A Short Introduction to Queueing Theory

AndreasWillig
Technical University Berlin, Telecommunication Networks Group
Sekr. FT 5-2, Einsteinufer 25, 10587 Berlin
email: awillig@ee.tu-berlin.de
July 21, 1999

An Excerpt of Chapter 1 & 2

Content:

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

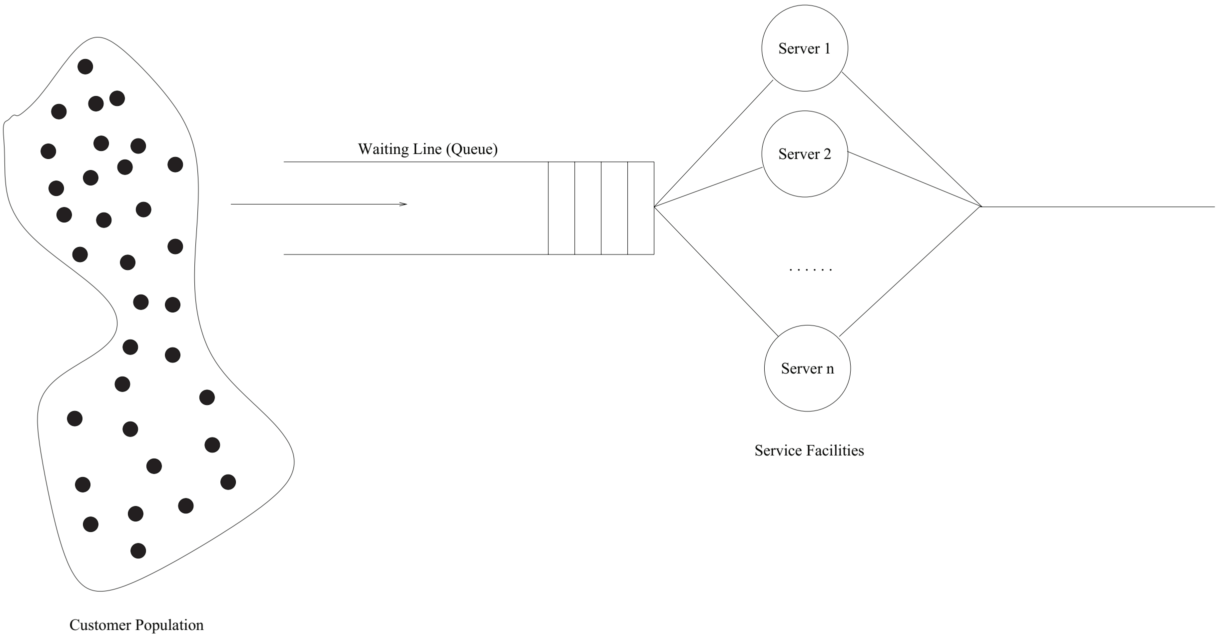

Figure 1.1: Model of a Service Center

(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

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

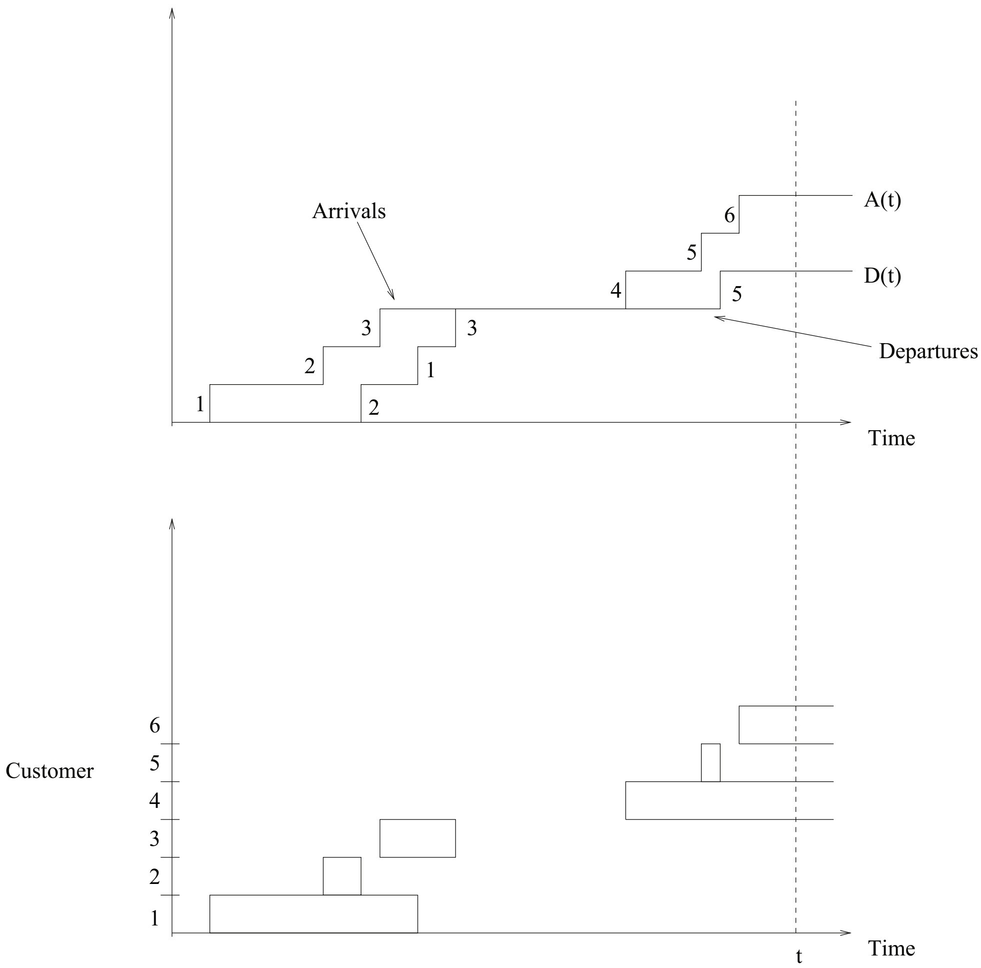

Figure 1.2: Little's Law

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.

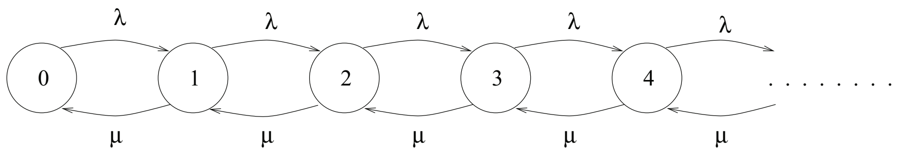

Figure 2.1: CTMC for the M/M/1 queue

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:

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

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*

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

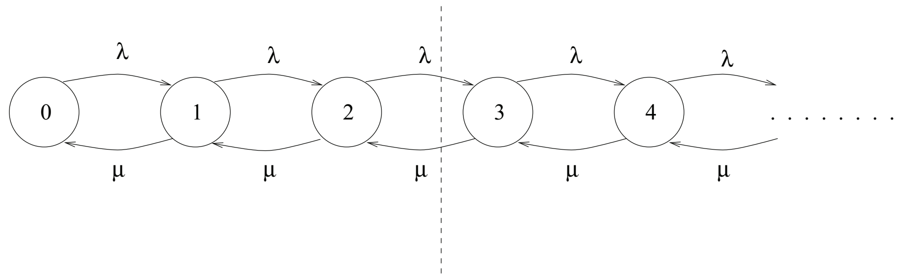

Figure 2.2: CTMC for the M/M/1 queue

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.

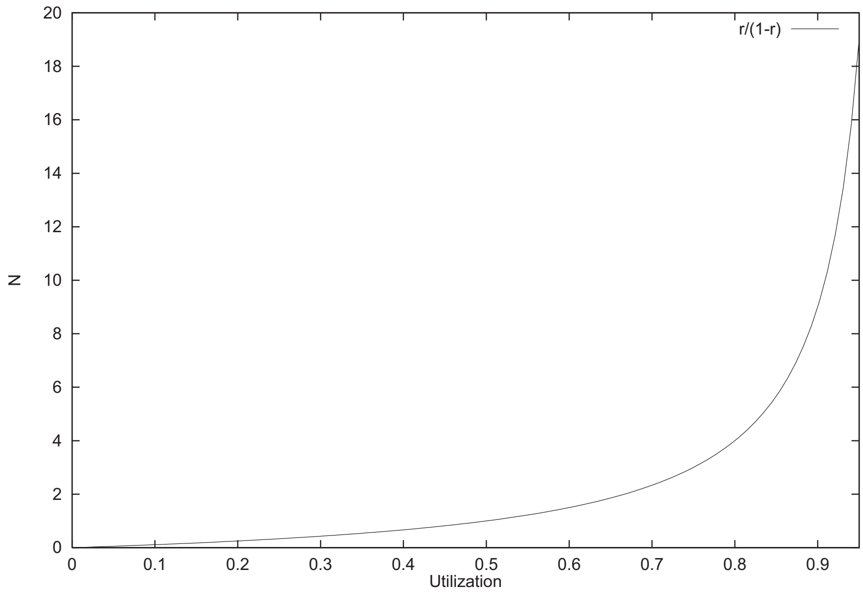

Figure 2.3: Mean Number of Customers vs. Utilization

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.

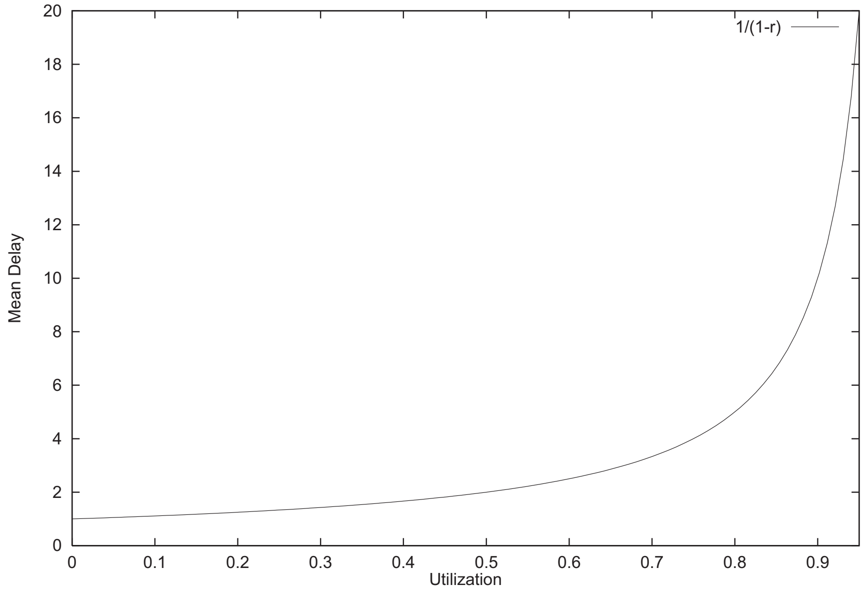

Figure 2.4: Mean Delay vs. Utilization

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

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

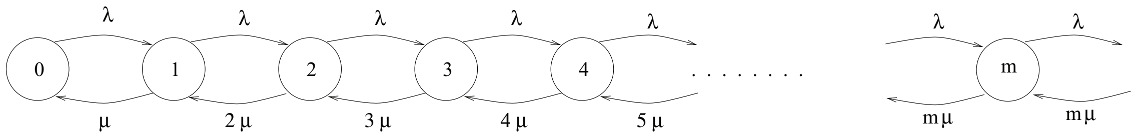

Figure 2.5: CTMC for the M/M/m queue

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:

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

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

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

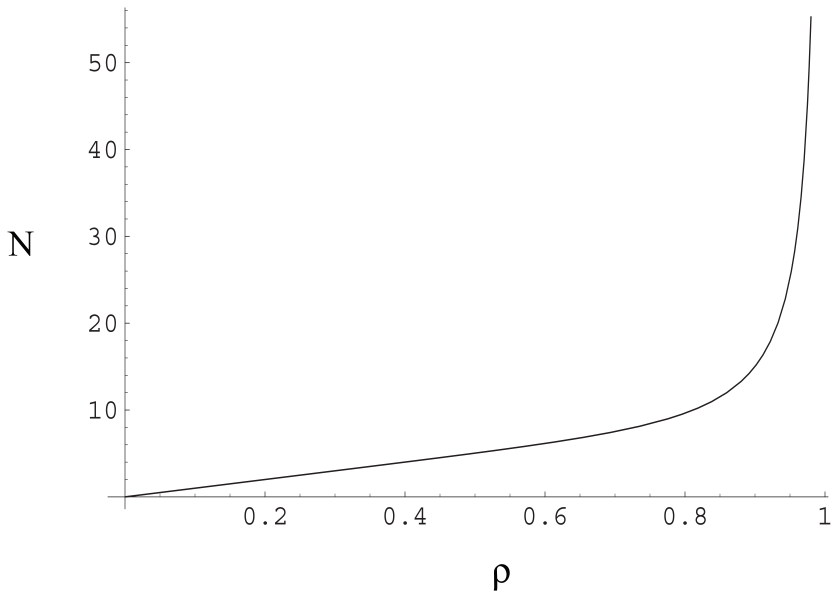

Figure 2.6: Mean Number of Customers in the system for the M/M/10-Queue

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,

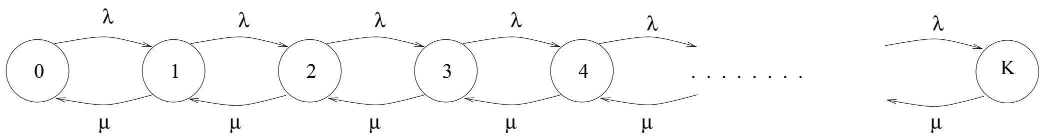

Figure 2.7: CTMC for the M/M/1/K queue

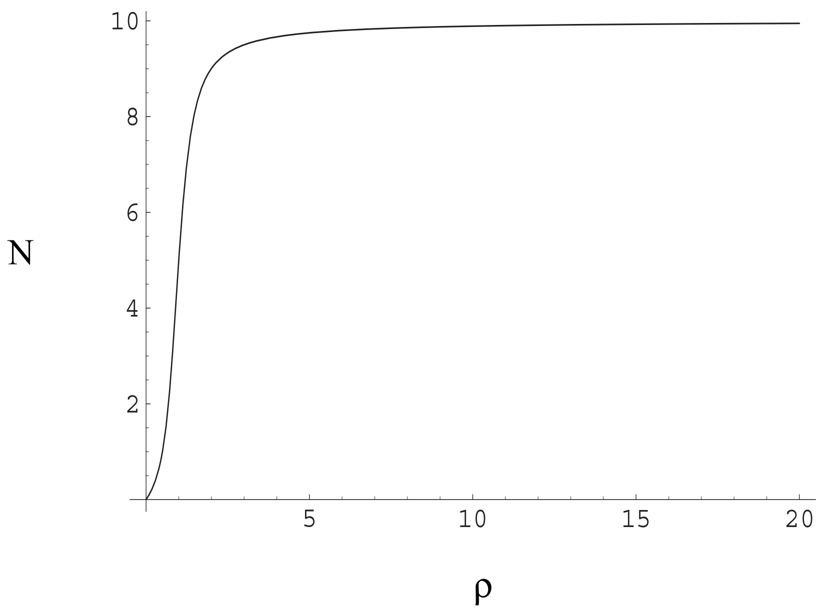

Figure 2.8: Mean number of Customers in the system for M/M/1/10-queue

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

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

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.

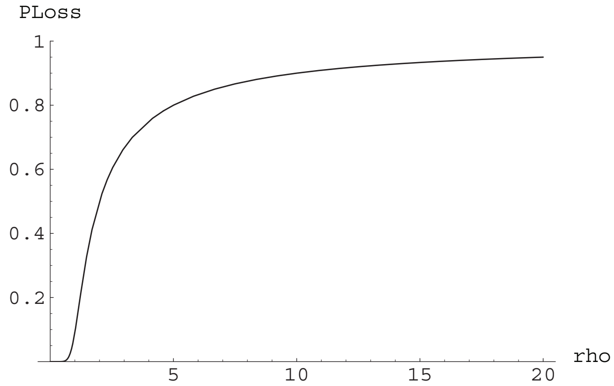

Figure 2.9: Loss Probability for M/M/1/20

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

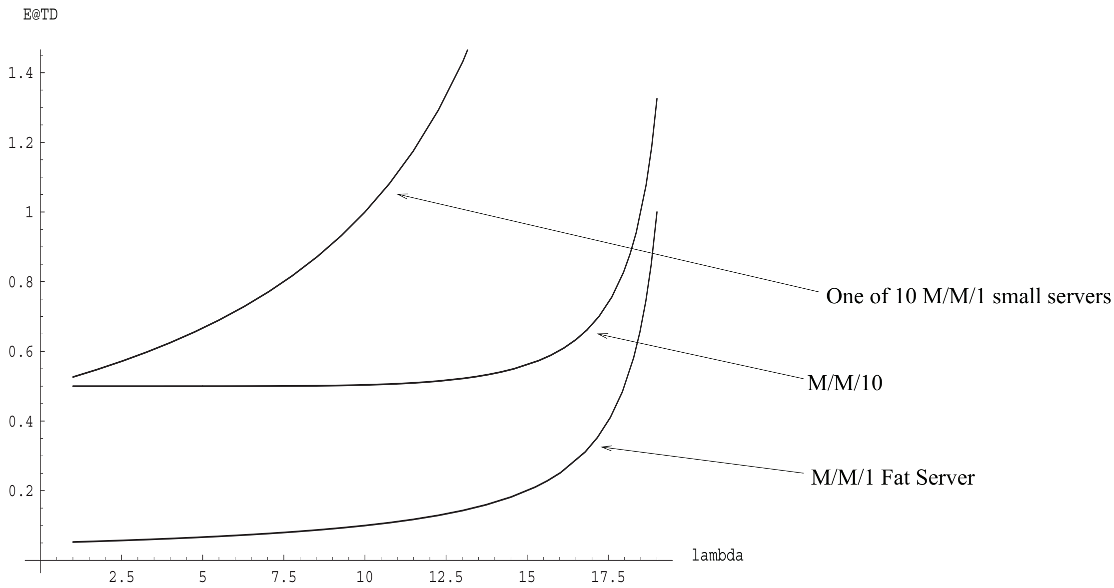

Figure 2.10: Mean Response Times for three different systems

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 $10 \cdot \mu = 20$