Part I — The Single Node,” in *Proc. of INFORMS JOC*, 2004.

[74] W. Fischer and K. Meier-Hellstern, ”The Markov Modulated Poisson process (MMPP) cookbook,” *Performance Evaluation* 18, 149-171, (1993)

[75] H. Heffes and D.M. Lucantoni, ”A Markov modulated characterisation of packetized voice and data traffic and related statistical multiplexer performance,” *IEEE Journal on Selected Areas in Communications* 4(6), 856-868, 1986.

[76] K.S. Meier-Hellstern, ”A fitting algorithm for Markov-modulated Poisson processes having two arrival rates,” *European Journal of Operational Research*, 377, 29-37, 1987.

[77] S. Robert and J. Y. Le Boudec, ”New models for pseudo self-similar traffic,” *Performance Evaluation* 30, 5768, 1997.

[78] T. Osogami, ”Analysis of multi-server systems via dimensionality reduction of Markov chains,” *PhD thesis*, Computer Science Department, Carnegie Mellon University, 2005.

[79] H. Masuyama and T. Takine, ”Analysis of an Infinite-Server Queue with Batch Markovian Arrival Streams,” *Queueing Systems* 42, no.3, 269-296, 2002.

[80] C.A. O’Cinneide, ”Characterization of phase-type distributions,” *Stochastic models* 6, 1-57, 1990.

[81] C.A. O’Cinneide, ”Phase-type distributions and invariant polytopes,” *Advances in Applied Probability* 23, 515-535, 1991.

[82] C.A. O’Cinneide, ”Triangular order of triangular phase-type distributions,” *Communications in statistics, Stochastic models* 9, 507-529, 1993.

[83] D.R. Cox, ”On the use of complex probabilities in the theory of stochastic processes,” *Proc. Camb. Phil. Soc* 51, 433-441, 1955.

[84] D. Aldous and L. Shepp, ”The least variable phase type distribution is erlang,” *Communications in statistics, Stochastic models* 3(3), 467-473, 1987.

[85] A. Cumani, ”On the canonical representation of markov processes modelling failure time distributions,” *Microelectronics and Reliability* 22(3), 583-602, 1982.

[86] M. Dehon and G. Latouche, ”A geometric interpretation of the relations between the exponential and generalized erlang distributions,” *Advances in Applied Probability*, Vol. 34, 885-897, 1982.

[87] Q.-M. HE and H. Zhang, ”Ph-invariant polytopes and coxian representations of phase type distributions,” *Communications in statistics, Stochastic models* 22, 383-410, 2006.

[88] S. Mocanu and C. Commault, ”Phase-type distributions and representations: some results and open problems for system theory,” *International Journal of Control*, 76(6), 566-580, 2003.

[89] R. Penrose, ”A generalized inverse for matrices,” *Proc. Cambridge Philos. Soc.* 51, 406-413, 1955.

[90] S. McCanne and S. Floyd, ”Ns (network simulator),” *http://www-nrg.ee.lbl.gov/ns*, 1995.

[91] National Institute of Standards and Technology *http://snad.ncsl.nist.gov/itg/nistnet/*

[92] I. N. Bronstein, K. A. Semendjajew, G. Musiol and H. Mühlig, ”Taschenbuch der Mathematik,” *Verlag Harri Deutsch*, (Frankfurt am Main, Thun) 1999.

[93] J. A. Shohat and J. D. Tamarkin, ”The Problem of Moments,” *American Mathematical Society*, (Providence, Rhode Island) 1943.

[94] N.G. Van Kampen, ”Stochastic Processes in Physics and Chemistry,” *North Holland*, 1981.

[95] L. Kleinrock, ”Queueing Systems, volume 1-2, *Theory, Wiley Interscience*, (New York), 1975-1976.

[96] R. Braden, ”Requirements for Internet Hosts - Communication Layers,” *RFC 1122*, October 1989.

[97] D. Chiu and R. Jain, ”Analysis of the Increase/Decrease Algorithms for Congestion Avoidance in Computer Networks,” *Journal of Computer Networks and ISDN*, vol. 17, 1-14, 1989.

[98] J. Stéger, P. Vaderna and G. Vattay, ”On the Propagation of Congestion Waves in the Internet,” *Physica A: Statistical Mechanics and its Applications*, Vol. 359, 784-792, January 2006.

[99] P. Vaderna and T. Éltető, ”Transient Analysis of a Queuing System with Matrix-Geometric Methods,” *Lecture Notes in Computer Science, Next Generation Teletraffic and Wired/Wireless Advanced Networking* Vol. 4003/2006, 21-33, 2006.

[100] T. Éltető and P. Vaderna, ”Időben változó sorbanállási rendszer vizsgálata mátrix-geometrikus eljárások segítségével,” *Híradástechnika* 2005/11, 38-44, 2005.

[101] T. Éltető and P. Vaderna, ”Finding upper-triangular representations for Phase-type distributions with 3 distinct real poles,” *Annals of Operations Research* 160/1, 139-172, April, 2008.

[102] T. Éltető, P. Vaderna and S. Molnár, ”Performance Analysis of TCP Networks Loaded by Web Traffic,” *18th International Teletraffic Congress, (ITC18)*, (Berlin, Germany), 2003.

[103] P. Vaderna, E. Strömberg and T. Éltető, ”Modelling Performance of HTTP/1.1,” *IEEE Global Telecommunications Conference (GLOBECOM 2003)*, (San Francisco, CA), 2003.