A Short Introduction to Queueing Theory

Andreas Willig

Technical University Berlin, Telecommunication Networks Group
Sekr. FT 5-2, Einsteinufer 25, 10587 Berlin
email: awillig@ee.tu-berlin.de

July 21, 1999

Contents

Chapter 1

Introduction

1.1 Disclaimer

This script is intended to be a short introduction to the field of queueing theory, serving as a module within the lecture “Leistungsbewertung von Kommunikationsnetzen” of Prof. Adam Wolisz from the Telecommunication Networks Group at Technical University Berlin. It covers the most important queueing systems with a single service center, for queueing networks only some basics are mentioned. This script is neither complete nor error free. However, we are interested in improving this script and we would appreciate any kind of (constructive) comment or “bug reports”. Please send all suggestions to *awillig@ft.ee.tu-berlin.de*.

In this script most of the mathematical details are omitted, instead often “intuitive” (or better: prosaic) arguments are used. Most of the formulas are only used during a derivation and have no numbers, however, the important formulas are numbered. The author was too lazy to annotate all statements with a reference, since most of the material can be found in the standard literature.

1.2 Scope of Queueing Theory

Queueing Theory is mainly seen as a branch of applied probability theory. Its applications are in different fields, e.g. communication networks, computer systems, machine plants and so forth. For this area there exists a huge body of publications, a list of introductory or more advanced texts on queueing theory is found in the bibliography. Some good introductory books are [9], [2], [11], [16].

The subject of queueing theory can be described as follows: consider a *service center* and a *population* of *customers*, which at some times enter the service center in order to obtain service. It is often the case that the service center can only serve a limited number of customers[1]. If a new customer arrives and the service is exhausted, he enters a *waiting line* and waits until the service facility becomes available. So we can identify three main elements of a service center: a population of customers, the service facility and the waiting line. Also within the scope of queueing theory is the case where several service centers are arranged in a *network* and a single customer can walk through this network at a specific path, visiting several service centers.

[1] Since queueing theory is applied in different fields, also the terms *job* and *task* are often used instead customer. The service center is often named *processor* or *machine*

As a simple example of a service center consider an airline counter: passengers are expected to check in, before they can enter the plane. The check-in is usually done by a single employee, however, there are often multiple passengers. A newly arriving and friendly passenger proceeds directly to the end of the queue, if the service facility (the employee) is busy. This corresponds to a FIFO service (first in, first out).

Some examples of the use of queueing theory in networking are the dimensioning of buffers in routers or multiplexers, determining the number of trunks in a central office in POTS, calculating end-to-end throughput in networks and so forth.

Queueing Theory tries to answer questions like e.g. the mean waiting time in the queue, the mean system response time (waiting time in the queue plus service times), mean utilization of the service facility, distribution of the number of customers in the queue, distribution of the number of customers in the system and so forth. These questions are mainly investigated in a stochastic scenario, where e.g. the interarrival times of the customers or the service times are assumed to be random.

The study of queueing theory requires some background in probability theory. Two modern introductory texts are [11] and [13], two really nice “classic” books are [7], [6].

1.3 Basic Model and Notation

A basic model of a service center is shown in figure 1.1. The customers arrive to the service center in a random fashion. The service facility can have one or several servers, each server capable of serving one customer at a time (with one exception), the service times needed for every customers are also modeled as random variables. Throughout this script we make the following assumptions:

- The customer population is of infinite size, the n-th customer C_n arrives at time τ_n. The interarrival time t_n between two customers is defined as $t_n := \tau_n - \tau_{n-1}$. We assume that the interarrival times t_n are iid random variables, i.e. they are independent from each other and all t_n are drawn from the same distribution with the distribution function

  $$A(t) := \Pr[t_n \leq t]$$

  and the probability density function (pdf) $a(t) := \frac{dA(t)}{dt}$
- The service times x_n for each customer C_n are also iid random variables with the common distribution function $B(t)$ and the respective pdf $b(t)$.

Queueing systems may not only differ in their distributions of the interarrival- and service times, but also in the number of servers, the size of the waiting line (infinite or finite), the service discipline and so forth. Some common service disciplines are:

FIFO: (First in, First out): a customer that finds the service center busy goes to the end of the queue.

LIFO: (Last in, First out): a customer that finds the service center busy proceeds immediately to the head of the queue. She will be served next, given that no further customers arrive.

Random Service: the customers in the queue are served in random order

Round Robin: every customer gets a time slice. If her service is not completed, she will re-enter the queue.

Priority Disciplines: every customer has a (static or dynamic) priority, the server selects always the customers with the highest priority. This scheme can use preemption or not.

The *Kendall Notation* is used for a short characterization of queueing systems. A queueing system description looks as follows:

$$A/B/m/N - S$$

where A denotes the distribution of the interarrival time, B denotes the distribution of the service times, m denotes the number of servers, N denotes the maximum size of the waiting line in the finite case (if $N = \infty$ then this letter is omitted) and the optional S denotes the service discipline used (FIFO, LIFO and so forth). If S is omitted the service discipline is always FIFO. For A and B the following abbreviations are very common:

- M (Markov): this denotes the exponential distribution with $A(t) = 1 - e^{-\lambda t}$ and $a(t) = \lambda e^{-\lambda t}$, where $\lambda > 0$ is a parameter. The name M stems from the fact that the exponential distribution is the only continuous distribution with the markov property, i.e. it is memoryless.

- D (Deterministic): all values from a deterministic "distribution" are constant, i.e. have the same value

- E_k (Erlang-k): Erlangian Distribution with k phases ($k \geq 1$). For the Erlang-k distribution we have

  $$A(t) = 1 - e^{-k\mu t} \sum_{j=0}^{k-1} \frac{(k\mu t)^j}{j!}$$

  where $\mu > 0$ is a parameter. This distribution is popular for modeling telephone call arrivals at a central office

- H_k (Hyper-k): Hyperexponential distribution with k phases. Here we have

  $$A(t) = \sum_{j=1}^{k} q_j (1 - e^{-\mu_j t})$$

  where $\mu_i > 0, q_i > 0, i \in \{1..k\}$ are parameters and furthermore $\sum_{j=1}^{k} q_j = 1$ must hold.

- G (General): general distribution, not further specified. In most cases at least the mean and the variance are known.

The most simple queueing system, the M/M/1 system (with FIFO service) can then be described as follows: we have a single server, an infinite waiting line, the customer interarrival times are iid and exponentially distributed with some parameter λ and the customer service times are also iid and exponentially distributed with some parameter μ.

We are mainly interested in *steady state* solutions, i.e. where the system after a long running time tends to reach a stable state, e.g. where the distribution of customers in the system does not change

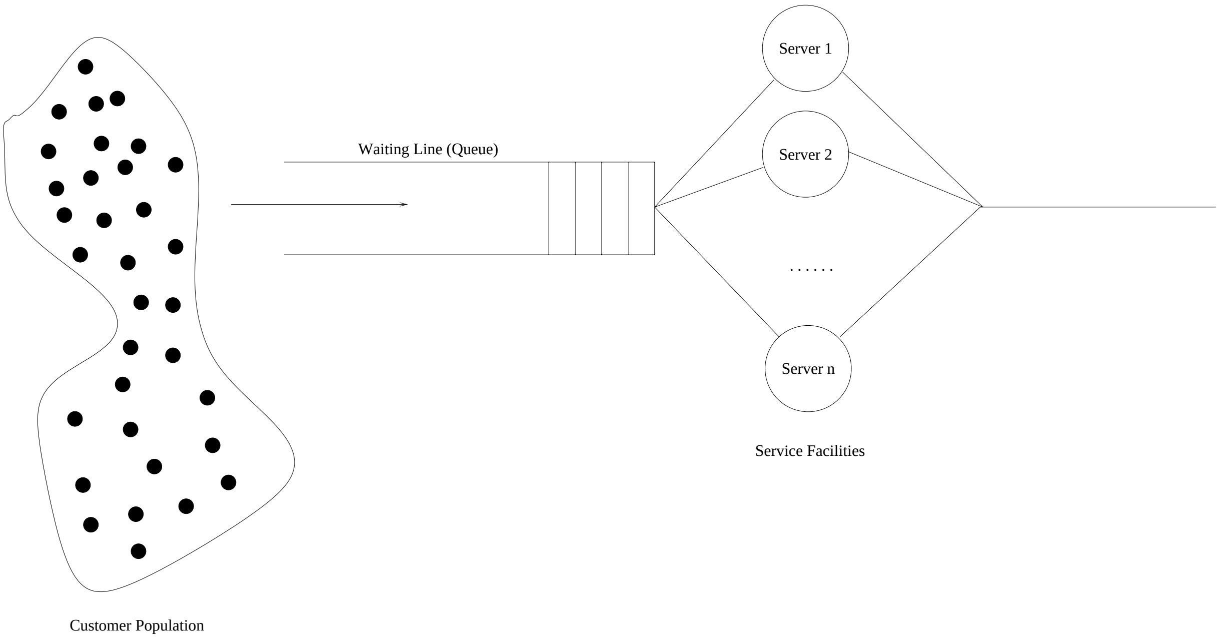

Figure 1.1: Model of a Service Center

(limiting distribution). This is well to be distinguished from *transient solutions*, where the short-term system response to different events is investigated (e.g. a batch arrival).

A general trend in queueing theory is the following: if both interarrival times and service times are exponentially distributed (markovian), it is easy to calculate any quantity of interest of the queueing system. If one distribution is not markovian but the other is, things are getting harder. For the case of G/G/1 queues one cannot do much; even the mean waiting times are not known.

1.4 Little's Law

Little's law is a general result holding even for G/G/1-Queues; it also holds with other service disciplines than FIFO. It establishes a relationship between the average number of customers in the system, the mean arrival rate and the mean customer response time (time between entering and leaving the system after getting service) in the steady state. The following derivation is from [11, chapter 7].

Denote $N(t)$ for the number of customers in the system at time t, $A(t)$ for the number of customer arrivals to the system in the time interval $[0,t]$, $D(t)$ for the number of customer departures from the system during $[0,t]$ and let T_i denote the response time of the i-th customer. Then clearly $N(t) = A(t) - D(t)$ holds (assuming the system is empty at $t = 0$). A sample path for $A(t)$ and $D(t)$ is shown in the upper part of figure 1.2 (Please be aware that customers do not necessarily leave the system in the same sequence they entered it). The *average number of arrivals* in the time interval $[0,t]$ is given by

$$\bar{A}(t) := \frac{A(t)}{t}$$

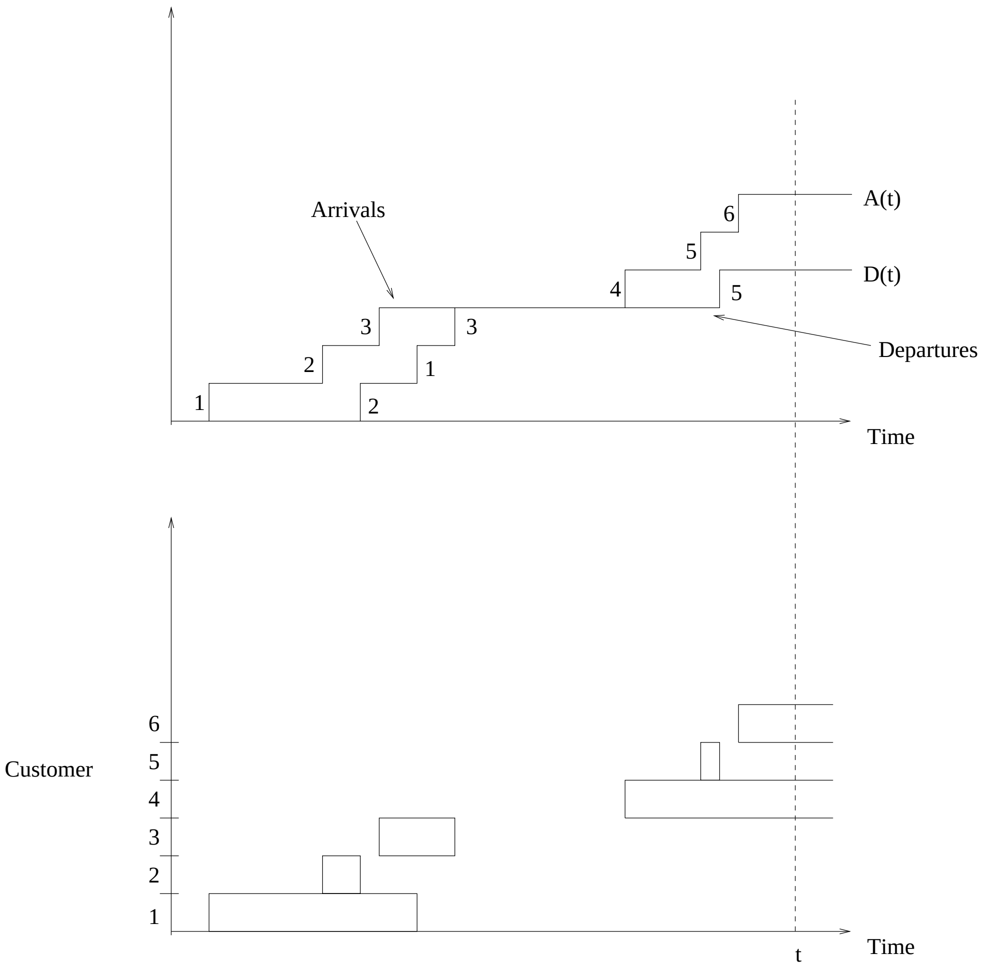

Figure 1.2: Little's Law

and we assume that

$$\lambda := \lim_{t\to\infty} \bar{A}(t)$$

exists and is finite. The value λ can be seen as the long term arrival rate. Furthermore the time average of the number of customers in the system is given by

$$\bar{N}(t) := \frac{1}{t}\int_0^t N(u)du$$

and we assume that $\bar{N} := \lim_{t\to\infty} \bar{N}(t)$ exists and is finite. Similarly we define the time customer average response time

$$\bar{T}(t) := \frac{1}{A(t)}\sum_{i=1}^{A(t)} T_i$$

Now consider a graph where $A(t)$ and $D(t)$ are shown simultaneously (see upper part of figure 1.2). Since always $A(t) \geq D(t)$ holds we have $N(t) \geq 0$ and the area between the two curves is given by

$$F(t) := \int_0^t (A(u) - D(u))du = \int_0^t N(u)du$$

We can take an alternative view to $F(t)$: it represents the sum of all customer response times which are active up to time t:

$$\sum_{i=1}^{A(t)} T_i$$

with the minor error that this expression takes also the full response times of the customers into account that are in the system at time t and which are present in the system up to a time $t_1 > t$ (see lower part of figure 1.2, where for each customer the bar corresponds to its system response time). This "overlap" is denoted $E(t)$ and now we can write

$$F(t) = \sum_{i=1}^{A(t)} T_i - E(t)$$

We assume that $E(t)$ is almost relatively small.

Now we can equate both expressions for $F(t)$:

$$\int_0^t N(u)du = \sum_{i=1}^{A(t)} T_i - E(t)$$

After division by $1/t$ and using $1 = \frac{A(t)}{A(t)}$ we arrive at:

$$\frac{1}{t}\int_0^t N(u)du = \frac{A(t)}{t}\frac{1}{A(t)}\sum_{i=1}^{A(t)} T_i - \frac{E(t)}{t}$$

Now we use the above definitions, go to the limit and use that $lim_{t\to\infty}\frac{E(t)}{t} = 0$ and finally arrive at **Little's Law**:

$$\bar{N} = \lambda\bar{T} \tag{1.1}$$

An alternative form of Little's Law arises when we assume that $\bar{N} = E[N]$ holds (with N being a steady state random variable denoting the number of customers in the system) and also $\bar{T} = E[T]$, then we have

$$E[N] = \lambda E[T] \tag{1.2}$$

A very similar form of Little's Law relates the mean number of customers in the queue (not in the system!!!), denoted as $\bar{N_q}$ (the underlying random variable for the number of customers in the queue is denoted as N_q) and the mean waiting time $\bar{W}$, i.e. the time between arrival of a customer and the start of its service. In this case Little's Law is

$$\bar{N_q} = \lambda\bar{W} \tag{1.3}$$

or in mean value representation

$$E[N_q] = \lambda E[W] \tag{1.4}$$

Chapter 2

Markovian Systems

The common characteristic of all *markovian systems* is that all interesting distributions, namely the distribution of the interarrival times and the distribution of the service times are exponential distributions and thus exhibit the markov (memoryless) property. From this property we have two important conclusions:

- The state of the system can be summarized in a single variable, namely the number of customers in the system. (If the service time distribution is not memoryless, this is not longer true, since not only the number of customers in the system is needed, but also the remaining service time of the customer in service.)

- Markovian systems can be directly mapped to a *continuous time markov chain* (CTMC) which can then be solved.

In this chapter we will often proceed as follows: deriving a CTMC and solve it by inspection or simple numerical techniques.

2.1 The M/M/1-Queue

The M/M/1-Queue has iid interarrival times, which are exponentially distributed with parameter λ and also iid service times with exponential distribution with parameter μ. The system has only a single server and uses the FIFO service discipline. The waiting line is of infinite size. This section is mainly based on [9, chapter 3].

It is easy to find the underlying markov chain. As the system state we use the number of customers in the system. The M/M/1 system is a pure birth-/death system, where at any point in time at most one event occurs, with an event either being the arrival of a new customer or the completion of a customer's service. What makes the M/M/1 system really simple is that the arrival rate and the service rate are not state-dependent. The state-transition-rate diagram of the underlying CTMC is shown in figure 2.1.

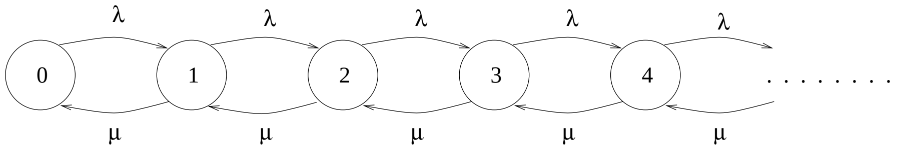

Figure 2.1: CTMC for the M / M / 1 queue

2.1.1 Steady-State Probabilities

We denote the steady state probability that the system is in state k ($k \in \mathbb{N}$) by p_k, which is defined by

$$p_k := \lim_{t \to \infty} P_k(t)$$

where $P_k(t)$ denotes the (time-dependent) probability that there are k customers in the system at time t. Please note that the steady state probability p_k does not dependent on t. We focus on a fixed state k and look at the *flows* into the state and out of the state. The state k can be reached from state $k-1$ and from state $k+1$ with the respective rates $\lambda P_{k-1}(t)$ (the system is with probability $P_{k-1}(t)$ in the state $k-1$ at time t and goes with the rate λ from the predecessor state $k-1$ to state k) and $\mu P_{k+1}(t)$ (the same from state $k+1$). The total flow into the state k is then simply $\lambda P_{k-1}(t) + \mu P_{k+1}(t)$. The state k is left with the rate $\lambda P_k(t)$ to the state $k+1$ and with the rate $\mu P_k(t)$ to the state $k-1$ (for $k = 0$ there is only a flow coming from or going to state 1). The total flow out of that state is then given by $\lambda P_k(t) + \mu P_k(t)$ The total rate of change of the flow into state k is then given by the difference of the flow into that state and the flow out of that state:

$$\frac{dP_k(t)}{dt} = (\lambda P_{k-1}(t) + \mu P_{k+1}(t)) - (\lambda P_k(t) + \mu P_k(t)),$$

, however, in the limit ($t \to \infty$) we require

$$\frac{dP_k(t)}{dt} = 0$$

so we arrive at the following steady-state flow equations:

$$\begin{array}{rcl} 0 & = & \mu p_1 - \lambda p_0 \\ 0 & = & \lambda p_0 + \mu p_2 - \lambda p_1 - \mu p_1 \\ 0 & = & \ldots\ldots \\ 0 & = & \lambda p_{k-1} + \mu p_{k+1} - \lambda p_k - \mu p_k \\ 0 & = & \ldots\ldots \end{array}$$

These equations can be recursively solved in dependence of p_0:

$$p_k = \left(\frac{\lambda}{\mu}\right)^k p_0$$

Furthermore, since the p_k are probabilities, the *normalization condition*

$$\sum_{k=0}^{\infty} p_k = 1$$

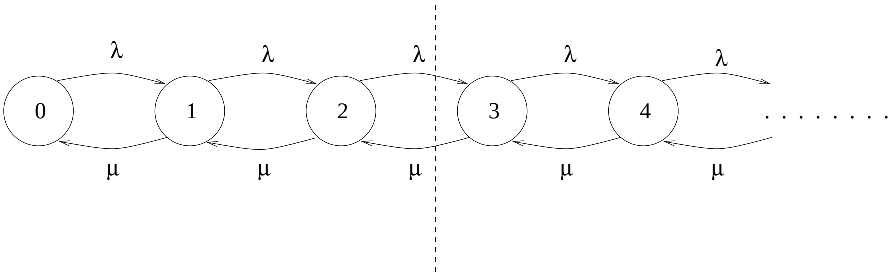

Figure 2.2: CTMC for the M / M / 1 queue

says that

$$1 = p_0 + \sum_{k=1}^{\infty} p_k = p_0 + \sum_{k=1}^{\infty} p_0 \left(\frac{\lambda}{\mu}\right)^k = p_0 \left(\sum_{k=0}^{\infty} \left(\frac{\lambda}{\mu}\right)^k\right) = p_0 \frac{1}{1-\frac{\lambda}{\mu}}$$

which gives

$$p_0 = 1 - \frac{\lambda}{\mu} =: 1 - \rho \tag{2.1}$$

To summarize the results, the **steady state probabilities** of the M / M / 1 markov chain are given by

$$p_0 = 1 - \frac{\lambda}{\mu} \tag{2.2}$$

$$p_k = \left(\frac{\lambda}{\mu}\right)^k p_0 \tag{2.3}$$

Obviously, in order for p_0 to exist it is required that $\lambda < \mu$, otherwise the series will diverge. This is the *stability condition* for the M / M / 1 system. It makes also sense intuitively: when more customers arrive than the system can serve, the queue size goes to infinity.

A second derivation making use of the flow approach is the following: in the steady state we can draw a line into the CTMC as in figure 2.2 and we argue, that in the steady state the following principle holds: the flow from the left side to the right side equals the flow from the right side to the left side. Transforming this into flow equations yields:

$$\begin{aligned} \lambda p_0 &= \mu p_1 \\ \lambda p_1 &= \mu p_2 \\ \ldots &= \ldots\ldots \\ \lambda p_{k-1} &= \mu p_k \\ \ldots &= \ldots\ldots \end{aligned}$$

This approach can be solved using the same techniques as above.

The just outlined method of deriving a CTMC and solving the flow equations for the steady state probabilities can be used for most markovian systems.

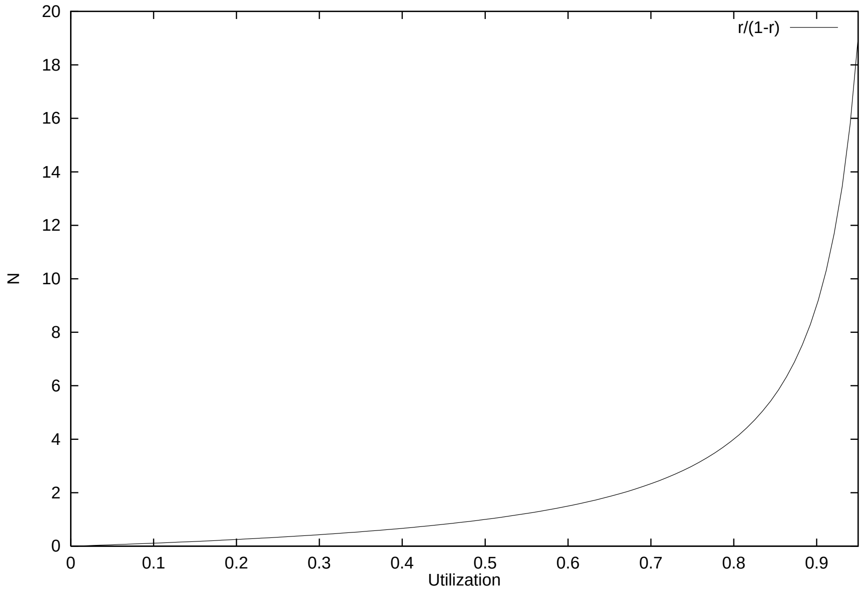

Figure 2.3: Mean Number of Customers vs. Utilization

2.1.2 Some Performance Measures

Utilization

The utilization gives the fraction of time that the server is busy. In the M/M/1 case this is simply the complementary event to the case where the system is empty. The utilization can be seen as the steady state probability that the system is not empty at any time in the steady state, thus

$$\text{Utilization} := 1 - p_0 = \rho \tag{2.4}$$

Mean number of customers in the system

The mean number of customers in the system is given by

$$\bar{N} = E[N] = \sum_{k=0}^{\infty} kp_k = p_0 \left(\sum_{k=0}^{\infty} k\rho^k \right) = (1-\rho)\frac{\rho}{(1-\rho)^2} = \frac{\rho}{1-\rho} \tag{2.5}$$

where we have used the summation

$$\sum_{k=0}^{\infty} kx^k = \frac{x}{(1-x)^2}$$

for $|x| < 1$

The mean number of customers in the system for varying utilizations is shown in figure 2.3. As can be seen $\bar{N}$ grows to infinity as $\rho \to 1$, thus for higher utilizations the system tends to get unstable. This trend is especially observable for utilizations of 70 % or more.

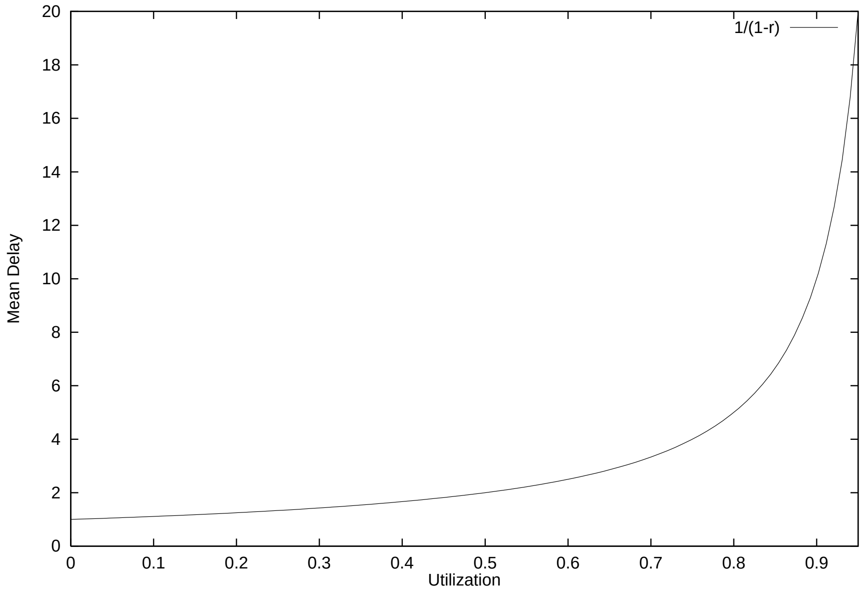

Figure 2.4: Mean Delay vs. Utilization

Mean Response Time

The mean response time T is the mean time a customer spends in the system, i.e. waiting in the queue and being serviced. We simply apply Little's law to find

$$\bar{T} = \frac{\bar{N}}{\lambda} = \frac{1/\mu}{1-\rho} = \frac{1}{\mu-\lambda} \tag{2.6}$$

For the case of $\mu = 1$ the mean response time (mean delay) of a customer is shown in figure 2.4 (for $\mu = 1$). This curve shows a behaviour similar to the one for the mean number of customers in the system.

Tail Probabilities

In applications often the following question arises: we assume that we have an M/M/1 system, however, we need to restrict the number of customers in the system to a finite quantity. If a customer arrives at a full system, it is lost. We want to determine the size of the waiting line that is required to lose customers only with a small probability. As an example consider e.g. a router for which the buffer space is finite and packets should be lost with probability 10^{-6}. In principle this is a M/M/1/N queue, however, we use an M/M/1 queue (with infinite waiting room) as an approximation. We are now interested in the probability that the system has k or more customers (the probability $\Pr[N > k]$ is called a *tail probability*) and thus would lose a customer in reality. We have

$$\Pr[N > k] = 1 - \Pr[N \leq k] = 1 - \sum_{\nu=0}^{k} p_\nu = 1 - p_0 \frac{1-\rho^{k+1}}{1-\rho} = \rho^{k+1} \tag{2.7}$$

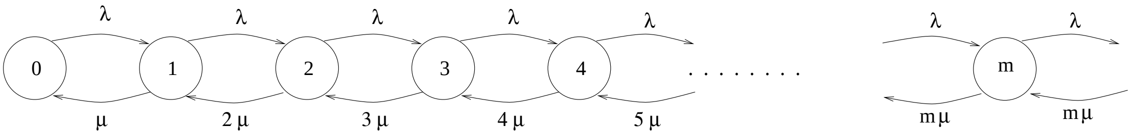

Figure 2.5: CTMC for the M / M / m queue

2.2 The M/M/m-Queue

The M / M / m-Queue ($m > 1$) has the same interarrival time and service time distributions as the M / M / 1 queue, however, there are m servers in the system and the waiting line is infinitely long. As in the M / M / 1 case a complete description of the system state is given by the number of customers in the system (due to the memoryless property). The state-transition-rate diagram of the underlying CTMC is shown in figure 2.5. The M / M / m system is also a pure birth-death system.

2.2.1 Steady-State Probabilities

Using the above sketched technique of evaluating the flow equations together with the well-known geometric summation yields the following steady state probabilities:

$$p_0 = \left[\sum_{k=0}^{m-1} \frac{(m\rho)^k}{k!} + \left(\frac{(m\rho)^m}{m!} \right) \left(\frac{1}{1-\rho} \right) \right]^{-1} \tag{2.8}$$

$$p_k = \begin{cases} p_0 \frac{(m\rho)^k}{k!} & : \quad k \le m \\ p_0 \frac{\rho^k m^m}{m!} & : \quad k \ge m \end{cases} \tag{2.9}$$

with $\rho = \frac{\lambda}{\mu}$ and clearly assuming that $\rho < 1$.

2.2.2 Some Performance Measures

Mean number of customers in the system

The mean number of customers in the system is given by

$$\bar{N} = E[N] = \sum_{k=0}^{\infty} k p_k = m\rho + \rho \frac{(m\rho)^m}{m!} \frac{p_0}{(1-\rho)^2} \tag{2.10}$$

The mean response time again can be evaluated simply using Little's formula.

For the case of M=10 we show the mean number of customers in the system for varying ρ in figure 2.6.

Queueing Probability

We want to evaluate the probability that an arriving customer must enter the waiting line because there is currently no server available. This is often used in telephony and denotes the probability that a newly arriving call at a central office will get no trunk, given that the interarrival times and service times (call durations) are exponentially distributed (in "real life" it is not so easy to justify

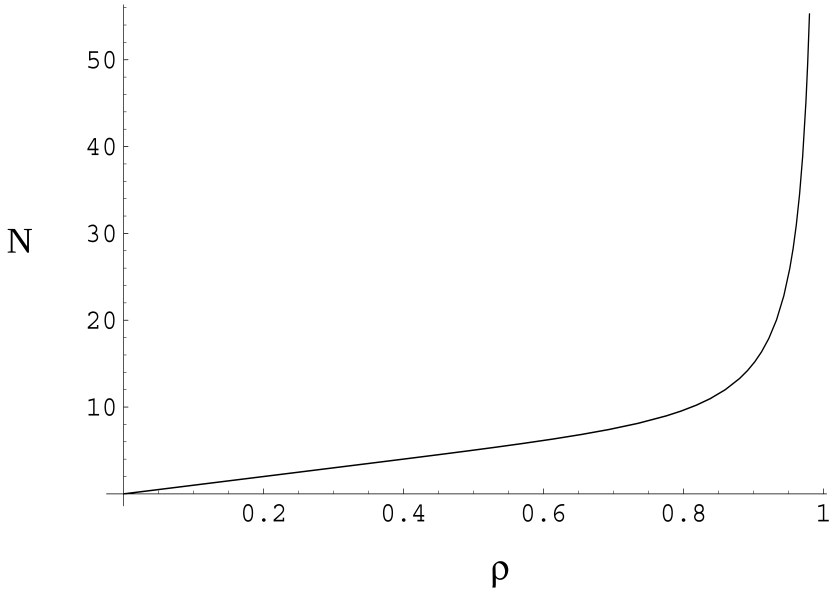

Figure 2.6: Mean Number of Customers in the system for the M/M/10-Queue

this assumption). This probability can be calculated as follows:

$$\Pr[\text{Queueing}] = \sum_{k=m}^{\infty} p_k = \sum_{k=m}^{\infty} p_0 \frac{(m\rho)^k}{m!} \frac{1}{m^{k-m}} = \frac{\left(\frac{(m\rho)^m}{m!}\right)\left(\frac{1}{1-\rho}\right)}{\left[\sum_{k=0}^{m-1} \frac{(m\rho)^k}{k!} + \left(\frac{(m\rho)^m}{m!}\right)\left(\frac{1}{1-\rho}\right)\right]} \tag{2.11}$$

and is often denoted as *Erlangs C Formula,* abbreviated with $C(m, \rho)$

2.3 The M/M/1/K-Queue

The M/M/1/K-Queue has exponential interarrival time and service time distributions, each with the respective parameters λ and μ. The customers are served in FIFO-Order, there is a single server but the system can only hold up to K customers. If a new customer arrives and there are already K customers in the system the new customer is considered lost, i.e. it drops from the system and never comes back. This is often referred to as *blocking.* This behaviour is necessary, since otherwise (e.g. when the customer is waiting outside until there is a free place) the arrival process will be no longer markovian. As in the M/M/1 case a complete description of the system state is given by the number of customers in the system (due to the memoryless property). The state-transition-rate diagram of the underlying CTMC is shown in figure 2.7. The M/M/1/K system is also a pure birth-death system. This system is better suited to approximate “real systems” (like e.g. routers) since buffer space is always finite.

2.3.1 Steady-State Probabilities

One can again using the technique based on evaluation of the flow equations to arrive at the steady state probabilities p_k. However, since the number of customers in the system is limited, the arrival process is state dependent: if there are fewer than K customers in the system the arrival rate is λ,

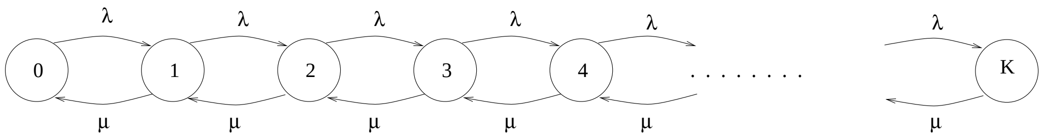

Figure 2.7: CTMC for the M/M/1/K queue

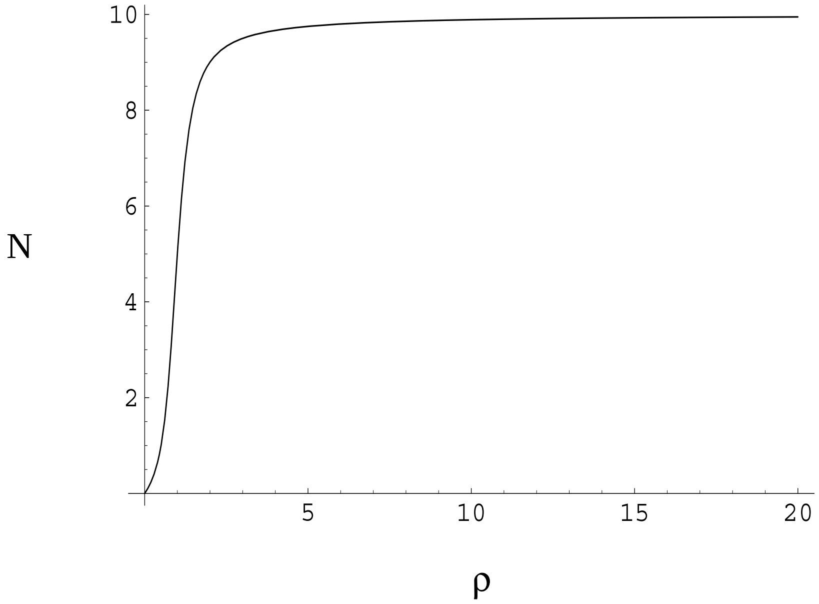

Figure 2.8: Mean number of Customers in the system for M/M/1/10-queue

otherwise the arrival rate is 0. It is then straightforward to see that the steady state probabilities are given by:

$$p_0 = \frac{1-\rho}{1-\rho^{K+1}} \tag{2.12}$$

$$p_k = p_0\rho^k \tag{2.13}$$

where $1 \le k \le K$ and again $\rho = \frac{\lambda}{\mu}$ holds. It is interesting to note that the system is stable even for $\rho > 1$

2.3.2 Some Performance Measures

Mean number of customers in the system

The mean number of customers in the system is given by

$$\bar{N} = E[N] = \sum_{k=0}^{K} kp_k = ... = \begin{cases} \frac{\rho}{1-\rho} - \frac{K+1}{1-\rho^{K+1}}\rho^{K+1} & : \quad \rho \neq 1 \\ \frac{K}{2} & : \quad \rho = 1 \end{cases} \tag{2.14}$$

The mean number of customers in the system is shown in figure 2.8 for varying ρ and for $K = 10$. Please note that for this queue ρ can be greater than one while the queueing system remains stable.

The mean response time again can be evaluated simply using Little's formula.

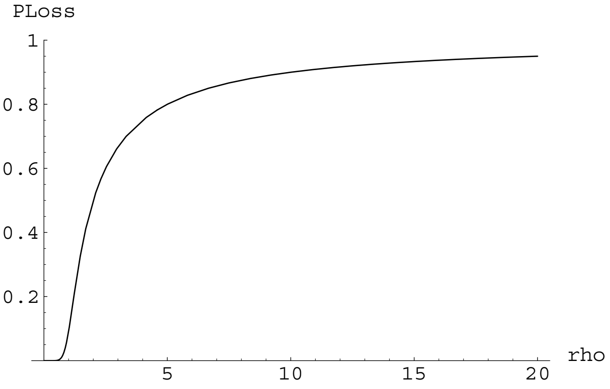

Figure 2.9: Loss Probability for M/M/1/20

Loss Probability

The loss probability is simply the probability that an arriving customer finds the system full, i.e. the loss probability is given as p_K with

$$p_{Loss} := p_K = \begin{cases} \frac{\rho^K - \rho^{K+1}}{1-\rho^{K+1}} & : \quad \rho \neq 1 \\ \frac{1}{K+1} & : \quad \rho = 1 \end{cases} \tag{2.15}$$

For the case of 10 servers the loss probability for varying ρ is shown in figure 2.9

In section 2.1 we have considered the problem of dimensioning a router's buffer such that customers are lost only with a small probability and used the M/M/1 queue as an approximation, where an M/M/1/K queue with unknown K may be more appropriate. However, it is not possible to solve equation 2.15 algebraically for K when p_{Loss} is given (at least if no special functions like LambertW [1] are used).

2.4 A comparison of different Queueing Systems

In this section we want to compare three different systems in terms of mean response time (mean delay) vs. offered load: a single M/M/1 server with the service rate $m\mu$, a M/M/m system and a system where m queues of M/M/1 type with service rate μ are in parallel, such that every customer enters each system with the same probability.

The answer to this question can give some hints on proper decisions in scenarios like the following: given a computer with a processor of type X and given a set of users with long-running number cruncher programs. These users are all angry because they need to wait so long for their results. So the management decides that the computer should be upgraded. There are three possible options:

- buy $n-1$ additional processors of type X and plug these into the single machine, thus yielding a multiprocessor computer
- buy a new processor of type Y, which is n times stronger than processor X and replacing it, and let all users work on that machine
- provide each user with a separate machine carrying a processor of type X, without allowing other users to work on this machine

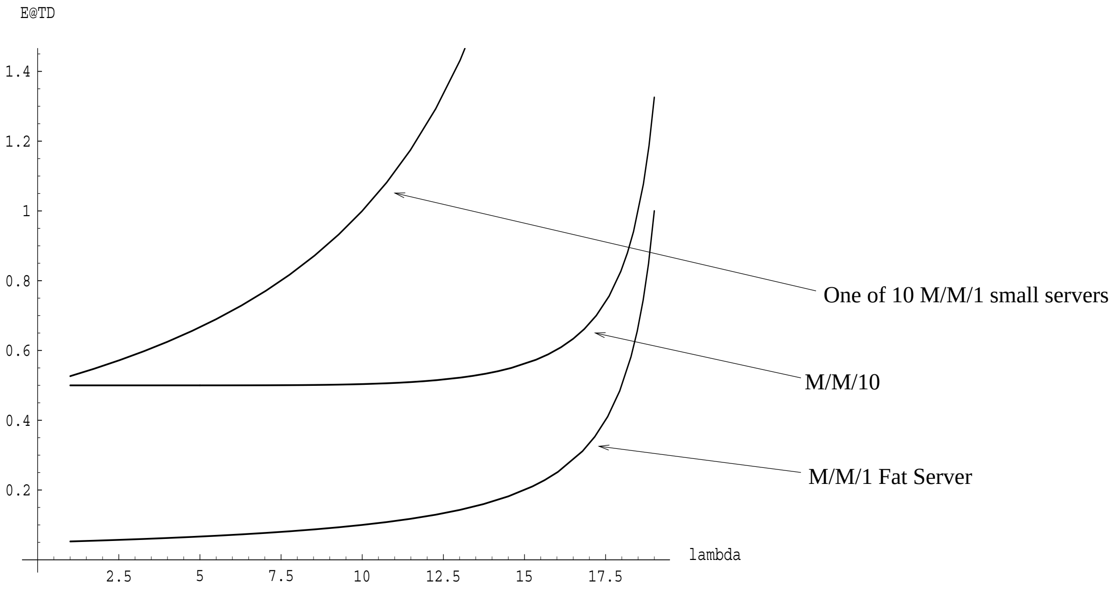

Figure 2.10: Mean Response Times for three different systems

We show that the second solution yields the best results (smallest mean delays), followed by the first solution, while the last one is the worst solution. The first system corresponds to an M / M / m system, where each server has the service rate μ and the arrival rate to the system is λ. The second system corresponds to an M / M / 1 system with arrival rate λ and service rate $m \cdot \mu$. And, from the view of a single user, the last system corresponds to an M / M / 1 system with arrival rate λ/m and service rate μ. The mean response times for $m = 10$ and $\mu = 2$ are for varying λ shown in figure 2.10.

An intuitive explanation for the behaviour of the systems is the following: in the case of 10 parallel M / M / 1 queues there is always a nonzero probability that some servers have many customers in their queues while other servers are idle. In contrast to that, in the M / M / m case this cannot happen. In addition to that, the fat single server is especially for lighter loads better than the M / M / 10 system, since if there are only $k < 10$ customers in the system the M / M / 10 system has a smaller overall service rate $k \cdot \mu$, while in the fat server all customers are served with the full service rate of $10 \cdot \mu = 20$

Chapter 3

The M/G/1-System

In this chapter we will show some basic properties of the more advanced M/G/1 system. In this system we have a single server, an infinite waiting room, exponentially distributed interarrival times (with parameter λ) and an arbitrary service time distribution, for which at least the mean value μ and the standard deviation is known. The service discipline is FIFO. However, before starting with the M/G/1 queue we present some facts from renewal theory.

3.1 Some Renewal Theory Results

Consider the following scenario: consider the time axis, going from far in the past to infinity (please refer to figure 3.1). There occur events E_k, $k \in \mathbb{N}$ at the times τ_k. The interevent time is defined as $x_k := \tau_k - \tau_{k-1}$, and all interevent times are drawn from the same distribution $F(x)$ (iid) with the density function (pdf) $f(x)$, the mean value $E[(\tau_k - \tau_{k-1})] = \int_0^\infty x f(x)dx = m_1$ and the second moment $E[(\tau_k - \tau_{k-1})^2] = \int_0^\infty x^2 f(x)dx = m_2$ (thus having the variance $\sigma^2 = m_2 - m_1^2$). We say that the time interval $[\tau_k, \tau_{k+1})$ is the *lifetime* of the event E_k. Now we look at an arbitrary time τ at the system. The last event that occured is E_{n-1} at time τ_{n-1}, the next event E_n occurs at time τ_n. We assume $\tau_{n-1} \leq \tau < \tau_n$, so τ is within the lifetime of the event E_{n-1}. We define $\tau - \tau_{n-1}$ the *age* of E_k and $\tau_n - \tau$ to be the *residual lifetime* of E_k. Now we may ask for the distribution function $G(x)$ or the corresponding pdf $g(x)$ of the residual lifetime Y, given the distribution of the interevent times. In addition we ask for the distribution function $\hat{F}(x)$ (or its pdf $\hat{f}(x)$) of the special interarrival time $X := \tau_n - \tau_{n-1}$. The surprising fact is that in general $F_X \neq \hat{F}_X$, what means that *if we look at a random point in time at the system, the special lifetime distribution of the currently active event is not the same as the lifetime distribution of all other events.*

One example where questions like these arise is the following: consider a machine that always needs some maintenance, especially it needs some oil from time to time. If the machine runs out of oil it will break down and a repairman must be called, who fills a new portion of oil into the machine. However, due to several circumstances the machine uses (randomly) varying rates of oil, so the next time when the machine runs out of oil, will be random. Now the machine is sold. The buyer then has an interest to know when the next breakdown will probably come. It is intuitively clear, that the probability, that the next breakdown happens within a short time is greater, when the last breakdown is far away and conversely, if the the last breakdown was yesterday we expect the machine to run for

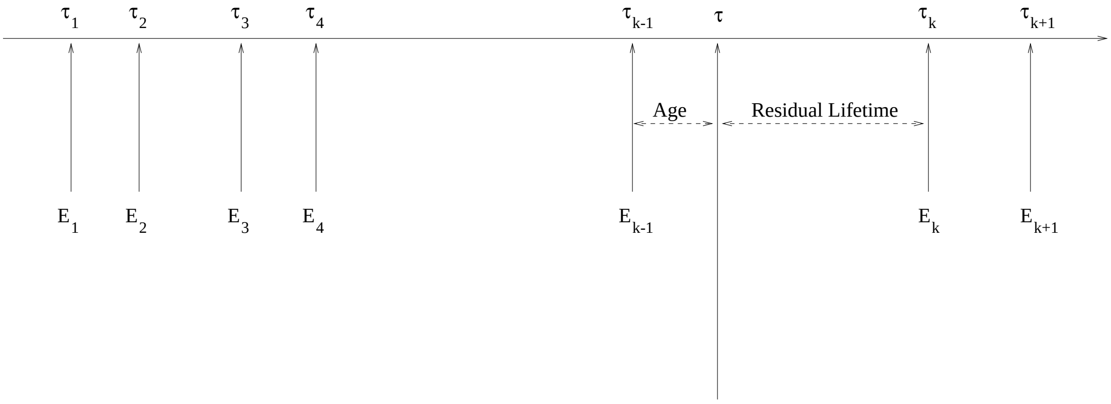

Figure 3.1: Residual Lifetime

a longer time. Thus the buyer is interested in the distribution of the residual lifetime.

In various textbooks on probability theory (e.g. [7]) the following important results are shown (quoted from [9]):

- The special interarrival time X has the density function

$$\hat{f}(x) = \frac{xf(x)}{m_1} \tag{3.1}$$

- The residual lifetime has the density function

$$g(x) = \frac{1 - F(x)}{m_1} \tag{3.2}$$

- The mean residual lifetime is given by

$$E[Y] = \frac{m_2}{2m_1} = \frac{m_1}{2} + \frac{\sigma^2}{2m_1} \tag{3.3}$$

  Another useful representation of the mean residual lifetime is given by

$$E[Y] = \frac{m_1}{2}(1 + C_X^2) \tag{3.4}$$

  where C_X^2 is the squared *coefficient of variation* of the distribution of the interarrival times. In general, for a random variable X with finite expectation and variance the squared coefficient of variation is defined as

$$C_X^2 := \frac{\text{Var}[X]}{(E[X])^2} \tag{3.5}$$

  It provides a rough measure on how much a random variable varies about its mean.

A special application of these results is the so-called *taxi paradoxon*: consider you are leaving your home at a random point in time and go to the border of the next street, planning to take a taxi. You know that taxis arrive at your location according to an exponential interarrival-time distribution with parameter λ. So whats the mean time that you must wait for the next taxi? Two answers would make sense:

- Since the mean interarrival time is $1/\lambda$ you can expect that you have hit “the middle” of an interarrival time, so in the mean you have to wait for $1/(2\lambda)$. This is the same expectation as when the interarrival times are deterministic.
- Since the exponential distribution is memoryless, the residual lifetime should have the same distribution as the interarrival times with the same parameter λ and thus having the same mean residual lifetime $1/\lambda$, twice the time as in the first answer. Furthermore for the same reasons we expect the age of the current event also be exponentially distributed with parameter λ. So we conclude that the interarrival time we have hit has the mean $2/\lambda$.

We can evaluate equation 3.3 for the exponential distribution with $m_1 = 1/\lambda$ and $m_2 = 2/\lambda^2$ to see that $E[Y] = 1/\lambda$. Furthermore we see that the density function of the special interarrival time $\hat{f}(x)$ corresponds to a gamma distribution with parameters $\alpha = 2$ and $\tilde{\lambda} = \lambda$, which is the distribution function of the sum of two exponential distributed random variables with the same parameter. So the second answer is correct. An intuitive justification of this answer is that it is more likely to hit a long interarrival time than to hit a short one.

3.2 The PASTA Theorem

In this section we investigate the following question: when we look at the number of customers in a queueing system at “random points” in time, do then have all “random points” the same properties or do there exist points where the results differ fundamentally?

The answer is: yes, it makes a difference how you choose the points. A simple example illustrating this is the D/D/1 queue (with fixed interarrival times of λ seconds and fixed service times of μ seconds, $\mu < \lambda$. If we now choose “randomly” the arrival times of new customers as our random points it is clear that we will see zero customers in the system every time (since customers are served faster than they arrive). If we choose the random times with uniformly distributed time differences there is always a probability of $\lambda/\mu > 0$ to find a customer in the system. Thus, the final result of this section is that *arrival times are not random times.* One important exception from this rule is when the arrival times come from a poisson process (with exponentially distributed interarrival times). In this case the arrival times are “random enough” and a customer arriving to a queue in the steady state sees exactly the same statistics of the number of customers in the system as for “real random times”. In a more condensed form this is expressed as **P**oisson **A**rrivals **S**ee **T**ime **A**verages, abbreviated with **PASTA**. This property comes also from the memoryless property of the exponential distribution.

3.3 The Mean Response Time and Mean Number of Customers in the System / Pollaczek-Khintchine Mean Value Formulas

We first derive the mean value for the response time (taken from [11]). We assume that the arbitrary service time distribrution is given by $B(t)$ with the pdf $b(t)$. The random variable for the service times itself is denoted by B. The service rate (the inverse of the mean service time) is denoted as $\mu = 1/E[B]$, the arrival rate of the poisson arrival process has rate λ. If we define $\rho = \min\{1, \lambda/\mu\}$

then ρ can be interpreted as the utilization of the server, i.e. the steady state fraction of time that the server is busy. The random variable for the number of customers in the queue is denoted by N_q.

Now we focus on a specific arriving customer. Since we have poisson arrivals this customer sees the same system statistics as seen at every other random point in time (PASTA). The expected waiting time of the customer can then be calculated as follows:

$$E[W] = E[N_q]E[B] + \rho E[R]$$

where the first term $E[N_q]E[B]$ accumulates all the service times of the customers currently waiting in the queue (FIFO service discipline) and the second term $\rho E[R]$ counts the residual service time (residual lifetime) of the customer currently in service, weighted by the probability ρ that the server is busy.

From section 3.1 we know that

$$E[R] = \frac{E[B]}{2}(1 + C_B^2)$$

where C_B^2 is the squared coefficient of variation for the service time random variable. Furthermore, Little's Law says that $E[N_q] = \lambda E[W]$ so we get

$$E[W] = \lambda E[W]E[B] + \rho E[R]$$

which can be solved for $E[W]$ to get

$$E[W] = \rho E[R]\frac{1}{1 - \lambda E[B]}$$

and since $E[B] = 1/\mu$ this can be simplified to

$$E[W] = E[R]\frac{\rho}{1-\rho} = \frac{E[B]}{2}\frac{\rho}{1-\rho}(1 + C_B^2)$$

Now the mean response time can be calculated as

$$E[T] = E[B] + E[W]$$

which yields the final result

$$E[T] = E[B]\left(1 + \frac{\rho(1 + C_B^2)}{2(1-\rho)}\right) \tag{3.6}$$

After applying Little's Law to the last formula we get

$$E[N] = \rho + \frac{\rho^2(1 + C_B^2)}{2(1-\rho)} \tag{3.7}$$

The last two equations are known as **Pollaczek-Khintchine Mean Value Formulas**. In figure 3.2 we show the mean number of customers in the system for different coefficients of variation and for varying ρ. This result is worth some notes:

- We can see that all M/G/1 systems have in common that they tend to get unstable as ρ approaches 1.

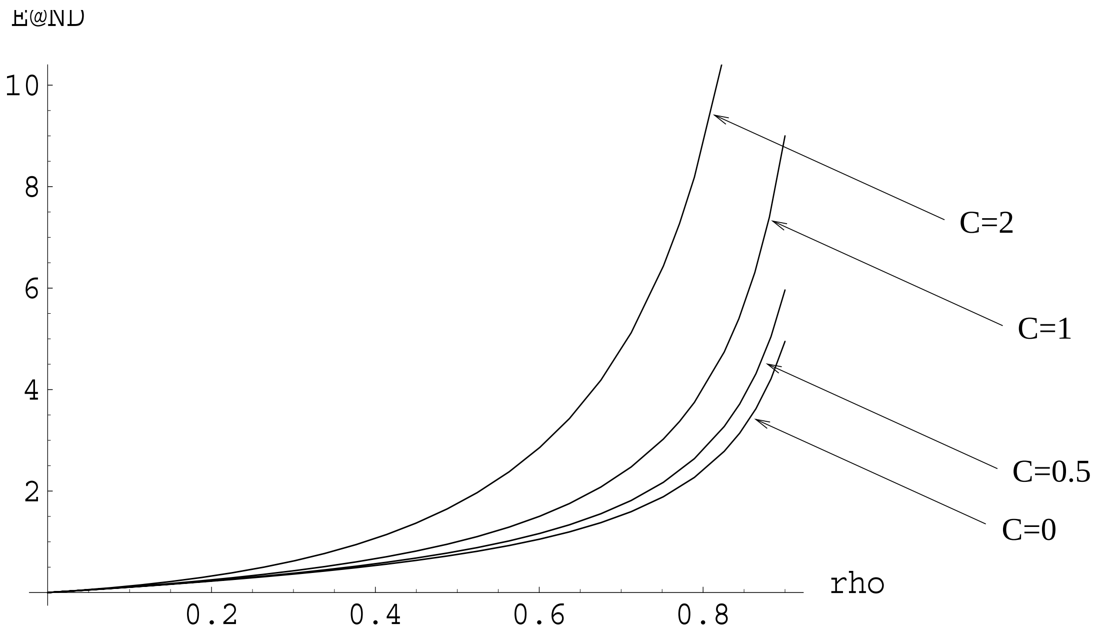

Figure 3.2: Mean number of Customers in the system for different coefficients of variation

- We can see that the coefficient of variation of the service times has a strong influence on the mean values. The exponential distribution has a squared coefficient of variation of 1, while the deterministic distribution (with constant service times) has a zero variance and thus $C_B^2 = 0$ for this distribution. For this reason we can achieve always the highest throughput (lowest delays) for deterministic service times. Randomness increases service times, especially for distribution with large variations, where large service times occur with higher probability, causing the customers in the queue to wait for a longer time.

3.4 Distribution of the Number of Customers in the System

In this section we show a method for deriving the distribution of the number of customers in the system in the steady state in an M/G/1 system. The derivation is mainly taken from [11] and [9]. This information is often very important for dimensioning of real systems, e.g. buffers in routers, since it allows for calculation of tail probabilities $P[N > x]$ where x is the amount of memory we can afford and the tail probability gives an estimation of the number of packets we will lose in our router. However, we will not directly derive expressions for this distribution, but for its *ordinary generating function* (ogf) or *probability generating function* (pgf), in terms of the Laplace transform of the service time distribution. A rough introduction to ogf's can be found in appendix A.1, to Laplace transforms in appendix A.2.

Since the service times are iid, we take a random variable B with distribution function $B(x)$ and density function $b(x)$ as representative for all service times. As usual, the arrival process is a poisson process with parameter λ. Next the family of random variables V_i denotes the respective number of customer arrivals in the ith service time. Clearly V_i depends on the length of the ith service period, which always has the distribution $B(x)$. Thus it is reasonable to assume that all V_i have the same distribution and thus form an iid sequence of random variables (a rigorous proof of this would be of only technical interest).

Now we look at the *imbedded markov chain*: if we want to describe the state of the system at a random point in time, in general we need two variables: the number $N(t)$ of customers in the system at that time and furthermore the service time $X(t)$ already received by the customer currently in service. The latter variable is needed, since the service times in general do not have the memoryless property. However, if we choose as special times the time instants t_n, where the nth customer in service leaves the system and the service time of the next customer starts, we have $X(t) = 0$. At this times the system state is completely given by $N(t_n)$ and so we have a markov chain like the ones for M/M/1.

Now we define $N_i := N(t_i)$ and it is straightforward to establish the following recurrence equation:

$$N_{i+1} = (N_i - 1)^+ + V_i \tag{3.8}$$

where $(x)^+ := \max\{0, x\}$. This equation in words: the number of customers in the system at time instant t_{n+1} is given by the number at time t_n minus the customer leaving the system plus newly arriving customers. Additionally we assume $N_0 = 0$. If the system is in steady state, the time dependency disappears in the long run and the random variables N_i and V_i converge to random variables N and V respectively. Taking the limits yields:

$$N = (N - 1)^+ + V$$

Let $G_N(z)$ be the probability generating function (pgf) of N. From the last equation and from the convolution property of the pgf[1] we have

$$G_N(z) = G_{(N-1)^+ + V}(z) = G_{(N-1)^+}(z) \cdot G_V(z) \tag{3.9}$$

where from the definition

$$G_N(z) = \sum_{k=0}^{\infty} \Pr[N = k] \cdot z^k$$

Also from the definition of the pgf we have

$$G_{(N-1)^+}(z) = \sum_{i=0}^{\infty} \Pr[(N-1)^+ = i] \cdot z^i$$

The random variable $(N-1)^+$ takes the value 0 if and only if $N = 0$ or $N = 1$ holds, thus we have $\Pr[(N-1)^+ = 0] = \Pr[N = 0] + \Pr[N = 1]$. On the other hand,$(N-1)^+$ takes the value i ($i > 0$) if and only if $N = i + 1$, thus $\Pr[(N-1)^+ = i] = \Pr[N = i + 1]$. Collecting terms yields:

$$\begin{aligned} G_{(N-1)^+}(z) &= (\Pr[N=0] + \Pr[N=1]) \cdot z^0 + \sum_{\nu=1}^{\infty} \Pr[N = \nu + 1] \cdot z^{\nu} \\ &= \Pr[N=0] + \frac{1}{z} \cdot \left(\Pr[N=1] \cdot z + \sum_{\nu=2}^{\infty} \Pr[N = \nu] \cdot z^{\nu} \right) \\ &= \Pr[N=0] + \frac{1}{z} \cdot (G_N(z) - \Pr[N=0]) \end{aligned}$$

[1] Recall that the expression $(N-1)^+ + V$ is the sum of two random variables with values from natural numbers and not the sum of two probability sequences. Thus the probability sequence of the sum is given by the convolution of the sequences for $(N-1)^+$ and V.

If we take ρ as the utilization of the server, then the server will in the steady state be free with probability $1-\rho$, thus we have $\Pr[N=0]=1-\rho$, yielding

$$G_{(N-1)^+}(z)=\frac{G_N(z)+(1-\rho)\cdot(z-1)}{z}$$

Next we derive the pgf for V, for which we need to introduce the Laplace transform of the service time B[2]. If the service interval has the length x then V has a poisson distribution with parameter λx, since arrivals are markovian. The poisson distribution is defined as

$$p(k;t)=e^{-t}\frac{t^k}{k!}$$

The probability generating function of V is defined as

$$\begin{aligned} G_V(z) &= \sum_{\nu=0}^{\infty}\Pr[\nu \text{ customers arrive during service time}]\cdot z^\nu \\ &= =\sum_{\nu=0}^{\infty}\Pr[V=\nu]\cdot z^\nu \end{aligned}$$

In order to compute $\Pr[V=\nu]$ we use the continuous analogon to the law of absolute probability by looking at all possible interval lengths, which yields

$$\Pr[V=\nu]=\int_0^\infty p(\nu,\lambda x)b(x)dx$$

(recall that $b(x)$ is the pdf of the service time distribution B). Now we collect all together and get

$$\begin{aligned} G_V(z) &= \sum_{i=0}^{\infty}\int_0^\infty p(i,\lambda x)b(x)dx\cdot z^i \\ &= \sum_{i=0}^{\infty}\int_0^\infty \frac{(\lambda x)^i}{i!}e^{-\lambda x}b(x)dx\cdot z^i \\ &= \int_0^\infty e^{-\lambda x}\sum_{i=0}^{\infty}\frac{(\lambda x z)^i}{i!}b(x)dx \\ &= \int_0^\infty e^{-\lambda x}e^{\lambda x z}b(x)dx \\ &= \int_0^\infty e^{-\lambda x(1-z)}b(x)dx \\ &= L_B(\lambda(1-z)) \end{aligned}$$

where we have deliberately ignored questions on whether we may change the limits or on proper convergence. So we have the result that we can express the ordinary generating function of V in terms of the Laplace transform of the service time B. Now we can plug all our results back into equation 3.9 to arrive at

$$G_N(z)=\frac{G_N(z)+(1-\rho)\cdot(z-1)}{z}\cdot L_B(\lambda(1-z))$$

which can be rearranged to find the final result

$$G_N(z)=L_B(\lambda(1-z))\cdot\frac{(1-\rho)(1-z)}{L_B(\lambda(1-z))-z} \tag{3.10}$$

which is called **Pollaczek-Khintchine Transform Equation**. The result is thus that we can express the distribution of the number of customers in its system indirectly through the Laplace transform of the service time distribution.

[2] A rough overview of Laplace transforms can be found in appendix A.2.

3.5 Two Examples

In this section we want to show two examples on the use of the Pollaczek-Khintchine Transform Equation. The first example is the M / M / 1 system and the second example an M / D / 1-queue. However, we will only quote the main intermediate results without carrying out all calculations.

3.5.1 The M/M/1 Queue

The service times are exponentially distributed with parameter μ, thus we have $B(t) = 1 - e^{-\mu t}$ and $b(t) = \mu e^{-\mu t}$, the Laplace transform of B can be evaluated to

$$L_B(s) = \frac{\mu}{s + \mu}$$

If we now compute $L_B(\lambda(1-z)) = \frac{1}{\rho(1-z)+1}$, plug this into equation 3.10 and perform some algebraic manipulation we get

$$G_N(z) = \frac{1 - \rho}{1 - \rho z}$$

It is now very easy to invert this ogf to find the probabilities $\Pr[N = k]$ if we simply use that for the geometric series we have

$$\sum_{i=0}^{\infty} x^i = \frac{1}{1 - x}$$

for $|x| < 1$. Then we can see that

$$G_N(z) = \sum_{i=0}^{\infty} (1 - \rho)\rho^i z^i$$

Since $\Pr[N = k]$ is from the definition of the ogf just the coefficient of the z^k term it is then clear that

$$\Pr[N = k] = (1 - \rho)\rho^k$$

which we already know from section 2.1.

3.5.2 The M/D/1 Queue

This example shows that the calculations are not always as neat as in the M / M / 1 case. The Laplace transform of the deterministic “distribution” is given by

$$L_B(\lambda(1 - z)) = e^{-\rho(1-z)}$$

which requires integration of the *unit impulse function*. Then we get

$$G_N(z) = (1 - \rho)(1 - z) \sum_{j=0}^{\infty} z^j e^{\rho j(1-z)}$$

Unfortunately this must be expressed in a form of $\sum_{i=0}^{\infty} b_k z^k$, where b_k does not depend on z, in order to give us the probabilities $\Pr[N = k] = b_k$. In this (lengthy) calculation the definition of the exponential series is used to finally arrive at

$$\Pr[N = k] = (1 - \rho) \sum_{i=0}^{k} e^{i\rho} (-1)^{k-i} \frac{(i\rho + k - i)(i\rho)^{k-i-1}}{(k - i)!}$$

3.6 Distribution of the Customer Response Times

This distribution can be found using the following argument: in section 3.4 we have established the following relationship between the ogf of the variable V (the number of customers arriving during a service time) and the laplace transform of the random variable B (service time duration), namely

$$G_V(z) = L_B(\lambda(1-z)) \tag{3.11}$$

However, since there is no direct relationship between the service times and the arrival process, we could have chosen any random variable X instead of B and the equation will still hold

$$G_V(z) = L_X(\lambda(1-z))$$

Now consider a newly arriving customer C that is marked upon arrival to the queue. While C experiences its response time T (waiting in the queue and getting service) new customers will arrive. For this number of new customers then we have

$$G_V(z) = L_T(\lambda(1-z))$$

Now we argue, that for this special duration (the customer response time) the variable V is in fact the same as the variable N. This is true, since the customers, that are in the system when C finishes service, are arrived during C's response time. This number is in the steady state stochastically equal to the number of customers in the system, when C has just entered, but, by PASTA, this random variable is equal to the steady state random variable for the number of customers in the system, N. Thus we have

$$G_N(z) = L_T(\lambda(1-z)) \tag{3.12}$$

If we now substitute $s = \lambda(1-z)$ and use the Pollaczek-Khintchine transform equation 3.10 we arrive at

$$L_T(s) = L_B(s)\frac{s(1-\rho)}{s-\lambda+\lambda L_B(s)} \tag{3.13}$$

As result we can note that the laplace transform of the response time can also be expressed as a function of the laplace transform of the service times. Now we have, at least in principle, all tools necessary to compute the interesting distributions (response times and number of customers in the system) and their moments (mean, variance).

Chapter 4

Queueing Networks

So far we have only looked at a single standalone queueing system. However, most real systems are better represented as a network of queues. An obvious example is the Internet, where we can model each outgoing link of each router as a single queueing system, and where an end-to-end path traverses a multitude of intermediate routers. In a queueing network a customer finishing service in a service facility A is immediately proceeding to another service facility or he is leaving the system.

One basic classification of queueing networks is the distinction between *open* and *closed* queueing networks. In an open network new customers may arrive from outside the system (coming from a conceptually infinite population) and later on leave the system. In a closed queueing network the number of customers is fixed and no customer enters or leaves the system. An arbitrary open queueing network is shown in figure 4.1. As you may notice, customer may enter the system at any queue and also may leave the system from any queue. The system may contain loopbacks and splitting points, where a customer has several possibilities for selecting the next queue. In the latter case to each possibility is often assigned a fixed probability for a customer taking this branch. A simple (?) example for an open queueing network may be the internet, where new packets arrive from "outside the system" (in fact, from the users).

As an example for a closed queueing network consider the simple central server computer model, shown in figure 4.2. There is a fixed set of tasks (in a multitasking system) and each task alternates between states where it performs some computations, thus using the processor and where it performs some I/O, e.g. access a hard disk, plot a file and so forth. After doing I/O a task continues using the processor (it is assumed that every task does just one thing at a time). Furthermore, since the operating system performs time slicing, a task looses the processor after a fixed amount of time and then again waits for the processor.

We will consider only the case of a *single class network* where all customers belong to the same class, i.e. share some important characteristics, e.g. service times. In a *multi class network* there are k classes of customers, each with different service times, paths through the network, and so forth. Customers may dynamically change their class. For some classes of customers the network may be closed, for other classes open (in this case we have a *mixed network*). Most of the material found in this chapter is from [4].

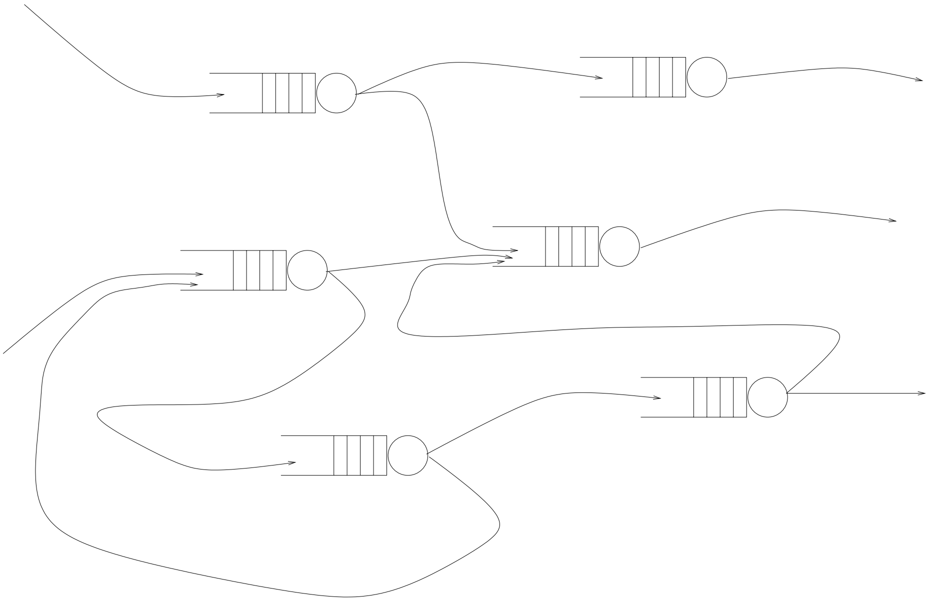

Figure 4.1: An Open Queueing Network

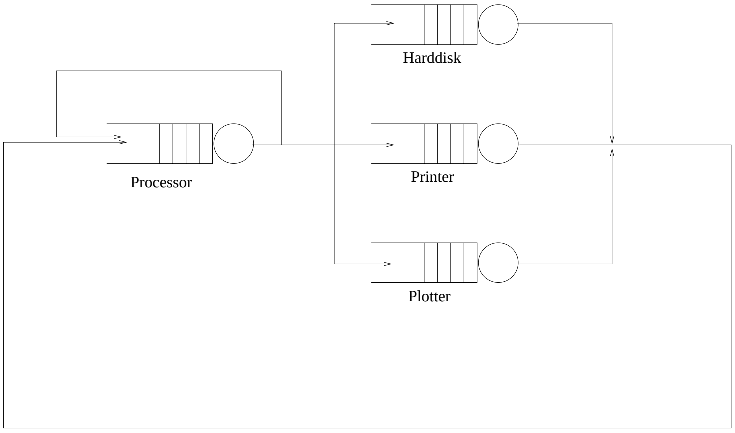

Figure 4.2: A Closed Queueing Network

4.1 Notation

- N: number of nodes (single service centers)
- K: the number of customers in a closed network
- k_i: the number of jobs at the i-th node. The nodes are numbered from 1 to N
- m_i: the number of parallel servers at node i, all servers have the same service rate
- μ_i: service rate of all the servers at node i. The overall service rate of this node is $m_i \cdot \mu_i$.
- p_{ij}: the routing probability that a customer leaving node i proceeds to node j. These probabilities remain fixed over time. Clearly, when there is no direct path from i to j we have $p_{ij} = 0$.
- p_{0j}: the probability that a new job entering the system from outside enters the system at node j. It must be true that $\sum_{j=1}^{N} p_{0j} = 1$
- p_{i0}: the probability that a job leaving the system does so from node i. $\sum_{i=1}^{N} p_{i0} = 1$
- λ_{0i} the arrival rate of jobs from outside to node i
- λ_i: the total arrival rate to node i.
- $e_i = \frac{\lambda_i}{\lambda}$ is the visit ratio of node i, i.e. how often the node is visited by a single job
- λ: the total arrival rate to all nodes in the network from outside. $\lambda = \sum_{i=1}^{N} \lambda_{0i}$

The arrival rate λ_i to node i is clearly the sum of all arrivals from the outside to i and from all nodes to i (also from i itself), thus we have

$$\lambda_i = \lambda_{0i} + \sum_{j=1}^{N} p_{ji}\lambda_j \tag{4.1}$$

where we take the output rate of node j as being equal to its arrival rate. This can be done in the case where the system has a steady-state solution. These equations are called *traffic equations* and they can be transformed into a set of N simultaneous linear equations. For the case of a closed queueing network the traffic equations reduce to

$$\lambda_i = \sum_{j=1}^{N} p_{ji}\lambda_j \tag{4.2}$$

which can also be transformed into a simultaneous set of N homogeneous linear equations. However, in this case the solution is not unique, the solution space is a subspace of dimension of at least one, since the vector $(\lambda_1, ..., \lambda_N) = (1, 1, ..., 1)$ is always a solution. Thus we can choose at least one variable λ_i free, a common setting is $\lambda_1 = 1$.

In the remainder we consider mainly fully markovian networks, unless noted otherwise. By fully markovian we mean that all arrival streams from outside are poisson streams and that all service centers transform markovian arrivals to markovian output. Some example service station types are M/M/1, M/M/m, M/M/1/N and so forth. In a markovian network the system state can be entirely characterized by the number of customers in each system, thus it is given by an N-tuple $(k_1, ..., k_N)$.

The steady-state probability of the system being in state $(k_1, ..., k_N)$ is denoted as $\pi(k_1, ..., k_N)$. The normalization condition holds, i.e. $\sum \pi(k_1, ..., k_N) = 1$

From this steady state probability we can derive the *marginal probability* $\pi_i(k)$ that the node i contains exactly k customers. This can be expressed for the case of open queueing networks as

$$\pi_i(k) = \sum_{\mathbf{k}=(k_1,...,k_N)\in\mathbb{N}^N, k_i=k} \pi(k_1, ..., k_N)$$

and for the case of closed queueing networks as

$$\pi_i(k) = \sum_{\mathbf{k}=(k_1,...,k_N),\sum_{j=1}^{N} k_j=K, k_i=k} \pi(k_1, ..., k_N) \qquad , k \leq N$$

For the marginal probabilities also the normalization condition $\sum_{k=0}^{\infty} \pi_i(k) = 1$ holds for every node i. The probability that a specific node i is busy is given by $\rho_i := 1 - \pi_i(0) = \sum_{k=1}^{\infty} \pi_i(k)$.

With these definitions we can look at each node as if it is a single node and we can evaluate the respective performance measures as described in Chapter 2.

4.2 Product Form Networks

In principle there exists a straightforward way to determine the state probabilities for a fully markovian queueing network: determine the set of all states, determine the transition rates between each pair of states, write this as an equation system for a steady state markov chain $\pi \cdot \mathbf{Q} = 0$ (eventually after flattening the N-dimensional state description into a one-dimensional) and solve for π. This sounds simple but it isn't, except for the smallest networks. The resulting equations are called *global balance equations*.

In some cases it is possible to represent the state probabilities as follows:

$$\pi(k_1, ..., k_N) = \frac{1}{G(K)} \pi_1(k_1) \cdot \pi_2(k_2) \cdot ... \cdot \pi_N(k_N) \tag{4.3}$$

where $G(K)$ is the so-called *normalization constant* (it depends on the number of customers in the system). In the case of an open queueing network we have always $G(K) = 1$, in the case of a closed queueing network $G(K)$ must be chosen such that the normalization condition $\sum \pi(k_1, ..., k_N) = 1$ holds. The equation 4.3 represents a *product form solution*. The nice thing about this equation is that we can decompose the system and look at every service center separately.

An example illustrating the power of the product form is an open network of M/M/1 queues, as given in figure 4.3 and with poisson arrivals. The solution is carried out by the following steps:

- Solve the traffic equations, thus determining for each node i its overall arrival rate λ_i
- determine the state probabilities for node i with arrival rate λ_i and service rate μ_i. These are for the M/M/1 case known to be (see section 2.1)

$$\pi_i(k) = \left(1 - \frac{\lambda_i}{\mu_i}\right) \cdot \left(\frac{\lambda_i}{\mu_i}\right)^k$$

- plug the single state probabilities for every queue together to yield

$$\begin{aligned}
\pi(k_1, k_2, k_3) &= \pi_1(k_1) \cdot \pi_2(k_2) \cdot \pi_3(k_3) \\
&= \left(1 - \frac{\lambda_1}{\mu_1}\right) \cdot \left(\frac{\lambda_1}{\mu_1}\right)^{k_1} \cdot \left(1 - \frac{\lambda_2}{\mu_2}\right) \cdot \left(\frac{\lambda_2}{\mu_2}\right)^{k_2} \cdot \left(1 - \frac{\lambda_3}{\mu_3}\right) \cdot \left(\frac{\lambda_3}{\mu_3}\right)^{k_3}
\end{aligned}$$

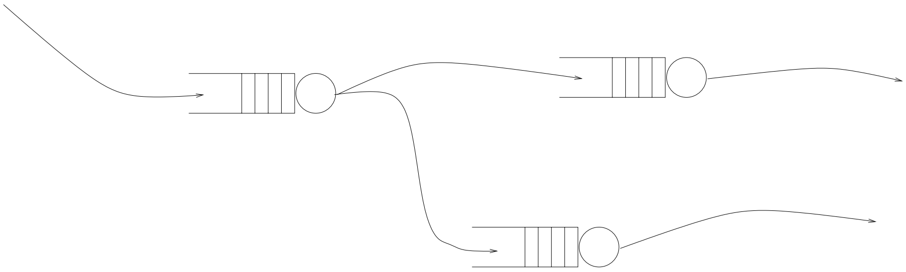

Figure 4.3: A Sample Open Network

More general, a product-form solution exists, if two conditions are met. First, all arrivals to any node must be of the poisson type. Second, each node must be of one of the following node types:

- M / M / m-FCFS
- M / G / 1-PS(RR), where the service discipline PS(RR) denotes round-robin processor sharing, as found e.g. in a multitasking operating system
- M / G / ∞ (Infinite Server)
- M / G / 1-LCFS PR (Last Come - First Served with Preemptive Resume): with this service discipline when a new customer arrives to a nonempty system, the customer currently in service is removed from the service facility, put into a LIFO queue and service starts for the new one. If the new customer has finished, the old customer re-enters service and continues at the point, where he was interrupted.

4.2.1 The Jackson Theorem for Open Queueing Networks

The theorem of Jackson specifies the condition, under which a product form solution in open queueing networks exist. These conditions are the following:

- The number of customers in the network is not limited
- Every node in the network can have poisson arrivals from outside the network
- A customer can leave the system from any node (or a subset)
- all service times are exponentially distributed
- In every node the service discipline is FCFS
- The i-th service facility consists of m_i identical servers, each with service rate μ_i (as a generalization the service rate μ_i may depend on the number of customers in system i).

Theorem. *If in an open network the condition $\lambda_i < \mu_i \cdot m_i$ holds for every $i \in \{1, .., N\}$ then the steady state probability of the network can be expressed as the product of the state probabilities of the individual nodes:*

$$\pi(k_1, ..., k_N) = \pi_1(k_1) \cdot \pi_2(k_2) \cdot ... \cdot \pi_N(k_N) \tag{4.4}$$

4.2.2 The Gordon-Newell Theorem for Closed Queueing Networks

The same assumptions hold as in the case of the Jackson theorem in subsection 4.2.1, with the exception that no customer can enter or leave the system. In this case there exists also a product-form solution, which, however has a special form. The steady-state probabilities are then given by

$$\pi(k_1, ..., k_N) = \frac{1}{G(K)} \prod_{i=1}^{N} F_i(k_i) \tag{4.5}$$

where $G(K)$ (the normalization constant) is given by

$$G(K) = \sum_{\mathbf{k}=(k_1,...,k_N), \sum_{j=1}^{N} k_j = K} \prod_{i=1}^{N} F_i(k_i) \tag{4.6}$$

and the function $F_i(k_i)$ is defined by

$$F_i(k_i) = \left(\frac{e_i}{\mu_i}\right)^{k_i} \cdot \frac{1}{\beta_i(k_i)} \tag{4.7}$$

where $e_i = \frac{\lambda_i}{\mu_i}$ and $\beta_i(k_i)$ is defined by

$$\beta_i(k_i) = \begin{cases} k_i! & : \quad k_i \leq m_i \\ m_i! \cdot m_i^{k_i - m_i} & : \quad k_i \geq m_i \\ 1 & : \quad m_i = 1 \end{cases} \tag{4.8}$$

The calculation of the normalization constant makes the treatment of closed queueing networks really awkward, since the whole state space needs to be enumerated.

4.3 Mean Value Analysis

For a larger closed product-form queueing network it is often very hard to compute the normalization constant. The *mean value analysis* provides a way to compute some main performance characteristics without the need to determine steady-state probabilities and the normalization constant. We focus here only on the case where each service center has only a single server.

We need to introduce a change in notation: in the preceding chapters we have used the letter N for the number of customers in a single queueing system, however, here N denotes the number of queues in a closed queueing network. So in this section we will denote the number of customers in a single queueing system i by M_i and its mean value by $\bar{M}_i$.

An important role in the mean value analysis plays the *arrival theorem*

Theorem. *In a closed queueing network with product-form solution the steady-state probability distribution function for the number of customers at node i when there are K customers in the network at the time instant of a customer arriving at node i is equal to the distribution function for the number of jobs at node i when there are $K-1$ jobs in the network (at any other time instant).*

Expressing this in terms of steady-state random variables $M_i(K)$ and $T_i(K)$ and taking into account, that the residual service time of the customer currently in service is by the memoryless property equal to its service time distribution B_i (the exponential distribution), we get

$$T_i(K) = B_i(1 + M_i(K-1)) \qquad , i \in \{1, .., N\} \tag{4.9}$$

Taking expectations this can be expressed as

$$\bar{T}_i(K) = \frac{1}{\mu_i}(1 + \bar{M}_i(K-1)) \qquad , i \in \{1, .., N\} \tag{4.10}$$

where $\bar{T}_i(K)$ is the mean response time of node i in a queueing network with K customers and $\bar{M}_i(K)$ is the mean number of customers in node i in a queueing network with K customers. However, this holds only for a single visit of a customer at queue i.

Next we fix node 1 and call the period between two successive departures from node 1 a *cycle* or *passage*. In a single cycle every node can be visited many times. We define the *visit count* V_i for node i implicitly by the equation

$$\lambda_i = V_i \cdot \lambda_1$$

saying that when node 1 is visited once, node i is in the mean visited V_i times during a cycle. The average response time of node i per cycle is then given by

$$\hat{T}_i(K) = \frac{V_i}{\mu_i}(1 + \bar{M}_i(K-1)) \qquad , i \in \{1, .., N\}$$

Now, if we denote the overall throughput of the network $\lambda(K)$ as the throughput at the specific node 1, Little's law gives us

$$\bar{M}_i(K) = \lambda_i(K) \cdot \bar{T}_i(K) = \lambda(K) \cdot V_i \cdot \bar{T}_i(K) = \lambda(K) \cdot \hat{T}_i(K) \tag{4.11}$$

Summing over all stations yields

$$\sum_{i=1}^{N} \bar{M}_i(K) = \lambda(K) \cdot \sum_{i=1}^{N} \hat{T}_i(K) = \lambda(K) \cdot \hat{T}(K) = \hat{M}(K) = K$$

where $\hat{T}(K)$ denotes the mean time for one cycle. So we have

$$\lambda(K) = \frac{K}{\hat{T}(K)}$$

Resubstituting the last result into equation 4.11 then yields

$$\begin{aligned} \bar{M}_i(K) &= \lambda(K) \cdot \hat{T}_i(K) \\ &= \frac{K}{\hat{T}(K)} \cdot \hat{T}_i(K) \end{aligned}$$

Now we can give an iterative algorithm for the computation of $\bar{M}_i(K)$ and $\bar{T}_i(K)$. Before we do that we summarize the needed equations as derived above, using a variable number k of customers instead of the fixed number K:

$$\begin{aligned} \bar{M}_i(k) &= V_i \cdot \lambda(k) \cdot \bar{T}_i(k) && (4.12) \\ \bar{T}_i(k) &= \frac{V_i}{\mu_i} \cdot (1 + \bar{M}_i(k-1)) && (4.13) \\ \lambda(k) &= \frac{k}{\sum_{j=1}^{N} V_j \bar{T}_j(k)} && (4.14) \end{aligned}$$

The algorithm starts with the observation that clearly for all $i \in \{1, .., N\}$ we have $\bar{M}_i(0) = 0$. Furthermore, the numbers V_i are given for $i \in \{1, .., N\}$. Then we proceed with the following steps:

1. for $i \in \{1, .., N\}$ calculate

$$\bar{T}_i(1) = \frac{V_i}{\mu_i} \cdot (1 + \bar{M}_i(0)) = \frac{V_i}{\mu_i}$$

2. Then we have all together to compute $\lambda(1)$:

$$\lambda(1) = \frac{1}{\sum_{j=1}^{N} V_j \bar{T}_j(1)} = \frac{1}{\sum_{j=1}^{N} V_j \frac{V_j}{\mu_j}}$$

3. Then we compute $\bar{M}_i(1)$ for $i \in \{1, .., N\}$ by

$$\bar{M}_i(1) = V_i \cdot \lambda(1) \cdot \bar{T}_i(1)$$

4. Now that we have given $\bar{M}_1(1), ..., \bar{M}_N(1)$ we can iterate the algorithm for $k = 2, .., K$.

Appendix A

Probability Generating Functions and Linear Transforms

In this appendix we introduce some transformations, which are widely used in probability theory in order to simplify calculations. These transformations are linear transformations. One of these transformations, the *Laplace Transform* is also widely used in electrical engineering. Its main value is that it allows the solution of linear differential equations with constant coefficients and given boundary values in a purely algebraic manner.

A.1 Probability Generating Functions

Consider a discrete random variable X which can take values from the natural numbers $0, 1, 2, 3, \ldots$. This random variable has the distribution $p_k := \Pr[X = k]$, however, we look at this as a simple sequence $(p_k)_{k \in \mathbb{N}}$ and introduce a transform from the set A of all sequences $(f_k)_{k \in \mathbb{N}}$ such that

$$\sum_{k=0}^{\infty} f_k < \infty$$

holds, to a function space. We will denote a sequence with

$$\sum_{k=0}^{\infty} f_k = 1$$

as a *probability sequence*. For a given sequence $(f_k)_{k \in \mathbb{N}} \in A$ we define the *ordinary generating function* (ogf) or *probability generating function* of f to be

$$G_f(z) = \sum_{k=0}^{\infty} f_k z^k \tag{A.1}$$

(for the moment we are not interested in convergence issues). This transformation has the following properties:

- It is linear: If $f = (f_k)_{k \in \mathbb{N}} \in A$ and $g = (g_k)_{k \in \mathbb{N}} \in A$ are two sequences and a and b are real numbers then

$$G_{af+bg}(z) = aG_f(z) + bG_g(z) \tag{A.2}$$

- We have $G_f(0) = f_0$ and for probability sequences we have $G_f(1) = 1$
- Within the convergence radius the power series converges absolutely and is an analytic function and thus we have

  $$f_n = \frac{1}{n!} \frac{d^n}{dx^n} G_f(x) \bigg|_{x=0} \tag{A.3}$$

  This shows that we can get the original sequence from its transform, however, these calculations can get a little bit clumsy if you need more than, say, five values.
- The mapping between sequences and transforms is bijective, i.e. for two sequences f and g we have $f = g$ if and only if $G_f = G_g$
- For two sequences f and g we define the *convolution sequence* h with

  $$h_i = f_0 g_i + f_1 g_{i-1} + ... + f_{i-1} g_1 + f_i g_0 \tag{A.4}$$

  Then the following equation holds:

  $$G_h(x) = G_f(x) \cdot G_g(x) \tag{A.5}$$

  Especially this is the reason why this transform is widely used in probability theory: if we have two independent discrete random variables X and Y with values from the natural numbers and we define their sum $Z = X + Y$, then in the probability distribution of Z convolution sequences arise naturally. In fact, the distribution of Z is just the convolution of the the distributions of X and Y. So, if we know the ogf of X and Y we quickly get the ogf of Z and by equation A.3 we also know the distribution itself. So if we introduce a slight change in notation we have

  $$G_Z(x) = G_{X+Y}(x) = G_X(x) \cdot G_Y(x) \tag{A.6}$$

  Thus we can say: for the sum of sequences this transform is linear, for the sum of random variables with values from the natural numbers this transform is multiplicative.

As already mentioned, in probability theory one often works not directly with distributions, instead the ogf is used, since in calculations simple algebra can be used in the transform space (think of the multiplication for the sum of two variables) whereas in the original sequence space we need to handle ugly things like convolutions. However, from time to time we need to get the sequence back from its ogf, and this can be very clumsy using equation A.3. If the ogf is a rational function (i.e. a ratio of two polynomials) then there is another way for the inverse transformation. This way uses two ingredients: a table of some basic sequence-transform pairs and partial fraction expansion. With partial fraction expansion a rational function is split into a sum of simpler rational functions, which can then be inverted more easily and the corresponding inverses may be simply added up to yield the inverse of the whole function. This is due to the linearity property.

In the most textbooks this transform is introduced together with a table of basic sequence-transform pairs.

A.2 Laplace Transform

Be $f : \mathbb{R} \to \mathbb{R}_{\geq 0}$ a function that takes nonnegative values only for $t \geq 0$. For $t < 0$ always $f(t) = 0$ holds. The Laplace transform of f is then defined as

$$L_f(s) := \int_{-\infty}^{\infty} f(t)e^{-st}dt = \int_{0}^{\infty} f(t)e^{-st}dt \qquad , s \in \mathbb{C} \tag{A.7}$$

Thus the Laplace transform maps each real valued function with real parameter to a complex valued function with complex parameter. Here we won't bother with existence of this integral. It is surely defined as long as f is almost continuous and if it grows not so fast as an exponential function. The reasons for using Laplace transforms are almost the same as for probability generating functions, but also the problem of the inverse transformation arises. In general this is a difficult problem involving complex contour integrals, but for rational laplace transforms the general technique sketched in appendix A.1 (which involves partial fraction expansion and a table of function-transform pairs) will work.

The Laplace transform has the following properties

- It is a linear transformation: if f and g are functions satisfying the above constraints and a and b are real numbers we have

$$L_{a\cdot f+b\cdot g}(s) = a \cdot L_f(s) + b \cdot L_g(s) \tag{A.8}$$

- It simplifies differentiation and integration:

$$L_{\frac{df(t)}{dt}} = s \cdot L_f(s) - f(0^-) \tag{A.9}$$

$$L_{\int_{-\infty}^{t} f(u)du} = \frac{L_f(s)}{s} \tag{A.10}$$

(where $f(0^-) = lim_{x\to 0, x<0} f(x)$). Especially these properties together with the linearity are the reason that Laplace transforms are often used for solving linear ordinary differential equations (ode) with constant coefficients, since the ode is transformed to a simple algebraic expression.

- If we define the *convolution* of two functions f and g to be

$$f(t) * g(t) = \int_{-\infty}^{\infty} f(t-x)g(x)dx \tag{A.11}$$

then for the Laplace transform we have

$$L_{f*g}(s) = L_f(s) \cdot L_g(s) \tag{A.12}$$

Again, the convolution arises when we look at the sum $Z = X + Y$ of two independent continuous random variables X and Y. We denote the Laplace transform of the *random variable* X as $L_X(s)$ and define that $L_X(s) = L_{f_X}(s)$ where $f_X(x)$ is the pdf of the random variable X. The pdf of Z is actually given by the convolution of f_X and f_Y and thus we may write

$$L_Z(s) = L_{X+Y}(s) = L_X(s) \cdot L_Y(s) \tag{A.13}$$

- Furthermore we can calculate all moments of a random variable X:

$$E[X^n] = (-1)^n \cdot \left(\left. \frac{d^n L_X(s)}{ds^n} \right|_{s=0} \right) \tag{A.14}$$

- In addition there is a funny relationship between the ordinary generating function and the Laplace transform: be f a function as defined above, and define $\{f_i : i \in \mathbb{N}, f_i := f(i)\}$ the sequence of “samples” of f taken at the discrete times $i = 0, 1, 2, 3, ...$, be $G_f(z)$ the ordinary generating function of that sequence and $L_f(s)$ the Laplace transform of f. Then it is straightforward to show

$$L_f(s) = G_f(e^{-s}) \tag{A.15}$$

A more exhaustive list of properties and a table of function-transform-pairs can be found in most of the textbooks in the bibliography.

Bibliography

[1] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions.* Dover Publications, 1965.

[2] Arnold O. Allen. *Probability, Statistics, and Queueing Theory – With Computer Science Applications.* Computer Science and Applied Mathematics. Academic Press, New York, 1978.

[3] Gunter Bolch. *Leistungsbewertung von Rechensystemen – mittels analytischer Warteschlangenmodelle.* B. G. Teubner, Stuttgart, 1989.

[4] Gunter Bolch, Stefan Greiner, Hermann de Meer, and Kishor S. Trivedi. *Queueing Networks and Markov Chains – Modeling and Performance Evaluation with Computer Science Applications.* John Wiley and Sons, New York, 1998.

[5] Christos G. Cassandras. *Discrete Event Systems – Modeling and Performance Analysis.* Aksen Associates, Boston, 1993.

[6] William Feller. *An Introduction to Probability Theory and Its Applications - Volume II.* John Wiley, New York, second edition, 1968.

[7] William Feller. *An Introduction to Probability Theory and Its Applications - Volume I.* John Wiley, New York, third edition, 1968.

[8] Boudewijn R. Haverkort. *Performance of Computer Communication Systems – A Model Based Approach.* John Wiley and Sons, Chichester / New York, 1998.

[9] Leonard Kleinrock. *Queueing Systems – Volume 1: Theory*, volume 1. John Wiley and Sons, New York, 1975.

[10] Leonard Kleinrock. *Queueing Systems – Volume 2: Computer Applications*, volume 2. John Wiley and Sons, New York, 1976.

[11] Randolph Nelson. *Probability, Stochastic Processes, and Queueing Theory – The Mathematics of Computer Performance Modeling.* Springer Verlag, New York, 1995.

[12] Thomas G. Robertazzi. *Computer Networks and Systems – Queueing Theory and Performance Evaluation.* Springer Verlag, New York, 1994.

[13] S. M. Ross. *A First Course In Probability.* Macmillan, fourth edition, 1994.

[14] Rolf Schassberger. *Warteschlangen.* Springer Verlag, Wien, 1973.

[15] William J. Stewart. *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, Princeton, New Jersey, 1994.

[16] Phuoc Tran-Gia. *Analytische Leistungsbewertung verteilter Systeme – eine Einführung*. Springer Verlag, Berlin, 1996.