# Stochastic Performance Modelling

O.J. Boxma

Department of Mathematics and Computer Science
Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven, The Netherlands

January 5, 2017

# Preface

This course presents an introduction to queueing theory and its application to the performance modelling and analysis of computers and communication systems. It starts with a global introduction to the subject area (Chapter 1). Queueing theory heavily relies on concepts from probability theory. Therefore, Chapter 2 presents a reminder of some aspects of probability theory that are very useful in queueing and performance analysis, and Chapter 3 gives a brief introduction to some important stochastic processes: Markov chains and Markov processes. Chapters 4-6 are devoted to a discussion of the main ideas of queueing theory and to an analysis of the basic single-node queueing models. Networks of queues and their applications in computer-communications form the subject of Chapter 7.

*Note.* These lecture notes are partly based on the lecture notes "Stochastic Performance Modelling" of J. van der Wal, and the lecture notes "Stochastic methods for design and planning" of I.J.B.F. Adan and J.A.C. Resing.

# Chapter 1

# Queueing models and some fundamental relations

## 1.1 Introduction

In everyday life, many situations arise where customers require some service, and where the service facility has limited capacity to supply this service. If a customer cannot immediately receive service, he/she may leave or wait. In the latter case, he/she joins a *queue*. Examples of such situations occur, e.g., at:
post offices and banks (customer: person, server: clerk),
traffic lights (customer: car, server: traffic light),
production systems (customer: part, server: machine),
computer systems (customer: job, server: processor),
communication systems (customer: message or packet, server: channel, telephone line).
These applications concern in particular *design problems*, where answers are required to questions like: Is the capacity sufficient?, What should be the layout? or How do we have to divide work among several resources? To answer such questions, one would like to have detailed information about the arrival process of customers at the facility and of the required services. However, *very* detailed information is usually neither available (in particular not about future arrivals and service requests) nor manageable. In queueing theory one therefore usually assumes that the interarrival and service times satisfy some probability distribution. Obviously, one then has to be satisfied with probability statements about key *performance measures*, like the probability one has to wait or the mean waiting time.

In many applications the variability in the arrival and service process has turned out to be essential to the behavior of the system. The analysis of some key queueing models in later chapters will illustrate that.

In this chapter we describe the basic queueing model and we discuss some important fundamental relations for this model. These results can be found in every standard textbook on this topic, see e.g. [9, 15, 26, 33].

## 1.2 Queueing models and Kendall's notation

The basic queueing model is shown in figure 1.1.

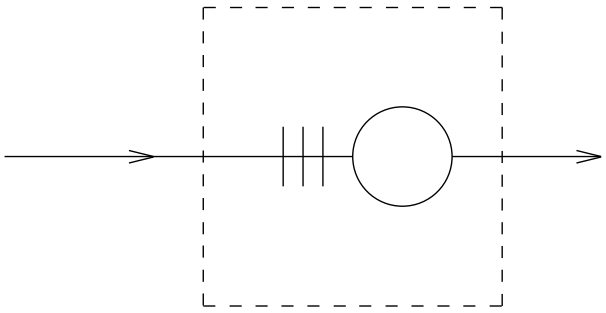

Figure 1.1: Basic queueing model

Among others, a queueing model is characterized by:

- The arrival process of customers.
  Usually we assume that the interarrival times are independent and have a common distribution. In many practical situations customers arrive according to a Poisson stream (i.e., with exponential interarrival times; see the next chapter). Customers may arrive one by one, or in batches. An example of batch arrivals is the customs office at the border where travel documents of bus passengers have to be checked.

- The behaviour of customers.
  Customers may be patient and willing to wait. Or customers may be impatient and leave after a while. For example, in call centers, customers will hang up when they have to wait too long before an operator is available, and they possibly try again after a while.

- The service times.
  Usually we assume that the service times are independent and identically distributed, and that they are independent of the interarrival times. For example, the service times can be deterministic or exponentially distributed. It can also occur that service times are dependent of the queue length. For example, the processing rates of the machines in a production system can be increased once the number of jobs waiting to be processed becomes too large.

- The service discipline.
  Customers can be served one by one or in batches. We have many possibilities for the order in which they enter service. We mention:

  – first come first served, i.e. in order of arrival;

  – random order;

  - last come first served (e.g. in a computer stack or a shunt buffer in a production line);
  - priorities (e.g. rush orders first, shortest processing time first);
  - processor sharing (in computers that equally divide their processing power over all jobs in the system).

- The service capacity.
  There may be a single server or a group of servers helping the customers.

- The waiting room.
  There can be limitations with respect to the number of customers in the system. For example, in a data communication network, only finitely many cells can be buffered in a switch. The determination of good buffer sizes is an important issue in the design of these networks.

Kendall introduced a shorthand notation to characterize a range of these queueing models. It is a three-part code $a/b/c$. The first letter specifies the interarrival time distribution and the second one the service time distribution. For example, for a general distribution the letter $G$ is used, $M$ for the exponential distribution ($M$ stands for Memoryless or Markov) and $D$ for deterministic times. The third letter specifies the number of servers. Some examples are $M/M/1$, $M/M/c$, $M/G/1$, $G/M/1$ and $M/D/1$. The notation can be extended with an extra letter to cover other queueing models. For example, a system with exponential interarrival and service times, one server and a waiting room only for $N$ customers (including the one in service) is abbreviated by the code $M/M/1/N$.

In the basic model $G/G/1$, customers arrive one by one and they are always allowed to enter the system, there is always room, there are no priority rules and customers are served in order of arrival. It will be explicitly indicated (e.g. by additional letters) when one of these assumptions does not hold.

## 1.3 Occupation rate

In a single-server system $G/G/1$ with arrival rate $\lambda$ and mean service time $E(B)$ the amount of work arriving per unit time equals $\lambda E(B)$. The server can handle 1 unit work per unit time. To avoid that the queue eventually grows to infinity, we have to require that $\lambda E(B) < 1$. Without going into details, we note that the mean queue length also explodes when $\lambda E(B) = 1$, except in the $D/D/1$ system, i.e., the system with no randomness at all.

It is common to use the notation

$$\rho = \lambda E(B).$$

If $\rho < 1$, then $\rho$ is called the *occupation rate* or *server utilization*, because it is the fraction of time the server is working.

In a multi-server system $G/G/c$ we have to require that $\lambda E(B) < c$. Here the occupation rate per server is $\rho = \lambda E(B)/c$.

## 1.4 Performance measures

Relevant performance measures in the analysis of queueing models are:

- The distribution of the waiting time and the sojourn time of a customer. The sojourn time is the waiting time plus the service time.
- The distribution of the number of customers in the system (including or excluding the one or those in service).

In particular, we are interested in mean performance measures, such as the mean waiting time, the mean sojourn time and the mean queue length.

## 1.5 Little's law

Little's law gives a very important relation between $E(L)$, the mean number of customers in the system, $E(S)$, the mean sojourn time and $\lambda$, the average number of customers entering the system per unit time. Little's law states that

$$E(L) = \lambda E(S). \tag{1.1}$$

Here it is assumed that the capacity of the system is sufficient to deal with the customers (i.e. the number of customers in the system does not grow to infinity; $\rho < 1$ should hold).

Intuitively, this result can be understood as follows. Suppose that all customers pay 1 dollar per unit time while in the system. This money can be earned in two ways. The first possibility is to let pay all customers "continuously" in time. Then the average reward earned by the system equals $E(L)$ dollar per unit time. The second possibility is to let customers pay 1 dollar per unit time for their residence in the system when they leave. In equilibrium, the average number of customers leaving the system per unit time is equal to the average number of customers entering the system. So the system earns an average reward of $\lambda E(S)$ dollar per unit time. Obviously, the system earns the same in both cases. For a rigorous proof, see [19, 30].

To demonstrate the use of Little's law we consider the basic queueing model in figure 1.1 with one server. For this model we can derive relations between several performance measures by applying Little's law to suitably defined (sub)systems. Application of Little's law to the system consisting of queue plus server yields relation (1.1). Applying Little's law to the queue (excluding the server) yields a relation between the queue length $L^q$ and the waiting time $W$, namely

$$E(L^q) = \lambda E(W).$$

Finally, when we apply Little's law to the server only, we obtain (cf. section 1.3)

$$\rho = \lambda E(B),$$

where $\rho$ is the mean number of customers at the server (which is the same as the fraction of time the server is working) and $E(B)$ the mean service time.

## 1.6 PASTA property

For queueing systems with Poisson arrivals, so for $M/\cdot/\cdot$ systems, the very special property holds that arriving customers find on average the same situation in the queueing system as an outside observer looking at the system at an arbitrary point in time. More precisely, the fraction of customers finding on arrival the system in some state $A$ is exactly the same as the fraction of time the system is in state $A$.

In general this property is not true. For instance, in a $D/D/1$ system which is empty at time 0, and with arrivals at $1, 3, 5, \ldots$ and service times 1, every arriving customer finds an empty system, whereas the fraction of time the system is empty is $1/2$.

This property of Poisson arrivals is called PASTA property, which is the acronym for Poisson Arrivals See Time Averages. Intuitively, this property can be explained by the fact that Poisson arrivals occur completely random in time (see below (2.5)). A rigorous proof of the PASTA property can be found in [37, 38].

In the following chapters we will show that in many queueing models it is possible to determine mean performance measures, such as $E(S)$ and $E(L)$, directly (i.e. not from the distribution of these measures) by using the PASTA property and Little's law. This powerful approach is called the *mean value approach*.

# Chapter 2

# Basic concepts from probability theory

This chapter is devoted to some basic concepts from probability theory, which play a key role in performance analysis.

## 2.1 Random variable

Random variables are denoted by capitals: $X$, $Y$, etc. The expected value or mean of $X$ is denoted by $E(X)$ and its variance (a measure for the deviation from the mean) by $\sigma^2(X) = E[\{X - E(X)\}^2] = E(X^2) - E^2(X)$, where $\sigma(X)$ is called the standard deviation of $X$.

An important quantity is the *coefficient of variation* of a positive random variable $X$, defined as (for $E(X) \neq 0$):

$$c_X = \frac{\sigma(X)}{E(X)}.$$

The coefficient of variation is a (dimensionless) measure of the variability of the random variable $X$.

## 2.2 Useful probability distributions

This section discusses a number of important distributions which have been found useful for describing random variables in many queueing applications.

### 2.2.1 Geometric distribution

A geometric random variable $X$ with parameter $p$ has probability distribution

$$P(X = n) = (1 - p)p^n, \qquad n = 0, 1, 2, \ldots$$

For this distribution we have

$$E(X) = \frac{p}{1-p}, \qquad \sigma^2(X) = \frac{p}{(1-p)^2}, \qquad c_X^2 = \frac{1}{p}.$$

### 2.2.2 Poisson distribution

A Poisson random variable $X$ with parameter $\mu$ has probability distribution

$$P(X = n) = \frac{\mu^n}{n!} e^{-\mu}, \qquad n = 0, 1, 2, \ldots$$

For the Poisson distribution it holds that

$$E(X) = \mu, \qquad \sigma^2(X) = \mu, \qquad c_X^2 = \frac{1}{\mu}.$$

### 2.2.3 Exponential distribution

The density of an exponential distribution with parameter $\mu$ is given by

$$f(t) = \mu e^{-\mu t}, \qquad t > 0.$$

The distribution function equals

$$F(t) = 1 - e^{-\mu t}, \qquad t \geq 0.$$

For this distribution we have

$$E(X) = \frac{1}{\mu}, \qquad \sigma^2(X) = \frac{1}{\mu^2}, \qquad c_X = 1.$$

An important property of an exponential random variable $X$ with parameter $\mu$ is the *memoryless property*. This property states that for all $x \geq 0$ and $t \geq 0$,

$$P(X > t + x | X > t) = P(X > x) = e^{-\mu x}.$$

So the remaining lifetime of $X$, given that $X$ is still alive at time $t$, is again exponentially distributed with the same mean $1/\mu$.

If $X_1, \ldots, X_n$ are independent exponential random variables with parameters $\mu_1, \ldots, \mu_n$ respectively, then $\min(X_1, \ldots, X_n)$ is again an exponential random variable with parameter $\mu_1 + \cdots + \mu_n$ and the probability that $X_i$ is the smallest one is given by $\mu_i/(\mu_1 + \cdots + \mu_n)$, $i = 1, \ldots, n$.

### 2.2.4 Erlang distribution

A random variable $X$ has an *Erlang-k* ($k = 1, 2, \ldots$) distribution with mean $k/\mu$ if $X$ is the sum of $k$ independent random variables $X_1, \ldots, X_k$ having a common exponential distribution with mean $1/\mu$. The common notation is $E_k(\mu)$ or briefly $E_k$. The density of an $E_k(\mu)$ distribution is given by

$$f(t) = \mu \frac{(\mu t)^{k-1}}{(k-1)!} e^{-\mu t}, \qquad t > 0.$$

The distribution function equals

$$F(t) = 1 - \sum_{j=0}^{k-1} \frac{(\mu t)^j}{j!} e^{-\mu t}, \qquad t \geq 0.$$

The parameter $\mu$ is called the scale parameter, $k$ is the shape parameter. A phase diagram of the $E_k$ distribution is shown in figure 2.1.

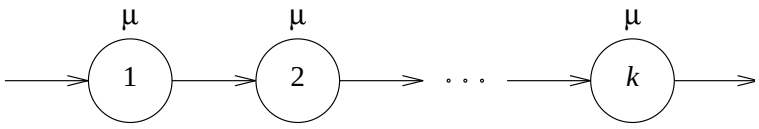


Figure 2.1: Phase diagram for the Erlang-$k$ distribution with scale parameter $\mu$

In figure 2.2 we display the density of the Erlang-$k$ distribution with mean 1 (so $\mu = k$) for various values of $k$.

The mean, variance and squared coefficient of variation are equal to

$$E(X) = \sum_{i=1}^{k} E(X_i) = \frac{k}{\mu}, \qquad \sigma^2(X) = \sum_{i=1}^{k} \sigma^2(X_i) = \frac{k}{\mu^2}, \qquad c_X^2 = \frac{1}{k}.$$

**Remark 2.1**
A convenient distribution arises when we take a mixture (weighted sum) of an $E_{k-1}$ and $E_k$ distribution with the same scale parameters. The notation used is $E_{k-1,k}$. A random variable $X$ has an $E_{k-1,k}(\mu)$ distribution, if $X$ is with probability $p$ (resp. $1-p$) the sum of $k-1$ (resp. $k$) independent exponentials with common mean $1/\mu$. The density of this distribution has the form

$$f(t) = p\mu \frac{(\mu t)^{k-2}}{(k-2)!} e^{-\mu t} + (1-p)\mu \frac{(\mu t)^{k-1}}{(k-1)!} e^{-\mu t}, \qquad t > 0,$$

where $0 \leq p \leq 1$. As $p$ runs from 0 to 1, the squared coefficient of variation of the mixed Erlang distribution varies from $1/k$ to $1/(k-1)$. It will appear (later on) that this distribution is useful for fitting a distribution if only the first two moments of a random variable are known.

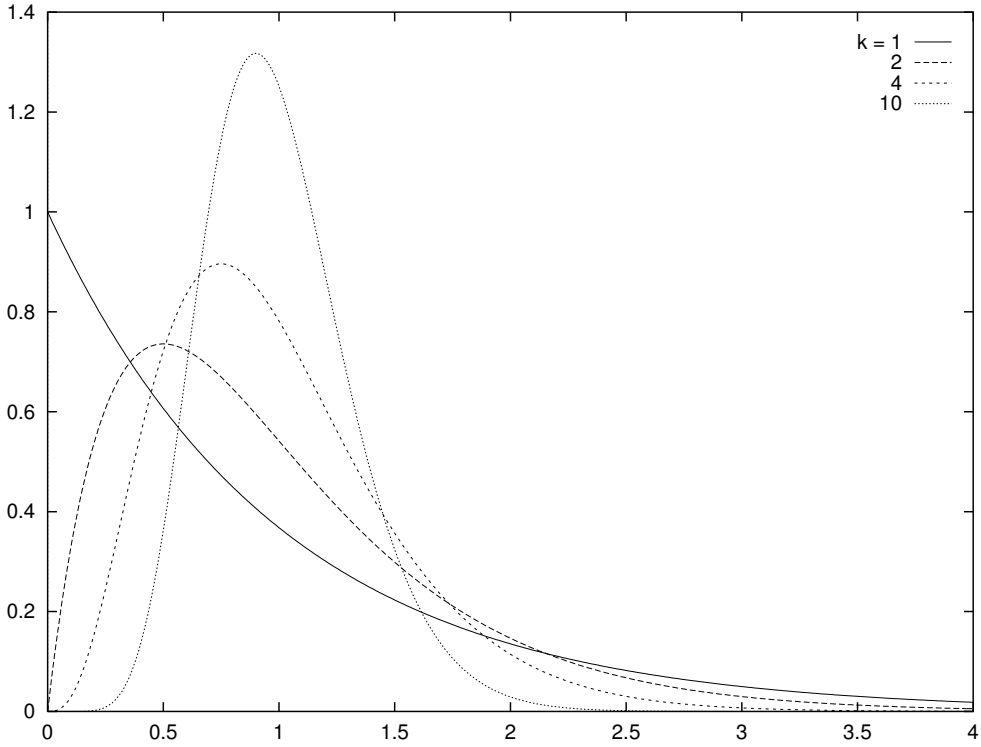


Figure 2.2: The density of the Erlang-$k$ distribution with mean 1 for various values of $k$

### 2.2.5 Hyperexponential distribution

A random variable $X$ is hyperexponentially distributed if $X$ is with probability $p_i$, $i = 1, \ldots, k$ an exponential random variable $X_i$ with mean $1/\mu_i$. For this random variable we use the notation $H_k(p_1, \ldots, p_k; \mu_1, \ldots, \mu_k)$, or simply $H_k$. The density is given by

$$f(t) = \sum_{i=1}^{k} p_i \mu_i e^{-\mu_i t}, \qquad t > 0,$$

and the mean is equal to

$$E(X) = \sum_{i=1}^{k} \frac{p_i}{\mu_i}.$$

The coefficient of variation $c_X$ of this distribution is always greater than or equal to 1. A phase diagram of the $H_k$ distribution is shown in figure 2.3.

## 2.3 Fitting distributions

In practice it often occurs that the only information of random variables that is available is their mean and standard deviation, or if one is lucky, some real data. To obtain an

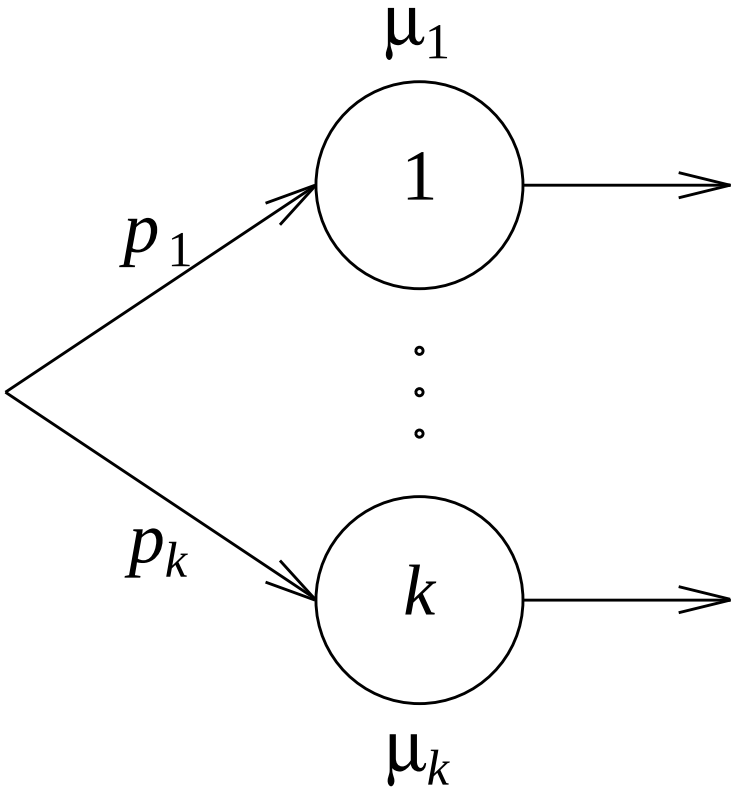


Figure 2.3: Phase diagram for the hyperexponential distribution

approximating distribution it is common to fit a phase-type distribution on the mean, $E(X)$, and the coefficient of variation, $c_X$, of a given positive random variable $X$, by using the following simple approach.

In case $0 < c_X < 1$ one fits an $E_{k-1,k}$ distribution (see subsection 2.2.4). More specifically, if

$$\frac{1}{k} \leq c_X^2 \leq \frac{1}{k-1},$$

for certain $k = 2, 3, \ldots$, then the approximating distribution is with probability $p$ (resp. $1-p$) the sum of $k-1$ (resp. $k$) independent exponentials with common mean $1/\mu$. By choosing (see e.g. [33])

$$p = \frac{1}{1+c_X^2}[kc_X^2 - \{k(1+c_X^2) - k^2c_X^2\}^{1/2}], \qquad \mu = \frac{k-p}{E(X)},$$

the $E_{k-1,k}$ distribution matches $E(X)$ and $c_X$.

In case $c_X \geq 1$ one fits a $H_2(p_1, p_2; \mu_1, \mu_2)$ distribution. However, the hyperexponential distribution is not uniquely determined by its first two moments. In applications, the $H_2$

distribution with *balanced means* is often used. This means that the normalization

$$\frac{p_1}{\mu_1} = \frac{p_2}{\mu_2}$$

is used. The parameters of the $H_2$ distribution with balanced means and fitting $E(X)$ and $c_X$ ($\geq 1$) are given by

$$p_1 = \frac{1}{2}\left(1 + \sqrt{\frac{c_X^2 - 1}{c_X^2 + 1}}\right), \qquad p_2 = 1 - p_1,$$

$$\mu_1 = \frac{2p_1}{E(X)}, \qquad \mu_2 = \frac{2p_2}{E(X)}.$$

## 2.4 Poisson process

Suppose that $Y_1, Y_2, \ldots$ are independent random variables, that are all exponentially distributed with mean $1/\lambda$, i.e., $P(Y_1 > t) = \mathrm{e}^{-\lambda t}$, $t \geq 0$. It follows from section 2.2.4 that

$$P(Y_1 + \cdots + Y_{k+1} \leq t) \;=\; 1 - \sum_{j=0}^{k} \frac{(\lambda t)^j}{j!}\mathrm{e}^{-\lambda t}, \quad t \geq 0, \quad k = 0, 1, \ldots, \tag{2.1}$$

and hence

$$P(Y_1 + \ldots Y_{k+1} > t) \;=\; \sum_{j=0}^{k} \frac{(\lambda t)^j}{j!}\mathrm{e}^{-\lambda t}, \quad t \geq 0, \quad k = 0, 1, \ldots. \tag{2.2}$$

Suppose that $Y_1, Y_2, \ldots$ are the intervals between successive occurences of events, like arrivals of customers at some service facility. Let $N(t)$ be the number of arrivals in $[0, t]$. Then $P(N(t) \leq k) = P(Y_1 + \cdots + Y_{k+1} > t)$. The probability distribution of the number of arrivals in $[0, t]$ is then a Poisson distribution:

$$P(N(t) = k) = \frac{(\lambda t)^k}{k!}\mathrm{e}^{-\lambda t}, \quad t \geq 0, \quad k = 0, 1, \ldots. \tag{2.3}$$

The process $\{N(t),\ t \geq 0\}$ that is constructed as counting the number of events in $[0, t]$ when the times between events are independent, identically $\exp(\lambda)$ distributed, is called a *Poisson process* with rate $\lambda$. The mean, variance and coefficient of variation of $N(t)$ are equal to (see subsection 2.2.2)

$$E(N(t)) = \lambda t, \qquad \sigma^2(N(t)) = \lambda t, \qquad c_{N(t)}^2 = \frac{1}{\lambda t}.$$

Using the memoryless property of the exponential distribution it follows for the Poisson process that, for small $\Delta t$,

$$P(\text{no arrival in } (t, t + \Delta t]) = \mathrm{e}^{-\lambda \Delta t} \approx 1 - \lambda \Delta t. \tag{2.4}$$

Similarly, it can be shown that

$$P(\text{one arrival in } (t, t+\Delta t]) \approx \lambda \Delta t. \tag{2.5}$$

So in each small time interval of length $\Delta t$ the occurence of an arrival is equally likely. In other words, Poisson arrivals occur completely random in time. In figure 2.4 we show a realization of a Poisson process and an arrival process with Erlang-10 interarrival times. Both processes have rate 1. The figure illustrates that Erlang arrivals are much more equally spread out over time than Poisson arrivals.

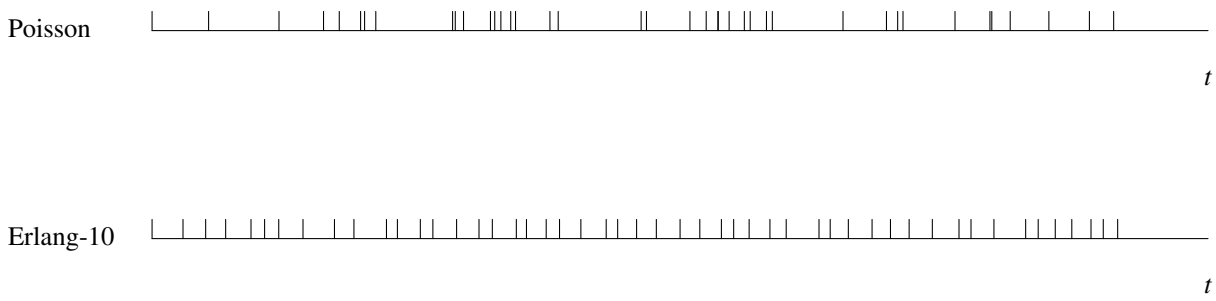


Figure 2.4: A realization of Poisson arrivals and Erlang-10 arrivals, both with rate 1

The Poisson process is an extremely useful process for modelling purposes in many practical applications, such as to model the occurence of software errors, of machine breakdowns, and of the arrival of jobs at a processor, of emails, and of orders at a production system. It is empirically found that in many circumstances the arising stochastic process can be well approximated by a Poisson process.

**Remark 2.2**
A theoretical motivation for the natural occurence of the Poisson process is the following. Let $X$ be binomially distributed with parameters $n$ and $p$.

$$P(X = k) = \left( \begin{array}{c} n \\ k \end{array} \right) p^k (1-p)^{n-k}, \quad k = 0, 1, \ldots, n. \tag{2.6}$$

Let $n \to \infty$, $p \to 0$ such that $np = \lambda t$ (fixed). Then one can prove that, in this limit,

$$P(X = k) \to \frac{(\lambda t)^k}{k!} \mathrm{e}^{-\lambda t}, \quad t \geq 0, \quad k = 0, 1, \ldots. \tag{2.7}$$

For example, if $n$ persons independently have a very small probability $p$ of submitting a job to a particular printer in a small interval $[0, t]$, then the probability of $k$ of them submitting in $[0, t]$ is given by the binomial probability in (2.6), which is closely approximated by the Poisson probability in (2.7).

Next we mention two important properties of a Poisson process (see e.g. [26]).

(i) *Merging.*
Suppose that $N_1(t)$ and $N_2(t)$ are two independent Poisson processes with respective rates $\lambda_1$ and $\lambda_2$. Then the sum $N_1(t) + N_2(t)$ of the two processes is again a Poisson process, with rate $\lambda_1 + \lambda_2$.

(ii) *Splitting.*
Suppose that $N(t)$ is a Poisson process with rate $\lambda$ and that each arrival is marked with probability $p$ independent of all other arrivals. Let $N_1(t)$ and $N_2(t)$ denote respectively the number of marked and unmarked arrivals in $[0, t]$. Then $N_1(t)$ and $N_2(t)$ are both Poisson processes with respective rates $\lambda p$ and $\lambda(1 - p)$. And these two processes are independent.

So Poisson processes remain Poisson processes under merging and splitting.

# Chapter 3

# Markov chains and Markov processes

## 3.1 Introduction

A *stochastic process* is a collection of random variables $\{X(t),\ t \in T\}$. That is, for each $t \in T$, $X(t)$ is a random variable. We often interpret $t$ as time, and call $X(t)$ the state of the process at time $t$. If $T = \{0, 1, \dots\}$, then we usually write $X_0, X_1, \dots$ and we speak of a discrete-time stochastic process. An important class of discrete-time stochastic processes is constituted by the *Markov chains*. A Markov chain is characterized by the (Markov) property that the future behaviour, given the past and present behaviour, only depends on the present and not on the past. A *Markov process* is the continuous-time analogue of a Markov chain. Many queueing models are in fact Markov processes. This chapter gives a short introduction into the theory of Markov chains and Markov processes, focussing on those characteristics that are needed for the modelling and analysis of queueing problems.

## 3.2 Markov chains

**Definition 3.1**
A discrete-time stochastic process $\{X_n,\ n = 0, 1, \dots\}$ with state space $S$ is a Markov chain if it has the following dependence structure between the successive random variables:

$$P(X_{n+1} = j | X_n = i, X_{n-1} = i_{n-1}, \dots, X_0 = i_0) \ = \ P(X_{n+1} = j | X_n = i), \tag{3.1}$$

for all $n = 0, 1, \dots$ and for all $i_0, \dots, i_{n-1}, i, j \in S$.

Usually (as in these lecture notes) it is assumed that the Markov chain has *stationary one-step transition probabilities*, viz.,

$$P(X_{n+1} = j | X_n = i) \ = \ p_{ij}, \quad i, j \in S,$$

for all $n = 0, 1, \dots$. A Markov chain, studied at the discrete time points $0, 1, 2, \dots$, is characterized by the state space $S$ and the one-step transition probabilities $p_{ij}$ between

the states. The matrix $P$ with elements $p_{ij}$ is called the *transition probability matrix* of the Markov chain. Note that the definition of the $p_{ij}$ implies that $P$ is a so-called *stochastic matrix*: the elements of $P$ are non-negative, and the row sums of $P$ are equal to 1.

**Example 3.1**
(i) Consider a communication system which transmits the digits 0 and 1. Each digit passes through several stages. At each stage there is a probability $p$ that the digit is unchanged. If $X_n$ denotes the digit entering stage $n$, then $\{X_n,\ n = 0, 1, \dots\}$ is a two-state Markov chain with transition probability matrix

$$P = \begin{pmatrix} p & 1-p \\ 1-p & p \end{pmatrix}.$$

(ii) (cf. Resnick [24]) Harry dines every night in one out of two restaurants, $A$ and $B$. His choice of restaurant on a particular night only depends on the location of the previous diner. Assuming that Harry, after having dined in $A$ ($B$), returns the next evening to $A$ ($B$) with probability 0.4 (0.7), his dining pattern follows a Markov chain with transition probability matrix

$$P = \begin{pmatrix} 0.4 & 0.6 \\ 0.3 & 0.7 \end{pmatrix}.$$

Given matrix $P$ and the initial probabilities $P(X_0 = i)$, $i \in S$, of a Markov chain $\{X_n,\ n = 0, 1, \dots\}$, one obtains $P(X_1 = j)$ for $j \in S$ via

$$P(X_1 = j) = \sum_{i \in S} P(X_1 = j, X_0 = i) = \sum_{i \in S} P(X_1 = j | X_0 = i) P(X_0 = i). \tag{3.2}$$

In vector-matrix notation, with $\pi^{(k)}$ denoting the row vector $(P(X_k = j),\ j \in S)$, $k = 0, 1, \dots$:

$$\pi^{(1)} = \pi^{(0)} P. \tag{3.3}$$

In order to study the performance characteristics of Markov chains, we should understand how the chain develops in time. Hence, we are interested in the $n$-step transition probabilities

$$p_{ij}^{(n)} \ := \ P(X_{r+n} = j | X_r = i), \quad n = 1, 2, \dots,$$

(where $p_{ij}^{(1)} = p_{ij}$), and in particular in their limiting behaviour for $n \to \infty$. The *Chapman-Kolmogorov* equations provide a method to compute the $n$-step transition probabilities from the one-step transition probabilities. These equations are:

$$p_{ij}^{(m+n)} = \sum_{k \in S} p_{ik}^{(m)} p_{kj}^{(n)}, \tag{3.4}$$

for all $i, j$ and all $m, n \geq 0$. They can be derived by manipulating conditional probabilities, determining the probability to reach $j$ from $i$ in $m+n$ steps by conditioning on the position reached after $m$ steps. Intuitively, they can be understood by observing that, once the process has reached state $k$ at time $m$, the past behaviour of the process plays no further role in the evolution of the process; the future behaviour only depends on the present state $k$. If we let $P^{(n)}$ denote the matrix of $n$-step transition probabilities, then (3.4) states that

$$P^{(m+n)} = P^{(m)} P^{(n)}; \tag{3.5}$$

we obtain $P^{(m+n)}$ from $P^{(m)}$ and $P^{(n)}$ by matrix multiplication! In particular, $P^{(2)} = P.P = P^2$, and by induction, $P^{(n)} = P^n$. Hence the $n$-step transition probability matrix can be obtained by multiplying the one-step transition probability matrix $P$ by itself $n$ times. Furthermore, the row vector $\pi^{(n)}$ with elements $\pi_j^{(n)} = P(X_n = j)$, $j \in S$, is given by

$$\pi^{(n)} = \pi^{(n-1)} P = \cdots = \pi^{(0)} P^n. \tag{3.6}$$

It is instructive to multiply the transition matrix of Harry's restaurants $2, 3, 4, \ldots$ times with itself. One then observes that $P^{(n)} = P^n$ is rapidly converging to the matrix

$$P^{(\infty)} = \begin{pmatrix} 1/3 & 2/3 \\ 1/3 & 2/3 \end{pmatrix}. \tag{3.7}$$

So, regardless where Harry started on night 0, in the long run he is in restaurant $A$ with probability $1/3$. For an important class of Markov chains, indeed a limiting distribution for $X_n$ exists when $n \to \infty$, *regardless* of the initial state $X_0$. Under the following

**Condition 3.1**

1. the Markov chain is *irreducible*,

2. the Markov chain is *aperiodic*,

3. the Markov chain is *positive recurrent*,

to be discussed below, the probability $\pi_i^{(n)}$ that the system is in state $i$ at time $n$ converges to a limit $\pi_i$ as $n$ tends to infinity, independently of the state at time 0. These limiting probabilities, or equilibrium probabilities, can be computed from a set of so-called balance equations.

The balance equations balance the probability of leaving and entering a state in equilibrium. This leads to the equations

$$\pi_i \sum_{j \neq i} p_{ij} = \sum_{j \neq i} \pi_j p_{ji} \ , \quad i \in S,$$

or (including on both sides the term $\pi_i p_{ii}$, corresponding to the one-step transition from $i$ to itself)

$$\pi_i = \sum_{j \in S} \pi_j p_{ji} \ .$$

In vector-matrix notation this becomes, with $\pi$ the row vector with elements $\pi_i$,

$$\pi = \pi P \ .$$

Together with the normalization condition

$$\sum_{i \in S} \pi_i = 1 \ ,$$

and under Condition 3.1, the solution of this equation is unique (cf. [25], p. 175).

**Remark 3.1**
If $\lim_{n \to \infty} \pi^{(n)}$ exists, then taking the limit in $\pi^{(n)} = \pi^{(n-1)} P$, cf. (3.6), indeed yields $\pi = \pi P$.

Let us now briefly discuss Condition 3.1 (for more extensive discussions of this condition, see textbooks like [24, 25, 33]).

**Discussion of Condition 3.1**
(i) **Irreducibility**
State $j$ is said to be *accessible* from state $i$ if $p_{ij}^{(n)} > 0$ for some $n \geq 0$, viz., starting from state $i$ there is a positive probability of ever reaching state $j$ (note that we define $p_{ij}^{(0)} = 1$ if $i = j$ and 0 otherwise). Two states $i$ and $j$ are said to *communicate* (to be communicating states, denoted by $i \leftrightarrow j$) if $j$ is accessible from $i$ and $i$ is accessible from $j$. The relation of communication can be shown to satisfy the following properties:
(i) $i \leftrightarrow i$,
(ii) if $i \leftrightarrow j$ then $j \leftrightarrow i$,
(iii) if $i \leftrightarrow j$ and $j \leftrightarrow k$ then $i \leftrightarrow k$.

Two states that communicate are said to be in the same class. It is seen from (i), (ii) and (iii) that any two classes are either disjoint or identical: communication is an *equivalence relation*. Hence communication breaks up the state space of a Markov chain in separate classes (equivalence classes). If all states communicate with each other, then there is only one class. We then call the Markov chain *irreducible*. The importance of the irreducibility concept w.r.t. the existence of the limiting distribution of a Markov chain can be easily understood. If there are two or more classes, then for the long-run behaviour it makes a difference in which class one starts. For example, a class may be *absorbing*, i.e., once entered, the class is never left anymore. An example occurs for a two-state Markov chain with, say, $p_{11} = p_{12} = 1/2$, $p_{21} = 0$, $p_{22} = 1$. The Markov chain is now reducible into two

classes $\{1\}$ and $\{2\}$, and state 2 is absorbing.

(ii) **Periodicity**
A state $i$ of a Markov chain is said to have *period* $d$ if $p_{ii}^{(n)} = 0$ whenever $n$ is not divisible by $d$, while $d$ is the largest integer with this property. A state with period 1 is said to be *aperiodic*. The states of a two-state Markov chain are periodic with period 2 if $p_{12} = p_{21} = 1$: starting in state $i$, that state is revisited at times $2, 4, 6, \ldots$. The importance of the concept of periodicity of a Markov chain w.r.t. the existence of its limiting distribution can be easily understood. In the above example, the limiting probability of being in state 1 depends crucially on the position of the Markov chain at time 0: $p_{ii}^{(n)} = 1$ if $n$ is even, and 0 if $n$ is odd, so no limiting distribution exists.

(iii) **Recurrence and transience**
A state is called *recurrent* if the process returns to that state with probability one. Otherwise the state is called *transient*. In the transient case, the state will only be visited a finite number of times. In the recurrent case, the state will be visited an infinite number of times (one can show that, equivalently, state $i$ is recurrent iff $\sum_{n=1}^{\infty} p_{ii}^{(n)} = \infty$; for details of this and the following statements about recurrence, see [25]). Absorbing states are obviously recurrent.

If the mean return time to a recurrent state is finite, then that state is called *positive recurrent*, otherwise *null recurrent*. If the Markov chain has only a finite number of states, then a recurrent state is always positive recurrent. Periodicity, aperiodicity, transience, positive recurrence and null recurrence all are class properties: All states of the class behave in that sense the same.

In practice one has the following types of Markov chains:
(i) one class, all states recurrent;
(ii) some transient classes and some recurrent classes; and eventually we end up in one of the recurrent classes (absorption).

In case (i) we would like to know the long-run fraction of time spent in each state. In case (ii) we would like to know the (mean) time until entering one of the recurrent classes, the probability of ending up in a particular recurrent class, as well as the long-run fraction of time spent in each of its states.

Below we first present two examples of determining the equilibrium distribution of an irreducible, aperiodic, positive recurrent Markov chain. Subsequently we give formulas for determining, when starting in a transient state, the mean time until absorption, and the probability of absorption in a particular state.

**Example 3.2**
Let us return to Example 3.1 (ii), viz., Harry's restaurants. Verify that the corresponding

Markov chain is irreducible, aperiodic and positive recurrent. Its limiting distribution is obtained by solving the set of equations

$$\pi_1 + \pi_2 = 1,$$

and either

$$\pi_1 = 0.4\pi_1 + 0.3\pi_2 \quad \text{or} \quad \pi_2 = 0.6\pi_1 + 0.7\pi_2.$$

We thus find $\pi_1 = 1/3$, $\pi_2 = 2/3$; compare with the matrix $P^{(\infty)}$ in (3.7).

**Example 3.3**
Consider the transmission of data cells in a switch in an ATM (Asynchronous Transfer Mode) network. The transfers are governed by a discrete time clock. Let us look at the following simple traffic example. At the beginning of each slot, with probability $q$ a batch of 2 cells arrives. With probability $p = 1 - q > 0$ no cell arrives. Cells arriving at the beginning of a slot can be processed in that slot. In each time slot exactly one cell can be transmitted (if there is any). Cells which cannot be transferred immediately are put into a buffer where they wait for transmission. What is the distribution of the number of packets in the buffer at the beginning of a time slot?

This system can be described by a Markov chain with states $i$, $i = 0, 1, 2, \ldots$ where $i$ denotes the number of cells in the buffer at the beginning of a time slot. All transition probabilities are 0 except $p_{i-1,i} = q$, $p_{i,i-1} = p$, $i \geq 1$ and $p_{0,0} = p$. We assume that $q < p$; otherwise at least as many cells would arrive per time unit as can be transmitted, and the Markov chain would not be positive recurrent. The Markov chain is irreducible and aperiodic (check). Let $\pi_i^{(n)}$ denote the probability that there are $i$ cells in the buffer at the beginning of time slot $n$, $n = 0, 1, 2, \ldots$. The probabilities $\pi_i^{(n)}$ satisfy

$$\pi_i^{(n+1)} = \pi_{i+1}^{(n)} p + \pi_{i-1}^{(n)} q.$$

If we let $n$ tend to infinity then the limiting probabilities $\pi_i$ satisfy the relation

$$\pi_i = \pi_{i+1} p + \pi_{i-1} q, \qquad i \geq 1,$$

and

$$\pi_0 = \pi_0 p + \pi_1 p.$$

Together with the normalizing condition that the probabilities sum up to 1 this gives the limiting distribution

$$\pi_i = (q/p)^i (1 - q/p), \qquad i \geq 0.$$

Note that this formula reflects the necessity of the condition $q < p$. A quick way to derive this formula is to realize that, for a limiting distribution to exist, there should be a balance

between the probability of moving from state $i$ to state $i+1$ and the probability of moving from state $i+1$ to state $i$ ('local balance'); hence

$$\pi_i q = \pi_{i+1} p, \qquad i = 0, 1, \ldots.$$

We end this example with the following observation. In reality buffer sizes are not infinite but finite. Suppose that the buffer offers room for at most $N$ data packets; packets arriving when the buffer is full are lost. A relevant question then is: What should be the size of the buffer such that no more than a given fraction of the packets will be lost?

**Remark 3.2**
Consider a Markov chain in which eventually absorption occurs in an absorbing class $A$. Starting in a transient state $i \in T$, the collection of all transient states, the mean time until absorption $a_i$ is given by the following set of equations:

$$a_i \;=\; 1 + \sum_{j \in T} p_{ij} a_j, \qquad i \in T. \tag{3.8}$$

This is easily seen by conditioning on the first step, starting in $i$ (where do we use the Markov property?).
And when $A$ consists of a number of absorbing states $A_1, \ldots, A_M$, then the probability of absorption in $A_j$, starting from $i \in T$, can in a similar way be shown to satisfy

$$z_{ij} \;=\; p_{ij} + \sum_{k \in T} p_{ik} z_{kj}, \qquad i \in T, \quad j = 1, \ldots, M. \tag{3.9}$$

## 3.3 Markov processes

**Definition 3.2**. A continuous-time stochastic process $\{X(t),\ t \geq 0\}$ with state space $S$ is a Markov process if

$$\begin{aligned} &\mathrm{P}(X(t+s) = j | X(s) = i,\ X(u) = x(u),\ 0 \leq u < s) = \\ &\mathrm{P}(X(t+s) = j | X(s) = i), \end{aligned}$$

for all $s, t \geq 0$, and all $i, j, x(u) \in S$, $0 \leq u < s$.

In words: The conditional distribution of $X(t+s)$, given the present $X(s)$ and the past, only depends on the present and not on the past.

We restrict ourself to Markov processes with stationary transition probabilities:

$$\mathrm{P}(X(t+s) = j | X(s) = i) \;=\; p_{ij}(t),$$

independently of $s$.

It can be shown that for this independence of the past to occur, it is necessary and sufficient that
(i) the time $T_i$ in state $i$ is $\exp(\nu_i)$ distributed, for some parameter $\nu_i$ and all $i$;

(ii) if state $i$ is left, the process moves to state $j$ with some probability $p_{ij}$ (with $\sum_{j\neq i} p_{ij} = 1$).

Property (ii) states that, at jump epochs, the Markov process behaves like a Markov chain. Property (i) states that $T_i$ has a memoryless distribution: No matter how long the process has been in state $i$, the remaining time in state $i$ is exponentially distributed with rate $\nu_i$. Notice that the exponential distribution is the only continuous distribution with the memoryless property:

$$\mathrm{P}(T_i > x + t | T_i > t) = \mathrm{P}(T_i > x).$$

Define the *transition rates* from $i$ to $j$:

$$\begin{aligned} q_{ij} &:= \nu_i p_{ij}, \quad j \neq i, \\ q_{ii} &:= -\nu_i. \end{aligned}$$

$Q = (q_{ij})_{i,j\in S}$ is called the rate matrix or generator of the Markov process. Note that the definition of $q_{ii}$ implies that $\sum_{j\neq i} q_{ij} = \nu_i$, so that all row sums of $Q$ are zero. For $j \neq i$, $q_{ij}$ really is the rate at which the process moves from $i$ to $j$: The departure rate from state $i$ is $\nu_i$, and a fraction $p_{ij}$ from the transitions out of $i$ is to $j$.

*Examples of Markov processes.*
(i) The Poisson process: take $q_{i,i+1} = \lambda$, $q_{ii} = -\lambda$, $q_{ij} = 0$ otherwise. This obviously is a transient process.
(ii) The birth-and-death process: take $q_{i,i+1} = \lambda_i$ for $i \geq 0$, $q_{i,i-1} = \mu_i$ for $i \geq 1$, $q_{ii} = -(\lambda_i + \mu_i)$ for $i \geq 0$, $q_{ij} = 0$ for $|i - j| > 1$. Only transitions to neighbouring states (births and deaths) are possible.

Kolmogorov has derived the following so-called *forward equations*: For all $i, j \in S$,

$$p'_{ij}(t) = \sum_{k\neq j} p_{ik}(t) q_{kj} - \nu_j p_{ij}(t) = \sum_{k\in S} p_{ik}(t) q_{kj},$$

or in matrix form: $p'(t) = p(t)Q$. In special cases, and given the initial condition $p(0)$, one may be able to solve Kolmogorov's set of forward equations and obtain the complete time-dependent distribution of the Markov process. An example is the Poisson process; a – difficult – exercise is to show by induction, using the Kolmogorov forward equations, that $p_{0j}(t) = \mathrm{e}^{-\lambda t}(\lambda t)^j / j!$. Usually one must be satisfied with the limiting distribution of the Markov process – to which we now turn. It is closely related to the limiting distribution of the underlying Markov chain at jump epochs.

*Limiting probabilities of the Markov process*
It can be shown that the Markov process $\{X(t),\ t \geq 0\}$ has a positive limiting distribution

$p_j := \lim_{t\to\infty} p_{ij}(t)$ if the underlying Markov chain at jump epochs has a positive limiting distribution $\{\pi_j,\ j \geq 0\}$, and then (if $\sum_{i\in S} \pi_i/\nu_i < \infty$),

$$p_j = \frac{\pi_j/\nu_j}{\sum_{i\in S} \pi_i/\nu_i}. \tag{3.10}$$

To show the latter relation, observe the following. If $p_j = \lim_{t\to\infty} p_{ij}(t)$ exists, then necessarily $p'_{ij}(t) \to 0$ as $t \to \infty$. Hence Kolmogorov's forward equations yield $0 = \sum_{k\neq j} p_k q_{kj} - \nu_j p_j$, or

$$\nu_j p_j = \sum_{k\neq j} p_k q_{kj} = \sum_{k\neq j} p_k \nu_k p_{kj}. \tag{3.11}$$

Remember that if the Markov chain with stationary transition probabilities $p_{kj}$ has a non-null limiting distribution $\pi_j$, then that distribution satisfies

$$\pi_j = \sum_{k\in S} \pi_k p_{kj} \quad \text{with} \quad \sum_{j\in S} \pi_j = 1.$$

Now (3.10) follows.

**Remark 3.3**
An interpretation of (3.11), usually called the balance equations of the Markov process, is:
the rate at which the process leaves state $j$ (the lefthand side) should equal
the rate at which the process enters state $j$ (the righthand side).
Interpretation of (3.10): In the underlying Markov chain, the process visits state $j$ a fraction $\pi_j$ of the steps. But the *mean time per visit* in $j$ is $1/\nu_j$ for the Markov process; so the Markov process spends a *fraction of time* proportional to $\pi_j/\nu_j$ in state $j$.

**Remark 3.4**
One can rewrite (3.11) in vector notation as (with 0 a vector):

$$0 = pQ. \tag{3.12}$$

Together with the normalizing condition $\sum p_i = 1$, the solution of this set of equations is known to be unique under the conditions of irreducibility and positive recurrence. In the next chapters we shall use these equations to determine the queue length distribution in some 'Markovian' queueing systems.

# Chapter 4

# Markovian queueing systems

## 4.1 Introduction

In this chapter we consider queueing systems with independent, exponentially distributed interarrival and service times. We mainly concentrate on the number of customers $X(t)$ in the system at time $t$, $t \geq 0$. Under the above exponential (memoryless) assumptions, the stochastic process $\{X(t),\ t \geq 0\}$ will be a Markov process. We shall successively discuss the $M/M/1$ queue; the $M/M/1/N$ queue with finite capacity of $N-1$ customers in the waiting buffer and 1 in service; and the multiserver queue $M/M/c$.

Throughout the chapter, the arrival rate is denoted by $\lambda$. The processing rate is denoted by $\mu$, so the mean processing time is $1/\mu$. The jobs are processed in order of arrival (FCFS = first come first served). We define

$$\rho := \frac{\lambda}{\mu}.$$

The quantity $\rho$ is the amount of work offered to the system per unit of time. As remarked in Section 2.4, the assumption of independent, exponentially distributed interarrival times (hence of a Poisson arrival process) is often a realistic one. The assumption of exponentially distributed service times is usually less realistic, but it can be viewed as a useful first approximation of reality. Although in reality a service system (processor, communication channel, machine) is never as mathematically simple as the $M/M/1$ or $M/M/c$ queue, these models contain most of the essential characteristics. Their analysis will clearly show the sometimes devastating consequences of randomness both in the arrival process and in the service times.

## 4.2 The $M/M/1$ queue

The $M/M/1$ queue models, e.g., a transmitter with rate $c$ bps where packets arrive as a Poisson process with rate $\lambda$, having independent, exponentially distributed packet lengths

with mean $1/\mu c$ bits. The successive transmission times of the packets are then exponentially distributed with mean $1/\mu$. Packets are being transmitted in order of arrival. The results of this subsection enable us to obtain the distribution of the number of packets at the transmitter and its mean, but also the distribution of the sojourn time of a packet at the transmitter and its mean. The latter distribution shows us which fraction of the packets has a sojourn time above a certain prescribed standard. The results translate immediately to many other applications. In the sequel, we shall usually speak of jobs or customers.

The assumptions we made about the system (i.e., Poisson arrivals, exponential processing times and FCFS servicing) make it possible to simply describe the state of the system at an arbitrary point in time by the number of jobs in the system. Without these assumptions, the state description would be very complicated and would have to contain not only the number of jobs in the system, but also, for example, the residual processing time of the job in service. The reason for this simplification is that, in the case of exponential interarrival times and exponential processing times, the distribution of the time until the next arrival and/or service completion is not affected by the time that elapsed since the last arrival and the last completion. This is the essential memoryless property of the exponential distribution (see Section 2.2.3). Further, the FCFS order of processing means that the past gives no information about the jobs waiting in the queue. Note that if, for instance, the processing order would be Shortest Processing Time First, then the jobs waiting in the queue will on average be longer than an arbitrary job. For the $M/M/1$ queue, the exact distribution $P(X(t) = j|X(0) = i)$ can be obtained from the theory of Markov processes, cf. [5]. In fact, the Markov process now is a birth-and-death process, with birth rate $q_{i,i+1} = \lambda_i = \lambda$ and death rate $q_{i,i-1} = \mu_i = \mu$. However, even for this relatively simple queueing system, the distribution of the time-dependent Markov process is quite complicated. Therefore we restrict ourself to the limiting distribution:

$$p_j := \lim_{t\to\infty} P(X(t) = j|X(0) = i), \qquad i, j = 0, 1, \ldots, \quad t \geq 0.$$

When does this limiting distribution exist? It is easily seen that the Markov process $\{X(t),\ t \geq 0\}$ is irreducible: all states can be reached from each other. The question remains whether the Markov process is also positive recurrent (see end of Chapter 3). One can prove (cf. [5]) that the Markov process is positive recurrent if and only if $\rho < 1$. This is intuitively obvious: every finite state $i$ ($i$ customers) will be visited infinitely often, with finite mean intervisit times, as long as the server is offered less customers per time unit than it can handle. The limiting distribution now follows from Chapter 3. The limiting probabilities $p_j$ satisfy the following equations, cf. (3.11):

$$0 = -\lambda p_0 + \mu p_1 \tag{4.1}$$

$$0 = \lambda p_{k-1} - (\lambda + \mu)p_k + \mu p_{k+1}, \quad k = 1, 2 \ldots\ , \tag{4.2}$$

and

$$\sum_{k=0}^{\infty} p_k = 1, \tag{4.3}$$

which is called the normalization equation.

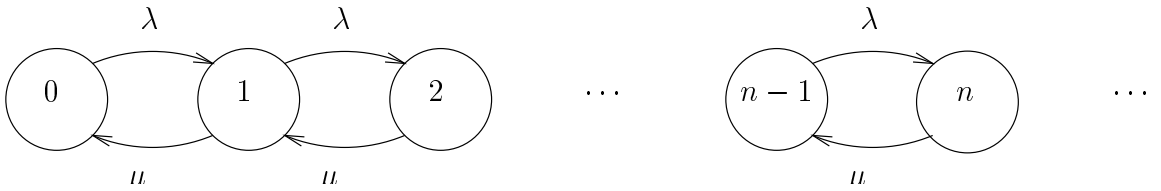


Figure 4.1: Flow diagram for the $M/M/1$ model

It is also possible to derive the equations (4.1) and (4.2) directly from a *flow diagram*, as shown in figure 4.1. The arrows indicate possible transitions. The rate at which a transition occurs is $\lambda$ for a transition from $k$ to $k+1$ (an arrival) and $\mu$ for a transition from $k+1$ to $k$ (a departure). The number of transitions per unit time from $k$ to $k+1$, which is also called the *flow* from $k$ to $k+1$, is equal to $p_k$, the fraction of time the system is in state $k$, times $\lambda$, the rate at which arrivals occur while the system is in state $k$. The equilibrium equations (4.1) and (4.2) follow by equating the flow out of state $k$ and the flow into state $k$.

**Remark 4.1**
The use of flow diagrams is quite intuitive, but also quite risky. It is absolutely necessary that the states give a *complete description* of the state of the queue length process. As we discussed above, the number of jobs in the system is usually not the full state description. If, e.g., the processing times are not exponential then the state description has to contain the remaining processing time, and thus a simple flow diagram is not available.

A further simplification of the equations can be obtained by using the typical transition structure in the system. State changes are always of the form $k$ to $k+1$ or $k$ to $k-1$. Equating the mean rate of transitions out of the set $\{0, 1, ..., k\}$ to the mean rate of transitions into that set, we get

$$p_k\lambda = p_{k+1}\mu, \quad k \geq 0. \tag{4.4}$$

We can rewrite this as

$$p_{k+1} = \rho p_k, \quad k \geq 0.$$

From this, we obtain $p_k = \rho^k p_0$ and, by setting the sum of the probabilities equal to 1,

$$p_k = (1-\rho)\rho^k, \quad k \geq 0.$$

From now on, we will call this the *equilibrium distribution* of the system.

### 4.2.1 Performance characteristics

From the equilibrium distribution we can compute the most important performance measures, such as the mean number of jobs in the system, denoted by $E(L)$, and the mean throughput time or system or sojourn time, denoted by $E(S)$.

The first one is easily obtained as

$$E(L) = \sum_{k=0}^{\infty} k p_k = \sum_{k=0}^{\infty} k(1-\rho)\rho^k = \frac{\rho}{1-\rho} \ .$$

As we see, if $\rho$, the load of the system or utilization, approaches 1 the mean number of jobs in the system goes to infinity. For example, if the load is 0.95 and the mean job size is 4 hours then the mean amount of work in the system is equal to 76 hours! This dramatic behaviour is caused by the variation in the arrival and service process and it is characteristic for almost every queueing system.

Equally, or even more important, is the mean sojourn time. From Little's formula we get

$$E(S) = E(L)/\lambda = \frac{\rho}{1-\rho}\frac{1}{\lambda} = \frac{1}{1-\rho}\frac{1}{\mu}.$$

So, we see that the behaviour of $E(S)$ is similar to the behaviour of $E(L)$, when $\rho$ approaches 1. For $\rho = 0.95$, the mean sojourn time is 20 times as big as the mean processing time (job size).

### 4.2.2 The Mean Value Approach

There is another direct way to compute the mean number of jobs in the system $E(L)$ and the mean sojourn time $E(S)$, without knowing the probabilities $p_k$. This approach uses three important properties. The first one is Little's formula (see Section 1.5), the second one is the PASTA property (see Section 1.6), and the third one is the fact that if a job has an exponentially distributed service time, then the residual service time of the job in service on an arrival instant is again exponential.

Based on the PASTA property we know that the mean number of jobs in the system seen at an arrival instant of a job equals $E(L)$. Furthermore, by the third property, each of them (also the one in service) has a (residual) service time with mean $1/\mu$. Finally, the sojourn time of a job also includes its own service time. Hence,

$$E(S) = E(L)\frac{1}{\mu} + \frac{1}{\mu} \ .$$

This relation is known as the *arrival relation*. Together with Little's formula,

$$E(L) = \lambda E(S),$$

we have two equations from which we get

$$E(S) = \lambda E(S)\frac{1}{\mu} + \frac{1}{\mu} = \rho E(S) + \frac{1}{\mu} \ .$$

Thus,

$$E(S) = \frac{1}{1-\rho}\frac{1}{\mu},$$

and

$$E(L) = \frac{\rho}{1-\rho} \ .$$

### 4.2.3 The distribution of the sojourn time

The mean value approach is, although a very powerful tool, not able to lead us to the distribution of the sojourn time. We can, however, compute this distribution from the equilibrium distribution. To do so, note that if an arriving job finds $k$ jobs in the system, then the sojourn time of this job consists of $k+1$ independent exponential service times (one of which may be a residual service time). Recall that the sum of $k+1$ independent and identically distributed service times is Erlang distributed with parameters $k+1$ and $\mu$, so with density

$$f_{k+1}(t) = \mu\frac{(\mu t)^k}{k!}e^{-\mu t} \ .$$

By PASTA, the probability that an *arriving* job finds $k$ jobs in the system is equal to $p_k$. So, we get for the overall density

$$f(t) = \sum_{k=0}^{\infty} p_k f_{k+1}(t) = \sum_{k=0}^{\infty}(1-\rho)\rho^k\mu\frac{(\mu t)^k}{k!}e^{-\mu t} = \mu(1-\rho)e^{-\mu(1-\rho)t}.$$

Hence, the sojourn time is also exponentially distributed, but with parameter $\mu(1-\rho)$.

## 4.3 The $M/M/1/N$ queue

In real life, buffers are often too small to be taken as infinite. Hence let us consider the $M/M/1/N$ queue. This is an $M/M/1$ queue with $N-1 < \infty$ waiting positions in the queue. Such a system can hold at most $N$ jobs. Arriving jobs which find the buffer full disappear (are processed elsewhere). Like in the $M/M/1$ case, $X(t)$ denotes the number of jobs or customers in the system at time $t$, and again $\{X(t),\ t \geq 0\}$ is an irreducible Markov process. All states of the Markov process are now positive recurrent, even if $\rho \geq 1$ (why?). The global balance equations (4.1) and (4.2), and also the local balance equations (4.4), remain valid, for $k < N$. The limiting distribution of the Markov process of number of customers is now given by $p_k = \rho^k p_0$ for $k = 0, \ldots, N$, with

$$p_0[\sum_{k=0}^{N}\rho^k] = 1,$$

and hence

$$p_k = \frac{1-\rho}{1-\rho^{N+1}}\rho^k, \qquad k = 0, 1, \ldots, N.$$

## 4.4 The $M/M/c$ queue

In this section we analyze the model with exponential interarrival times with mean $1/\lambda$, exponential service times with mean $1/\mu$ and $c$ parallel, identical servers. Jobs are served in order of arrival. We suppose that the occupation rate per server,

$$\rho := \frac{\lambda}{c\mu},$$

is smaller than one.

The $M/M/c$ queue is an important model from the viewpoint of applications. Its study allows us to obtain insight into the advantages of having several parallel servers (processors, or counters at the post office or supermarket). For $c = \infty$ obviously no queue arises; the $M/M/\infty$ and, more generally, the $M/G/\infty$ may be used to model, e.g., transportation along a conveyor belt.

### 4.4.1 The queue length distribution

The state of the system is completely characterized by the number of jobs in the system. Let $p_n$ denote the equilibrium probability that there are $n$ jobs in the system. Similar as for the $M/M/1$ system, we can derive the equilibrium equations for the probabilities $p_n$ from the flow diagram shown in figure 4.2.

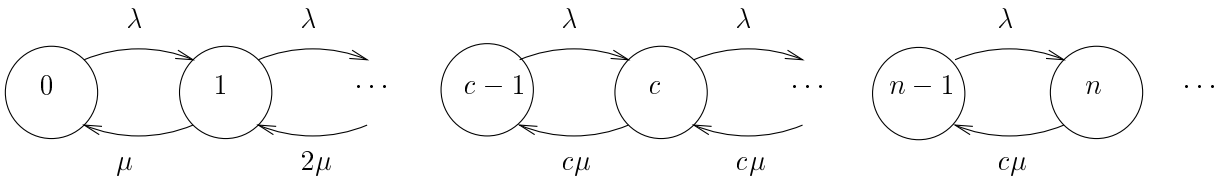


Figure 4.2: Flow diagram for the $M/M/c$ model

Instead of equating the flow into and out of a single state $n$, we get simpler equations by equating the flow out of and into the set of states $\{0, 1, \ldots, n-1\}$. This amounts to equating the flow between the two neighboring states $n-1$ and $n$ yielding

$$\lambda p_{n-1} = \min(n, c)\mu p_n, \qquad n = 1, 2, \ldots$$

Iterating gives

$$p_n = \frac{(c\rho)^n}{n!}p_0, \qquad n = 0, \ldots, c,$$

and

$$p_{c+n} = \rho^n p_c = \rho^n \frac{(c\rho)^c}{c!} p_0, \qquad n = 0, 1, 2, \ldots$$

The probability $p_0$ follows from normalization, yielding

$$p_0 = \left( \sum_{n=0}^{c-1} \frac{(c\rho)^n}{n!} + \frac{(c\rho)^c}{c!} \, \frac{1}{1-\rho} \right)^{-1}.$$

An important quantity is the probability that a job has to wait. Denote this probability by $\Pi_W$. By PASTA it follows that

$$\begin{aligned} \Pi_W &= p_c + p_{c+1} + p_{c+2} + \cdots \\ &= p_c[1 + \rho + \rho^2 + \ldots] \; = \; \frac{p_c}{1-\rho} \\ &= \frac{(c\rho)^c}{c!} \left( (1-\rho) \sum_{n=0}^{c-1} \frac{(c\rho)^n}{n!} + \frac{(c\rho)^c}{c!} \right)^{-1}. \end{aligned} \tag{4.5}$$

### 4.4.2 Mean queue length and mean waiting time

From the equilibrium probabilities we directly obtain for the mean queue length,

$$\begin{aligned} E(L^q) &= \sum_{n=0}^{\infty} n p_{c+n} \\ &= \frac{p_c}{1-\rho} \sum_{n=0}^{\infty} n(1-\rho)\rho^n \\ &= \Pi_W \sum_{n=0}^{\infty} n(1-\rho)\rho^n \\ &= \Pi_W \cdot \frac{\rho}{1-\rho} \end{aligned} \tag{4.6}$$

and then from Little's law,

$$E(W) = \Pi_W \cdot \frac{1}{1-\rho} \cdot \frac{1}{c\mu}. \tag{4.7}$$

These formulas for $E(L^q)$ and $E(W)$ can also be found by using the mean value technique: If not all servers are busy on arrival the waiting time is zero. If all servers are busy and there are zero or more jobs waiting, then a new arriving job first has to wait until the first departure and then continues to wait for as many departures as there were jobs waiting upon arrival. An interdeparture time is the minimum of $c$ exponential (residual) service times with mean $1/\mu$, and thus it is exponential with mean $1/c\mu$. So we obtain

$$E(W) = \Pi_W \frac{1}{c\mu} + E(L^q) \frac{1}{c\mu}.$$

Together with Little's law we retrieve the formulas (4.6)–(4.7). Table 4.1 lists the waiting probability $\Pi_W$ and the mean waiting time $E(W)$ in an $M/M/c$ with mean service time 1 for $\rho = 0.9$.

| $c$ | $\Pi_W$ | $E(W)$ |
|---|---|---|
| 1 | 0.90 | 9.00 |
| 2 | 0.85 | 4.26 |
| 5 | 0.76 | 1.53 |
| 10 | 0.67 | 0.67 |
| 20 | 0.55 | 0.28 |

Table 4.1: Performance characteristics for the $M/M/c$ queue with $\mu = 1$ and $\rho = 0.9$

We see that the waiting probability slowly decreases as $c$ increases. The mean waiting time however decreases fast (a little faster than $1/c$). One can also look somewhat differently at the performance of the system. We do not look at the occupation rate of a server, but at the average number of idle servers. Let us call this the surplus capacity. Table 4.2 shows for fixed surplus capacity (instead of for fixed occupation rate as in the previous table) and $c$ varying from 1 to 20 the mean waiting time and the mean number of customers in the system.

| $c$ | $\rho$ | $E(W)$ | $E(L)$ |
|---|---|---|---|
| 1 | 0.90 | 9.00 | 9 |
| 2 | 0.95 | 9.26 | 19 |
| 5 | 0.98 | 9.50 | 51 |
| 10 | 0.99 | 9.64 | 105 |
| 20 | 0.995 | 9.74 | 214 |

Table 4.2: Performance characteristics for fixed surplus capacity of 0.1 server

Although the mean number of jobs in the system sharply increases, the mean waiting time remains nearly constant. It is clear that a $c$-server system is more efficient than $c$ separate single-server queues: in the latter system, some servers may be idle while there are waiting customers at other queues.

### 4.4.3 Distribution of the waiting time

The derivation of the distribution of the waiting time is very similar to the one in section 4.2.3 for the $M/M/1$ queue. By conditioning on the state seen on arrival we obtain

$$P(W > t) = \sum_{n=0}^{\infty} P(\sum_{k=1}^{n+1} D_k > t) p_{c+n},$$

where $D_k$ is the $k$th interdeparture time. Clearly, the random variables $D_k$ are independent and exponentially distributed with mean $1/c\mu$. Hence, we find

$$\begin{aligned} P(W > t) &= \sum_{n=0}^{\infty} \sum_{k=0}^{n} \frac{(c\mu t)^k}{k!} e^{-c\mu t} p_{c+n} \\ &= \sum_{k=0}^{\infty} \sum_{n=k}^{\infty} \frac{(c\mu t)^k}{k!} e^{-c\mu t} p_c \rho^n \\ &= \frac{p_c}{1-\rho} \sum_{k=0}^{\infty} \frac{(c\mu\rho t)^k}{k!} e^{-c\mu t} \\ &= \Pi_W e^{-c\mu(1-\rho)t}, \qquad t \geq 0. \end{aligned}$$

This yields for the conditional waiting time,

$$P(W > t | W > 0) = \frac{P(W > t)}{P(W > 0)} = e^{-c\mu(1-\rho)t}, \qquad t \geq 0.$$

Hence, the conditional waiting time $W|W > 0$ is exponentially distributed with parameter $c\mu(1-\rho)$.

# Chapter 5

# The $M/G/1$ system

## 5.1 Introduction

In the previous chapter we considered queueing systems with Poisson arrivals, which is in many cases a fairly realistic model for the arrival process, and with exponentially distributed service times. In practice exponentially distributed service times are not very common. Therefore it is important to extend the theory to the case of more generally distributed service times.
In this chapter we will study the case of Poisson arrivals and generally distributed, though independent, service times. So we will be looking at the $M/G/1$ system. The service order is again FCFS. We use the same notation as before; in particular, $\rho = \lambda E(B)$ denotes the traffic load: the product of the arrival rate and the mean service time. One can show that, like for the $M/M/1$ queue, $\rho$ is the probability that in equilibrium there is at least one customer present.

## 5.2 The mean value approach

Performance measures like the mean waiting time and the mean number of jobs in the queue can be, similar to the $M/M/1$ queue, obtained by the mean value approach. First, let us derive the arrival relation. A new arriving job first has to wait for the *residual processing time* of the job in service (if there is one) and then continues to wait for the processing of all jobs which are already waiting in the queue on arrival. By PASTA we know that with probability $\rho$ the machine is working on arrival. Let the random variable $R$ denote the residual service time and let $L^q$ denote the number of jobs waiting in the queue. Hence,

$$E(W) = E(L^q)E(B) + \rho E(R).$$

Furthermore, we get by Little's law (applied to the queue),

$$E(L^q) = \lambda E(W).$$

Combining these two relations, we find

$$E(W) = \frac{\rho E(R)}{1-\rho}\,. \tag{5.1}$$

Formula (5.1) is commonly referred to as the Pollaczek-Khinchin mean value formula. It remains to calculate the mean residual service time. In the following section we will show that

$$E(R) = \frac{E(B^2)}{2E(B)}\,, \tag{5.2}$$

which may also be written in the form

$$E(R) = \frac{E(B^2)}{2E(B)} = \frac{\sigma_B^2 + E(B)^2}{2E(B)} = \frac{1}{2}(c_B^2+1)E(B). \tag{5.3}$$

An important observation is that, clearly, the mean waiting time depends on the service time distribution only through its first two moments. So in practice it is sufficient to know the mean and standard deviation of the service time in order to estimate the mean waiting time and mean queue length:

$$E(W) = \frac{\lambda E(B^2)}{2(1-\rho)}, \qquad E(L^q) = \frac{\lambda^2 E(B^2)}{2(1-\rho)}. \tag{5.4}$$

Once we know $E(W)$ and $E(L^q)$, expressions for $E(S)$ and $E(L)$ follow of course using $E(S) = E(W) + E(B)$ and $E(L) = E(L^q) + \rho$.

**Example 5.2.1** *(Exponential service times)*
For exponential service times we have $E(R) = E(B)$ (memoryless property!), which indeed correponds to (5.3). So, in this case the expressions for the mean performance measures simplify to

$$E(W) = \frac{\rho}{1-\rho}E(B), \quad E(L^q) = \frac{\rho^2}{1-\rho}, \quad E(S) = \frac{1}{1-\rho}E(B), \quad E(L) = \frac{\rho}{1-\rho}.$$

**Example 5.2.2** *(Deterministic service times)*
For deterministic service times we have $c_B^2 = 0$ and hence $E(R) = E(B)/2$. In this case we have

$$\begin{aligned} E(W) &= \frac{\rho}{1-\rho}\frac{E(B)}{2}, \quad E(L^q) = \frac{\rho^2}{2(1-\rho)}, \\ E(S) &= \frac{\rho}{1-\rho}\frac{E(B)}{2} + E(B), \quad E(L) = \rho + \frac{\rho^2}{2(1-\rho)}. \end{aligned}$$

## 5.3 Residual service time

Suppose that a job arrives when the server is working and denote the total service time of the job in service by $Y$. Further let $f_Y(\cdot)$ denote the density of $Y$. The basic observation to find $f_Y(\cdot)$ is that it is more likely that a job arrives during a long service time than during a short one. So the probability that $Y$ is of length $x$ should be proportional to the length $x$ as well as the frequency of such service times, which is denoted by $f_B(x)dx$. Thus we may write

$$P(x \leq Y \leq x + dx) = f_Y(x)dx = Cxf_B(x)dx,$$

where $C$ is a constant to normalize this density. So

$$C^{-1} = \int_0^\infty x f_B(x)dx = E(B).$$

Hence

$$f_Y(x) = \frac{x f_B(x)}{E(B)}\,.$$

We conclude that

$$E(Y) = \int_0^\infty x f_Y(x)dx = \frac{1}{E(B)} \int_0^\infty x^2 f_B(x)dx = \frac{E(B^2)}{E(B)}\,.$$

Because the arrival point of the job will be a random point within this service time $Y$, we conclude that the mean residual service time is given by

$$E(R) = \frac{E(Y)}{2} = \frac{E(B^2)}{2E(B)}\,.$$

**Example 5.3.1** *(Erlang service times)*
For an Erlang-$r$ service time with mean $r/\mu$ we have

$$E(B) = \frac{r}{\mu}\,, \qquad \sigma^2(B) = \frac{r}{\mu^2}\,,$$

so

$$E(B^2) = \sigma^2(B) + (E(B))^2 = \frac{r(1+r)}{\mu^2}\,.$$

Hence

$$E(R) = \frac{1+r}{2\mu}.$$

## 5.4 The busy period

An important performance measure of queueing systems is the *busy period*, a period during which the server is uninterruptedly busy. We denote the length of a busy period by $P$. Each busy period is followed by an *idle period* in which the server is idle. A busy period and the subsequent idle period together constitute a *busy cycle*. In this section we indicate two ways to determine the mean $E(P)$ of the length of a busy period in the $M/G/1$ queue.

*Method 1.*
The server is busy during a fraction $\rho$ of the time, and hence

$$\frac{E(P)}{E(C)} = \frac{E(P)}{E(P)+E(I)} = \rho. \tag{5.5}$$

The memoryless property of the interarrival times implies that $E(I) = 1/\lambda$. Hence it follows that

$$E(P) = \frac{E(B)}{1-\rho}. \tag{5.6}$$

*Method 2.*
A busy period consists of its first service plus the service of all the work that enters while the busy period is ongoing. The mean of the latter term equals $\rho E(P)$, since $\rho$ work arrives per unit of time. (why is it important to have Poisson arrivals here?) Hence

$$E(P) = E(B) + \rho E(P), \tag{5.7}$$

again leading to (5.6). It is at first sight paradoxal, that the mean length of a whole busy period in the $M/G/1$ queue can be *smaller* than the mean sojourn time of one customer (for example, in the $M/M/1$ case they are equal; in the $M/H_k/1$ queue, $E(P)$ is indeed smaller). Can you explain this paradox?

## 5.5 The queue length distribution

In this section we shall consider the equilibrium distribution of the number of customers in the system just before arrival epochs, immediately after departure epochs, and at arbitrary points in time. Let $X_{a,n}$ be the number of customers found by the $n$-th arriving customer after $t = 0$, let $X_{d,n}$ be the number of customers that is left behind by this customer upon departure, and let $X(t)$ be the number of customers in the system at time $t$. $X_a$, $X_d$ and $X$ have the corresponding equilibrium distributions. We indicate how the equilibrium distribution $\mathrm{P}(X_d = j)$ can be obtained; subsequently we argue that $\mathrm{P}(X = j) = \mathrm{P}(X_a = j) = \mathrm{P}(X_d = j)$.

Let $A_n$ be the number of arrivals during the service of the $n$-th customer. The following recursion holds:

$$\begin{aligned} X_{d,n+1} &= X_{d,n} - 1 + A_{n+1} \text{ if } X_{d,n} > 0, \\ &= A_{n+1} \text{ if } X_{d,n} = 0. \end{aligned}$$

Verify that $\{X_{d,n}, n = 1, 2, \dots\}$ is a Markov chain. The transition probabilities $p_{ij}$ of this Markov chain are (to make the argument slightly easier, we assume that the density $b(\cdot)$ of the service time distribution exists): for $j = 0, 1, \dots,$

$$p_{0j} = \mathrm{P}(A_n = j) = \int_0^\infty \mathrm{e}^{-\lambda t} \frac{(\lambda t)^j}{j!} b(t) \mathrm{d}t,$$

and for $i = 1, 2, \dots,$ $j = i - 1, i, \dots$:

$$p_{ij} = \mathrm{P}(A_n = j - i + 1) = \int_0^\infty \mathrm{e}^{-\lambda t} \frac{(\lambda t)^{j-i+1}}{(j-i+1)!} b(t) \mathrm{d}t,$$

while $p_{ij} = 0$ for $j = 0, 1, \dots, i - 2$.

**Example 5.5.1** *(The $M/M/1$ queue)*
In this case, $b(t) = \mu \mathrm{e}^{-\mu t}$. Check that $p_{0j} = (\frac{\lambda}{\lambda+\mu})^j \frac{\mu}{\lambda+\mu}$. This can be proven by observing that

$$\begin{aligned} p_{0j} &= \int_0^\infty \mathrm{e}^{-\lambda t} \frac{(\lambda t)^j}{j!} \mu \mathrm{e}^{-\mu t} \mathrm{d}t \\ &= (\frac{\lambda}{\lambda+\mu})^j \frac{\mu}{\lambda+\mu} \int_0^\infty \mathrm{e}^{-(\lambda+\mu)t} \frac{((\lambda+\mu)t)^j}{j!} (\lambda+\mu) \mathrm{d}t, \end{aligned}$$

and noticing that the last integrand is exactly the density of the Erlang-$j$ distribution with parameter $\lambda + \mu$. That the last integral equals one can also be verified via partial integration.
Another method to prove that $p_{0j} = (\frac{\lambda}{\lambda+\mu})^j \frac{\mu}{\lambda+\mu}$ is the following. The arrival intervals and service times are exponentially distributed (memoryless). The probability that an arrival interval is shorter than a service time equals $\frac{\lambda}{\lambda+\mu}$. The memoryless property of the service time distribution implies that the probability that the first $j$ arrivals all take place before the end of that same service equals $(\frac{\lambda}{\lambda+\mu})^j$.

The equilibrium distribution $\mathrm{P}(X_d = j)$ can be obtained from the above recurrence relations between $X_{d,n+1}$ and $X_{d,n}$ via generating functions. We give the result without proof. Let $A(z) = \sum_{j=0}^\infty \mathrm{P}(A_n = j) z^j$:

$$\begin{aligned} A(z) &= \int_0^\infty \mathrm{e}^{-\lambda t} b(t) \sum_{j=0}^\infty \frac{(\lambda t z)^j}{j!} \mathrm{d}t \\ &= \int_0^\infty \mathrm{e}^{-\lambda t(1-z)} b(t) \mathrm{d}t. \end{aligned}$$

It can be shown that

$$\sum_{j=0}^{\infty} \mathrm{P}(X_d = j) z^j = \frac{(1-\rho)(1-z)A(z)}{A(z)-z}.$$

By differentiation of this expression w.r.t. $z$ and substitution of $z = 1$, it follows after some calculation that

$$\mathrm{E}X_d = \rho + \frac{\lambda^2((E(B))^2 + \sigma^2(B))}{2(1-\rho)} = \rho + \frac{\lambda^2 E(B^2)}{2(1-\rho)},$$

which is the same value as we found for $E(L) = \rho + E(L^q)$ in (5.4). For given service time distribution $\mathrm{P}(X_d = j)$ can be determined from the generating function.

**Example 5.5.2** *(The $M/M/1$ queue)*
Now

$$A(z) = \int_0^{\infty} \mathrm{e}^{-\lambda t(1-z)} \mu \mathrm{e}^{-\mu t} \mathrm{d}t = \frac{\mu}{\mu + \lambda(1-z)}.$$

Hence

$$\sum_{j=0}^{\infty} \mathrm{P}(X_d = j) z^j = \frac{(1-\rho)(1-z)\mu}{\mu - z(\mu + \lambda(1-z))} = \frac{1-\rho}{1-\rho z},$$

so that (geometric sum):

$$\mathrm{P}(X_d = j) = (1-\rho)\rho^j, \quad j = 0, 1, \ldots.$$

Now that we have obtained the (generating function of the) probability distribution $\mathrm{P}(X_d = j)$, we shall indicate why that distribution equals the distribution of the numbers of customers just before arrivals and at arbitrary moments in time. That $\mathrm{P}(X_a = j) \equiv \mathrm{P}(X_d = j)$ follows from the fact that in equilibrium the number of customers changes just as many times from $j$ to $j+1$ as from $j+1$ to $j$ (this equality holds much more generally, for queueing systems in which customers arrive individually and are served individually). That $\mathrm{P}(X_a = j) \equiv \mathrm{P}(X = j)$ follows from PASTA: arriving customers see the system in steady state.

**Remark 5.1**
For the $G/G/1$ system with general interarrival times and arbitrary processing times only approximations exist. The simplest approximation for the mean waiting time assumes that the randomness of the interarrival process has more or less the same effect on the mean waiting time as the randomness in the service times. Denoting the coefficient of variation of the interarrival times by $c_A$ and the coefficient of variation of the processing times by $c_B$, the approximation is given by

$$E(W) \approx \frac{\rho}{1-\rho} \cdot \frac{c_A^2 + c_B^2}{2} \cdot E(B). \tag{5.8}$$

Other approximations that are proposed in the literature are

$$E(W) \approx \frac{\rho}{1-\rho} \cdot \frac{(1+c_B^2)\,(c_A^2+\rho^2 c_B^2)}{2\,(1+\rho^2 c_B^2)} \cdot E(B) \tag{5.9}$$

and

$$E(W) \approx \frac{\rho}{1-\rho} \cdot \frac{(1+c_B^2)\;((2-\rho)c_A^2+\rho c_B^2)}{2\,(2-\rho+\rho c_B^2)} \cdot E(B) \tag{5.10}$$

Remark that all three approximations are exact for the case that the arrival process is a Poisson process, i.e., $c_A^2 = 1$ (cf. (5.1) and (5.3)). In the next table we compare the approximations with exact results in the case of Erlang distributed interarrival and processing times. We have chosen $\rho = 0.9$, $E(B) = 1$, $c_B^2 = 1/4$ and $c_A^2 = 1/4$, $1/3$ and $1/2$, respectively.

| | $c_A^2$ | | |
|---|---|---|---|
| | 1/4 | 1/3 | 1/2 |
| (5.8) | 2.250 | 2.625 | 3.375 |
| (5.9) | 2.117 | 2.507 | 3.287 |
| (5.10) | 2.122 | 2.512 | 3.290 |
| exact | 2.076 | 2.466 | 3.250 |

Table 5.1: Comparison of the approximations of the mean waiting times with exact results

# Chapter 6

# The $M/G/1$ priority system

## 6.1 Introduction

Usually not all jobs have the same urgency. In production systems, some jobs are supposed to be ready within a day or a week, while other jobs have a delivery date of 4 to 6 weeks after the placement of the order. Furthermore, some customers may be regular ones with contracts specifying short sojourn times, others are occasional and receive a delivery date according to the present amount of work in process. And in a Real Time Database (RTDB) system that is consulted from many different sites, it is also natural to have priorities (e.g., give write jobs priority over read jobs). In communication networks, there are various types of traffic with very different characteristics to which, accordingly, often different priorities must be assigned. For example, one may wish to distinguish between data-, voice-, and video traffic.

These circumstances make it quite natural to study a simplified queueing system in which arriving jobs belong to different job classes and in which the job classes have different priorities. If we number the priority classes from 1 upto $r$, then class 1 is top priority, class 2 has the second highest priority, etc.

Furthermore, the job classes may have different service time characteristics. We denote the service time of class $i$ by $B_i$, with mean $E(B_i)$ and mean residual service time $E(R_i)$ with $E(R_i) = E(B_i^2)/2E(B_i)$. Class $i$ jobs arrive according to a Poisson process with rate $\lambda_i$.

We will consider two variants of the priority rule. In the first one a job that has started can not be interrupted, in the second one the service of a job can be interrupted by newly arrived jobs of higher priority classes. If all higher priority jobs are served, the servicing of the job is resumed where it was preempted, i.e., no work is lost. The first type of priority is called *non-preemptive*, the second type is called *preemptive-resume*.

If we think of the situation in which all servicing is done on one processor, non-preemptive priorities are far more natural, since an interrupt might lead to extra setup time or even to destruction of the job. If however one considers a production environment in which the production capacity is mainly labour, then switching from one job to another

might be fairly easy.

## 6.2 The non-preemptive priority system

The analysis is again based on the mean value approach, and a more or less straightforward extension of what we have seen in Chapter 5. The quantities of interest are for class $i$ jobs the mean waiting time $E(W_i)$, the mean number of jobs waiting in the queue $E(L_i^q)$, the mean sojourn time $E(S_i)$ and the mean number of jobs in the system $E(L_i)$.

Let us denote by $\rho_i := \lambda_i E(B_i)$ the utilization by class $i$ jobs. Then according to the PASTA property an arriving job finds with probability $\rho_i$ a class $i$ job in service. Furthermore, upon arrival the job finds on the average $E(L_i^q)$ jobs of class $i$ in the queue.

Let us first look at a job of class 1. This job must wait for jobs of its own class that arrived before, and also for the job (if any) in service. So,

$$E(W_1) = E(L_1^q)E(B_1) + \sum_{i=1}^{r} \rho_i E(R_i) \ .$$

Defining,

$$\rho := \sum_{i=1}^{r} \rho_i$$

and

$$E(R) := \sum_{i=1}^{r} \frac{\rho_i}{\rho} E(R_i),$$

(note that $\rho_i/\rho$ is *not* the probability that an arbitrary customer is of type $i$, but the fraction of time that a service is a type $i$ service), this becomes

$$E(W_1) = E(L_1^q)E(B_1) + \rho E(R), \tag{6.1}$$

where the term $\rho E(R)$ can be interpreted as the expected remaining amount of work currently present at the server. Further we have Little's formula again, stating

$$E(L_1^q) = \lambda_1 E(W_1). \tag{6.2}$$

Combining (6.1) and (6.2) gives us

$$E(W_1) = \frac{\rho E(R)}{1-\rho_1}.$$

Thus

$$E(S_1) = \frac{\rho E(R)}{1-\rho_1} + E(B_1), \quad E(L_1^q) = \frac{\lambda_1 \rho E(R)}{1-\rho_1}, \quad E(L_1) = \frac{\lambda_1 \rho E(R)}{1-\rho_1} + \rho_1.$$

For the job classes $i = 2, ..., r$ the situation is more complicated. Besides the amount of work found upon arrival that a job has to wait for, a job also has to wait for higher priority jobs that arrive later while it is waiting in the queue.

Now let us consider a job of class $i$. According to the reasoning above we get intuitively

$$E(W_i) = \sum_{j=1}^{i} E(L_j^q)E(B_j) + \rho E(R) + E(W_i)\sum_{j=1}^{i-1} \rho_j \ .$$

One may formally show the correctness of the third term, but we will not do this here. Moving the third term to the lefthand side we have

$$E(W_i)(1 - \sum_{j=1}^{i-1} \rho_j) = \sum_{j=1}^{i} E(L_j^q)E(B_j) + \rho E(R) \ . \tag{6.3}$$

Using Little

$$E(L_i^q) = \lambda_i E(W_i),$$

we can rewrite (6.3) as

$$E(W_i)(1 - \sum_{j=1}^{i} \rho_j) = \sum_{j=1}^{i-1} E(L_j^q)E(B_j) + \rho E(R) \ .$$

The right-hand side in this expression is exactly the righthand side of (6.3) with $i$ replaced by $i - 1$. Thus

$$E(W_i)(1 - \sum_{j=1}^{i} \rho_j) = \sum_{j=1}^{i-1} E(L_j^q)E(B_j) + \rho E(R) = E(W_{i-1})(1 - \sum_{j=1}^{i-2} \rho_j) \ .$$

From this and the expression for $E(W_1)$ we easily derive recursively

$$E(W_i) = \frac{\rho E(R)}{(1 - \sum_{j=1}^{i} \rho_j)(1 - \sum_{j=1}^{i-1} \rho_j)}, \ i = 1, 2, .., r. \tag{6.4}$$

From this we directly find $E(S_i)$, $E(L_i^q)$ and $E(L_i)$ using

$$E(S_i) = E(W_i) + E(B_i),$$

$$E(L_i^q) = \lambda_i E(W_i)$$

and

$$E(L_i) = E(L_i^q) + \rho_i.$$

**Example 6.2.1** In an RTDB three types of jobs have to be processed. Jobs of different types arrive according to independent Poisson processes. Type 1 jobs have a mean service time of 1 second, type 2 jobs require 4 seconds on average, and type 3 jobs 10 seconds. Within each class the service times are exponentially distributed. 70 % of all arriving jobs is type 1, 20 % is type 2 and 10 % is type 3. The total arrival rate is 20 jobs per minute. Since the overall mean service time is 2.5 seconds (check), the traffic load is $\rho = 5/6$. If we treat all jobs equally (no priorities), then the mean waiting time is 27.8. With non-preemptive priorities, we get $EW_1 = 6.04$, $EW_2 = 12.09$ and $EW_3 = 55.6$ (we have used that $ER = 5.56$; please verify).

## 6.3 The preemptive-resume priority system

If jobs are preempted, everything becomes more complicated. We no longer have at most one residual service time, but there may be several, although at most one per job class. This makes it difficult to follow the reasoning we used in the non-preemptive case. However, there is an easy way out: don't concentrate on jobs, but on work in the system and note that the performance of a job of class $i$ is completely unaffected by the jobs of the classes $j > i$. The system only works on these jobs if there are no class $i$ or higher priority jobs. So if we try to derive the performance of a class $i$ job we can act as if $\lambda_j = 0$ for $j > i$.

Now note that the mean time in the system, $E(S_i)$, for a class $i$ job consists of three parts:

$$E(S_i) = E(W_i) + E(B_i) + E(I_i). \tag{6.5}$$

The first term, $E(W_i)$, is the mean time until the job goes into service for the first time. The second term, $E(B_i)$, is the job's own mean service time and the third term, $E(I_i)$, is the mean total interruption time during its service time.

So let us look at a system in which the classes $i + 1$ upto $r$ do not exist. And let us compare the waiting time in this system for a lowest priority job, i.e. a class $i$ job, in the non-preemptive and preemptive-resume systems.

First note that the total amount of work in the system is not affected by any of these two priority rules. The rules are so-called *work conserving*, i.e., no capacity is lost. So, although the order in which jobs are treated is different from the non-preemptive case, the total amount of work found upon arrival is the same. Thus

$$E(W_i) = \frac{\sum_{j=1}^{i} \rho_j E(R_j)}{(1 - \sum_{j=1}^{i} \rho_j)(1 - \sum_{j=1}^{i-1} \rho_j)} .$$

Notice that this is the same expression as in (6.4), in case there are only $i$ types of customers. The second term in (6.5) is the mean service time $E(B_i)$ and the third term is the sum of the interruptions. The total interruption time consists of two parts, the sum of the service times of the jobs that interrupt the job we are looking at, and the sum of service times of the jobs with higher priority that arrive during periods in which the job is already

interrupted.
Hence, we get (do we again use that the arrival process is Poisson?)

$$E(I_i) = E(B_i) \sum_{j=1}^{i-1} \lambda_j E(B_j) + E(I_i) \sum_{j=1}^{i-1} \lambda_j E(B_j) \ .$$

So the second and third term in (6.5), the mean service time and the mean total interrupt time, add up to what one might call the *generalized service time*

$$E(B_i) + E(I_i) = \frac{E(B_i)}{(1 - \sum_{j=1}^{i-1} \rho_j)} \ .$$

Hence the total mean sojourn time becomes

$$E(S_i) = \frac{\sum_{j=1}^{i} \rho_j E(R_j)}{(1 - \sum_{j=1}^{i-1} \rho_j)(1 - \sum_{j=1}^{i} \rho_j)} + \frac{E(B_i)}{(1 - \sum_{j=1}^{i-1} \rho_j)} \ .$$

**Example 6.3.1** Returning to the RTDB example, we now allow preemptions. Then the mean total sojourn times of types 1 and 2 decrease, at the expense of those of type 3 jobs: $ES_1 = 1.30$, $ES_2 = 8.61$ and $ES_3 = 75.6$.

# Chapter 7

# Queueing networks

## 7.1 Introduction

Often, congestion phenomena do not just occur in one isolated service facility but at various interconnected stages.

- When a product is made in a plant, it usually passes several production stages with different machines, and in addition it may experience delays in packing and billing.
- A computer job may be delayed at several successive visits to a processor and to memory devices, and finally also at the printer.
- A message traveling from sender to receiver along a path in a communication network may have to wait at the entrance to the network, and also in various switches between the successive channels.

All these service systems may be described as *queueing networks*. In the last fifty years, often spurred by questions from the manufacturing, computing and communications industries, much progress has been made in the performance analysis of queueing networks. A key development has been the identification and analysis of the so-called *product-form networks*, a class of queueing networks for which the joint steady-state distribution of the numbers of customers $X_1, \ldots, X_N$ at the $N$ queues has a *product form*:

$$P(X_1 = n_1, \ldots, X_N = n_N) \;=\; C \prod_{i=1}^{N} p_i(n_i), \tag{7.1}$$

for all *possible* values of $n_1, \ldots, n_N$, with $p_i(n_i)$ a term that *only* refers to the $i$-th station, and with $C$ a normalizing constant.

In this chapter we shall give a global introduction to such product-form networks. More detailed discussions and results can be found in [6, 8, 13, 14, 16, 20, 34]. In the next section we give a model description. Sections 7.3 and 7.4 successively discuss closed and open models of two exponential single server queues. Sections 7.5 and 7.6 consider more general

open and closed networks of exponential single server queues. Extensions to more general networks, also involving other service disciplines than FCFS (first-come first-served), are briefly mentioned in Section 7.7. An efficient method for numerically determining mean values of numbers of customers and sojourn times in product-form queueing networks is presented in Section 7.8.

## 7.2 Model description

We consider in this chapter a network of $N$ stations or queues, $Q_1, \ldots, Q_N$. With the exception of Section 7.7, it is assumed that the service requests of a customer at $Q_i$ are exponentially distributed with mean $1/\mu_i$, $i = 1, \ldots, N$. If a customer has been served in $Q_m$, then it is routed to $Q_n$ with probability $p_{mn}$. This is called *Markov routing*: the routing is described by a Markov chain.

An important distinction is the one between *open* and *closed* networks. In a closed network, there are always $K$ customers. There are no external arrivals, and $\sum_{n=1}^{N} p_{mn} = 1$, viz., no customer ever leaves the network. In an open network, we assume that new customers arrive at the network according to a Poisson process with rate $\Lambda$, choosing $Q_i$ with probability $\lambda_i/\Lambda$. Put differently (cf. Section 2.4), new customers arrive at the queues according to independent Poisson processes, with rate $\lambda_i$ at $Q_i$. We also assume in an open network that a customer who has received service in $Q_m$ leaves the network with probability $p_{m0} := 1 - \sum_{n=1}^{N} p_{mn}$. Throughout the chapter it is assumed that all external interarrival times and all service times are independent. In the next four sections we assume that each queue has only one server, and that this server serves according to the first-come first-served discipline.

**Remark 7.1 (examples of closed queueing networks)**
Closed networks of queues may seem unnatural at first sight, but they do occur quite naturally.

- In a production plant there may be $K$ pallets in use, each one carrying a product in some phase of its development. As soon as a product is completed, it is transferred with its pallet to another department, and the pallet then returns with new parts (a new job). One can model this as a closed network with $K$ customers.

- In the classical multiprogrammed computer system with fixed level $K$ of multiprogramming, at most $K$ programs are admitted in main memory. Customers (programs) alternately visit the central processing unit and a data storage and transfer facility which has access to and can transfer information for only one program at a time. If one is interested in the performance of this system, it is natural to consider the 'heavy-traffic' situation in which always $K$ programs are running; as soon as a program is completed, it is replaced by another one.

- In exercising flow control along a path between sender and receiver in a communication network, a 'window' mechanism is often being used. Suppose that a source

generates messages according to a Poisson process with rate $\Lambda$. The window flow control mechanism requires that at most $K$ messages may be on their way to the destination. The destination acknowledges each received message, and subsequently the source may transmit a new message. To model this situation [22], replace a model of $N$ queues in series (representing the $N$ successive switches in the communication path) by a closed model of $N+1$ queues, by adding a single server queue $Q_0$ with $\exp(\Lambda)$ service times which feeds into $Q_1$ and which receives customers from $Q_N$. If $Q_1, \ldots, Q_N$ together contain $K$ customers (i.e., $K$ unacknowledged messages), then $Q_0$ is empty, so that no new customers can arrive at $Q_1$. If $Q_1, \ldots, Q_N$ together contain less than $K$ customers, then new customers arrive at $Q_1$ according to a Poisson process with rate $\Lambda$.

## 7.3 Two simple closed systems

In this section we use two simple examples of closed queueing systems to develop some intuition for the general closed queueing network.

**The CP - Disk model**
In the CP-Disk system (actually a model for a computer with fixed level of multiprogramming) we distinguish two stations: A central processor CP, and a disk D. $K$ jobs move around in the system, each representing one program in execution. We assume that both CP and D operate as a FCFS single server. The jobs alternately visit the two stations. In station CP their service requires an exponential time with mean $1/\mu_{CP}$; in station D the service time is exponential with mean $1/\mu_D$. Since all service times are exponential, the number of jobs at the CP at time $t$ forms a Markov process.

For this Markov process we can easily make a flow diagram, which subsequently yields the following balance equations for the equilibrium probabilities $p_k$ of having $k$ jobs at the CP:

$$p_k \mu_D = p_{k+1} \mu_{CP}, \qquad k = 0, 1, \ldots, K-1,$$

so

$$p_k = \left(\frac{\mu_D}{\mu_{CP}}\right)^k p_0, \qquad k = 0, 1, \ldots, K.$$

If instead of the one-dimensional state description we use the two-dimensional state description $(k_{CP}, k_D)$ of numbers of jobs in CP and D, we can write this in the following elegant symmetric *product* form.

$$p(k_{CP}, k_D) = C \left(\frac{1}{\mu_{CP}}\right)^{k_{CP}} \left(\frac{1}{\mu_D}\right)^{k_D}, \tag{7.2}$$

with $C$ the normalization constant. Going back to the one-dimensional form, it is easily verified that

$$p_{k_{CP}} = p(k_{CP}, k_D) = \frac{(\frac{\mu_D}{\mu_{CP}})^{k_{CP}}}{\sum_{j=0}^{K}(\frac{\mu_D}{\mu_{CP}})^j} = \frac{1-\frac{\mu_D}{\mu_{CP}}}{1-(\frac{\mu_D}{\mu_{CP}})^{K+1}}(\frac{\mu_D}{\mu_{CP}})^{k_{CP}}, \qquad k_{CP} = 0, 1, \ldots, K. \quad (7.3)$$

**Remark 7.2 (connection to $M/M/1/K$)**
Notice that $p_{k_{CP}} = p(k_{CP}, k_D)$ in (7.3) equals the probability of having $k_{CP}$ customers in an $M/M/1/K$ queue with arrival rate $\mu_D$ and service rate $\mu_{CP}$ (explanation?).

**Remark 7.3 (example)**
Suppose that $\mu_D = 1$, $\mu_{CP} = 2$ and $K = 100$. What is the mean number of jobs at CP? Instead of exactly evaluating this mean (which is given by $\sum kp(k, K-k)$), we shall present an insightful and very accurate approximation (please guess first what the mean approximately is!). The probability that $D$ is empty equals

$$p(100, 0) = \frac{1-\frac{1}{2}}{1-(\frac{1}{2})^{101}}(\frac{1}{2})^{100} \approx (\frac{1}{2})^{101},$$

which is $O(10^{-30})$. Hence, for all practical purposes, one can say that $D$ is never empty. So $CP$ receives customers according to a Poisson process with rate $\mu_D = 1$. Hence $CP$ very closely resembles an $M/M/1$ queue with arrival rate 1 and service rate 2. The mean number of customers in an $M/M/1$ queue with traffic load $\rho = 1/2$ equals $\frac{\rho}{1-\rho} = 1$. The above-described phenomenon is not uncommon: The bottleneck queue, $D$, has almost all jobs (99, on average). Verify that, if $\mu_D = 1$ and $\mu_{CP} = 3/2$, then the division of mean numbers of jobs is approximately 98 versus 2.

**Computer - Terminal model**
In the so-called computer - terminal model there are two stations: a terminal station denoted by T and a computer denoted by C. The terminal station consists of a set of $K$ identical terminals. The computer is modelled as a single server with FCFS service discipline. All terminals are occupied. The system works as follows. A terminal user creates a job for the computer and sends it away. Then he waits for the results. As soon as he gets his reply he starts to create a new job. So for every terminal there is an alternation of think and wait periods.

Let us assume that the job sizes at the computer are exponentially distributed with mean $1/\mu_C$ and that the think times are exponential with mean $1/\mu_T$.

Although each terminal is used by a different user, it is possible to look at the terminal station as one multi-server station. If we are not interested in which terminals are thinking but only in how many, then a one-dimensional state description suffices. Using as state descriptor the number of 'thinking' terminals, $k_T$, we can make a flow diagram. From the flow diagram we easily find

$$p_k = \frac{1}{k!}\left(\frac{\mu_C}{\mu_T}\right)^k p_0, \qquad k = 0, 1, \ldots, K.$$

As in the previous example the one-dimensional state description hides the structure. If we take as state $(k_T, k_C)$, then the equilibrium distribution can be given in the form

$$p(k_T, k_C) = D\frac{1}{k_T!}\left(\frac{1}{\mu_T}\right)^{k_T}\left(\frac{1}{\mu_C}\right)^{k_C}, \tag{7.4}$$

with $D$ again a normalizing constant.
Since $k_T + k_C = K$, it is easily verified that

$$p_{k_T} = p(k_T, k_C) = \frac{(\frac{\mu_C}{\mu_T})^{k_T}/k_T!}{\sum_{j=0}^{K}(\frac{\mu_C}{\mu_T})^j/j!}, \qquad k_T = 0, 1, \ldots, K. \tag{7.5}$$

**Remark 7.4 (the machine - repairman model)**
Mathematically, this model is the same as the Machine - Repairman model. That is a model with $K$ machines, which break down after an $\exp(\mu_T)$ distributed time, and then go to a repairman who repairs machines in FCFS order, with $\exp(\mu_C)$ repair times. So the terminals are replaced by the machines and the computer by the repairman. The think time corresponds to the time until the next failure and the computer time corresponds to the repair time.

**Remark 7.5**
Consider the $M/M/K/K$ system, also known as the Erlang loss system (A.K. Erlang was a Danish telephone engineer, who made important contributions to queueing theory and its communications applications): Customers arrive according to a Poisson($\mu_C$) process at $K$ servers (e.g., calls requesting a telephone line, or cars requesting a parking place). When all servers are busy, the arriving customer disappears never to return. Service times of admitted customers are $\exp(\mu_T)$ distributed. The number of customers in this system forms a Markov process. The corresponding balance equations are *identical* to those for the number of thinking terminals (or operative machines). Hence, Formula (7.5) also gives the steady-state distribution of the number of busy servers in the $M/M/K/K$ system.

## 7.4 Two $M/M/1$ queues in series

Following R.R.P. Jackson [12], we now consider the *open* tandem queue: Two single server queues $Q_1$ and $Q_2$ in series. $Q_1$ is an $M/M/1$ queue with arrival rate $\lambda$ and service rate $\mu_1$. All customers who leave $Q_1$ are routed to $Q_2$, where they require an $\exp(\mu_2)$ distributed service time. Upon service completion in $Q_2$, a customer leaves the system.

The two-dimensional process $\{(X_1(t), X_2(t)),\ t \geq 0\}$ of numbers of customers in $Q_1$ and $Q_2$ at time $t$ is an irreducible Markov process. If $\lambda < \min(\mu_1, \mu_2)$ (which is henceforth assumed to hold: It guarantees that less work arrives at each queue per time unit than its server can handle), then this Markov process is positive recurrent, and a steady-state

distribution $p(m,n) := \lim_{t\to\infty} P(X_1(t) = m, X_2(t) = n)$ exists. The balance equations for this steady-state distribution are given by:

$$\begin{aligned}
(\lambda+\mu_1+\mu_2)p(m,n) &= \lambda p(m-1,n) + \mu_1 p(m+1,n-1) + \mu_2 p(m,n+1), \\
&\qquad m,n = 1,2,\ldots, \qquad (7.6)\\
(\lambda+\mu_1)p(m,0) &= \lambda p(m-1,0) + \mu_2 p(m,1), \qquad m = 1,2,\ldots, \\
(\lambda+\mu_2)p(0,n) &= \mu_1 p(1,n-1) + \mu_2 p(0,n+1), \qquad n = 1,2,\ldots, \\
\lambda p(0,0) &= \mu_2 p(0,1),
\end{aligned}$$

with normalizing condition

$$\sum_{m=0}^{\infty}\sum_{n=0}^{\infty} p(m,n) = 1.$$

Jackson [12] has verified that this set of equations has the following solution (and the theory of Markov processes implies that it is the unique solution):

$$p(m,n) \;=\; (1-\frac{\lambda}{\mu_1})(\frac{\lambda}{\mu_1})^m(1-\frac{\lambda}{\mu_2})(\frac{\lambda}{\mu_2})^n, \qquad m,n = 0,1,\ldots. \qquad (7.7)$$

Summing over $n$ yields the marginal distribution $p_1(m) := \lim_{t\to\infty} P(X_1(t) = m) = (1-\frac{\lambda}{\mu_1})(\frac{\lambda}{\mu_1})^m$, the result we should have expected: $Q_1$ is an $M/M/1$ queue. It is more remarkable that, similarly,

$$p_2(n) := \lim_{t\to\infty} P(X_2(t) = n) = (1-\frac{\lambda}{\mu_2})(\frac{\lambda}{\mu_2})^n, \qquad (7.8)$$

so that $Q_2$ *also* behaves like an $M/M/1$ queue.

**Remark 7.6**
Notice the *product form* of the steady-state distribution:

$$p(m,n) = p_1(m)p_2(n),$$

and notice that, apparently, the steady-state numbers of customers at $Q_1$ and $Q_2$ are *independent*.

Inspired by Jackson's result, P.J. Burke [2] has proven his *output theorem*:
(i) the successive interdeparture times of an $M/M/c$ queue in steady state form a Poisson process, with the same rate as the arrival process.
(ii) The departure process of an $M/M/c$ queue *before* time $t$ is independent of the number of customers *at* time $t$.

Statement (i) immediately implies the result (7.8) for $Q_2$. Statement (ii) implies the independence of the simultaneously present numbers of customers at $Q_1$ and $Q_2$.

**Remark 7.7 (reversibility)**
Reich [21] has proved the above 'output theorem' of Burke using the concept of *reversibility*. A stochastic process is called reversible if it has the following property: When the direction of time is reversed, then the behavior of the process remains the same. Speaking intuitively, if we take a film of such a process and run the film backwards, the resulting process will be statistically indistinguishable from the original process. This property is formally described in the following definition:
*A stochastic process $X(t)$ is reversible if $(X(t_1), X(t_2), \ldots, X(t_n))$ has the same distribution as $(X(\tau - t_1), X(\tau - t_2), \ldots, X(\tau - t_n))$ for all $t_1, t_2, \ldots, t_n$ and $\tau$.*

It is well known that the queue length process in an $M/M/c$ queue *is* reversible. This reversibility immediately leads to the two statements of the output theorem:
(i) the output process of $Q_1$ is Poisson($\lambda$), since in the time-reversed situation, the output process becomes the original input process (which is Poisson($\lambda$)).
(ii) the number of customers in $Q_1$ at time $t$ is independent of the departure process from $Q_1$ *before* time $t$. Indeed, in the time-reversed case this is the obvious statement that the number of customers in $Q_1$ at time $t$ is independent of the arrival process at $Q_1$ *after* time $t$ (we used the memoryless property of the arrivals here!).

## 7.5 Open Jackson networks

Inspired by the results of R.R.P. Jackson [12] and Burke [2], J.R. Jackson [10] has considered the open queueing network of $N$ $./M/1$ queues that has been described in Section 7.2: Customers arrive according to independent Poisson processes, and they travel from queue to queue according to Markovian routing with matrix $(p_{mn})$, requiring exponentially distributed service times at single FCFS servers. Because of all the memoryless assumptions, the process $\{(X_1(t), X_2(t), \ldots, X_N(t)),\ t \geq 0\}$ of numbers of customers at $Q_1, \ldots, Q_N$ is a Markov process. Jackson [10] has verified that the balance equations for its steady-state distribution $p(n_1, \ldots, n_N)$ are satisfied by

$$p(n_1, \ldots, n_N) \ = \ \prod_{i=1}^{N} (1 - \frac{\Lambda_i}{\mu_i})(\frac{\Lambda_i}{\mu_i})^{n_i}, \qquad n_1, \ldots, n_N = 0, 1, \ldots, \tag{7.9}$$

with

$$\Lambda_i = \lambda_i + \sum_{j=1}^{N} \Lambda_j p_{ji}, \tag{7.10}$$

(in vector-matrix notation: $\Lambda = \lambda + \Lambda P$). Here $\Lambda_i$ is the *throughput* of $Q_i$: The sum of external arrival rate $\lambda_i$ and all the internal flows going from $Q_1, \ldots, Q_N$ to $Q_i$. It can be shown that the steady-state distribution exists if $\Lambda_i < \mu_i$ for $i = 1, \ldots, N$. (Exercise: write down the balance equations, and verify that they are satisfied by (7.9)).

The *product form* of (7.9) implies that in steady state the numbers of customers at the various queues at each particular point in time are *independent*. We further conclude that each queue $Q_i$ in this so-called *Jackson network* behaves like an $M/M/1$ queue with arrival rate $\Lambda_i$ and service rate $\mu_i$. In particular, with $\rho_i := \Lambda_i/\mu_i$, the mean number of customers at $Q_i$ is $EL_i = \rho_i/(1-\rho_i)$, and Little's formula (which again holds; think of the money argument!) implies that the mean sojourn time at $Q_i$ is $ES_i = 1/(\mu_i(1-\rho_i))$.

**Remark 7.8**
Jackson[10] believed that the $M/M/1$ behavior of $Q_i$ is not surprising in view of Burke's output theorem. As observed in Section 2.4, if two independent Poisson processes with rates $\nu_1$ and $\nu_2$ are *merged*, the resulting sum process is a Poisson process with rate $\nu_1+\nu_2$; and if in a Poisson arrival process with rate $\nu$ each arrival is sent left with probability $p$ and right with probability $1-p$, then this *splitting* results in two independent Poisson processes with rates $\nu p$ and $\nu(1-p)$. So one would indeed readily believe that all flows in the above Jackson network are Poisson, leading to the result that each queue behaves like an $M/M/1$ queue. However, this argument is incorrect when *feedback* can occur in the network, viz., a customer may return to a queue where it has been served before. The resulting dependence destroys the Poisson property, as is demonstrated in the following example. Consider a single server queue with external Poisson($\lambda$) arrival process, independent exp($\mu$) service times, and feedback: After a service completion, a customer leaves the system with probability $1-p$ and returns to the end of the queue with probability $p$. The number of customers in this system is a Markov process, and its steady-state distribution exists if $\lambda < \mu(1-p)$. In the latter case, it is easily verified by writing down the balance equations that the steady-state distribution is given by

$$P(X=k) \;=\; (1-\frac{\lambda}{\mu(1-p)})(\frac{\lambda}{\mu(1-p)})^k, \qquad k=0,1,\ldots,$$

(as if the arrival process is Poisson($\lambda/(1-p)$)). But is the *total* arrival process at this queue really Poisson? Take $\lambda = 10^{-3}$, $\mu = 10^6$, $p = 1-10^{-6}$; then typically there will be a burst of arrivals, continuing for about $1/(\mu(1-p)) = 1$ time unit, and then on average one has to wait $1/\lambda = 1000$ time units for another burst of arrivals. This is clearly *not* a Poisson process!
Thanks to the work of Kelly [13] and others, much insight has been obtained into the phenomenon that each queue in the above-described Jackson network behaves *as if* it is an $M/M/1$ queue, although the arrival processes at a queue are not necessarily Poisson. The concept of *quasi-reversibility* [6, 13] plays a key role here; but its discussion lies outside the scope of these lecture notes.

## 7.6 Closed Jackson networks

In 1963, J.R. Jackson [11] extended the results of his paper [10] to *closed* networks of $./M/1$ (actually, $./M/c$) queues. This model is obtained from that of the previous section

by taking $\lambda_i = 0$ and $p_{i0} = 0$, $i = 1, \ldots, N$, and starting with, say, $K$ customers. Jackson [11] proved that the steady-state distribution of the numbers of customers $X_1(t), \ldots, X_N(t)$ at time $t$ is given by

$$\begin{aligned} p(n_1, \ldots, n_N) &= \frac{1}{G(N,K)} \prod_{i=1}^{N} (\frac{\Lambda_i}{\mu_i})^{n_i}, \\ & \qquad n_1, \ldots, n_N = 0, 1, \ldots, \quad n_1 + \cdots + n_N = K, \end{aligned} \tag{7.11}$$

with $(\Lambda_1, \ldots, \Lambda_N)$ satisfying

$$\Lambda_i = \sum_{j=1}^{N} \Lambda_j p_{ji}, \tag{7.12}$$

and the normalizing constant $G(N, K)$ being given by

$$G(N,K) = \sum \prod_{i=1}^{N} (\frac{\Lambda_i}{\mu_i})^{n_i},$$

the sum being taken over the $\binom{N+K-1}{K}$ possible combinations with $n_1, \ldots, n_N \geq 0$, $n_1 + \cdots + n_N = K$. (Question: In how many ways can you put $K$ balls into $N$ urns? Equivalently, in how many ways can you form a row of $K$ zeros and $N-1$ ones?)

**Remark 7.9**
Equation (7.12) has the same form as the equation $\pi = \pi P$ for the steady-state distribution of a Markov chain with transition matrix $P$. However, the normalizing condition $\sum \Lambda_i = 1$ does not hold: The $\Lambda_i$ are determined up to a multiplicative constant. Indeed, if $(\Lambda_1, \ldots, \Lambda_N)$ is a solution, then so is $(c\Lambda_1, \ldots, c\Lambda_N)$. Of course, the numerator in (7.11) then is multiplied by $c^{n_1 + \cdots + n_N} = c^K$, but so is the normalizing constant $G(N, K)$ in the denominator. The *actual* throughput of node $Q_i$ depends on the total number of customers $K$ in the system. In Section 7.8 we show how one can easily determine this throughput $\Lambda_i(K)$.

**Remark 7.10**
In the special case that $p_{12} = p_{23} = \cdots = p_{N1} = 1$, (7.12) yields $\Lambda_1 = \cdots = \Lambda_N$, which makes sense for this closed cyclic system. We can now choose all $\Lambda_i$ equal to one, say, or all $\Lambda_i$ equal to $\mu_1$, for the purpose of presenting the steady-state distribution in an elegant form. Verify that we thus obtain the result of (7.3) for the model of a CPU and disk.

**Remark 7.11**
Although (7.11) again has a product form, here – contrary to the situation in *open* Jackson networks – the numbers of customers at the various queues are *not* independent. For $N = 2$, they are even linearly correlated, since $X_1(t) + X_2(t) = N$ for all $t$.

**Remark 7.12 (the curse of dimensionality)**
For networks of a realistic size, the number of states of the $N$-dimensional Markov process may be excessively large. E.g., if there are $N = 10$ stations and $K = 30$ customers, then the number of states is $\binom{39}{9} \approx 2\ 10^8$. Hence, it may be prohibitively difficult to evaluate $G(N, K)$ by just summing all $\binom{N+K-1}{K}$ terms. Fortunately, there is an efficient algorithm to compute this normalizing constant: Buzen's convolution algorithm. Details may be found in [16] or [17].

## 7.7 Generalizations

The product-form results of Sections 7.5 and 7.6 have been extended in many directions. Key publications are [1] and [13]. Without proof we state the following. Consider the queueing network of Section 7.2, with the following extensions.

1. Instead of FCFS with exponential service times at all nodes, one may have *processor sharing* at some nodes (i.e., if there are $j$ customers in a node, then they all receive service simultaneously, each with service speed $1/j$), and Last-Come First-Served Preemptive Resume at some other nodes. In nodes with those two disciplines, the service time distribution may be *general*, with mean $1/\mu_i$ at $Q_i$; the product-form expressions (7.9) and (7.11) then remain unchanged. This demonstrates a remarkable *insensitivity* for the actual service time distribution in nodes with processor sharing and with LCFS Preemptive Resume.

2. A FCFS queue $Q_i$ may have a service intensity $\mu_i(n_i)$ if there are $n_i$ customers in $Q_i$, $i = 1, \ldots, N$. In particular, we may allow $c_i \geq 1$ servers in $Q_i$ (take $\mu_i(n_i) = \min(n_i, c_i)\mu_i$), where $c_i$ may even be infinite. In the latter case, it has been shown that the above-mentioned insensitivity for the service time distribution again holds.

**Remark 7.13**
In the case of a closed network of $N = 2$ queues, with $K$ customers, and with $Q_1$ an infinite server queue with mean service time $1/\mu_1$ and $Q_2$ a single server queue with $\exp(\mu_2)$ distributed service times and FCFS service discipline, one has

$$p(n_1, n_2) \;=\; \frac{1}{G(2,K)} \frac{(\frac{\Lambda_1}{\mu_1})^{n_1}}{n_1!} (\frac{\Lambda_2}{\mu_2})^{n_2}, \qquad n_1, n_2 \geq 0, \quad n_1 + n_2 = K. \tag{7.13}$$

The balance of flows (throughputs) (7.12) here results in $\Lambda_1 = \Lambda_2$, which may be chosen equal to, say, $\mu_2$. This results in

$$p(n_1, K - n_1) \;=\; \frac{(\frac{\mu_2}{\mu_1})^{n_1}/n_1!}{\sum_{j=0}^{K} (\frac{\mu_2}{\mu_1})^j/j!}, \qquad n_1 = 0, 1, \ldots, K. \tag{7.14}$$

This is the same form as (7.5) which has been derived for the computer - terminal and machine - repair models (and for the Erlang loss model) of Section 7.3. Indeed, working

machines or thinking terminals experience no waiting time, and this part of the network may be viewed as a service facility with an *infinite* number of servers.

A third extension of the product-form networks is obtained by allowing *several customer classes* (but without priorities). Each class may have its own arrival rates, service rates and routing probabilities. Some classes may be open, others closed. In a station with FCFS service, all service times of all classes must be exponentially distributed with the same service rate. The joint distribution of numbers of customers of each class at the various queues again has a product form – in fact, this becomes a double product form, as one product is taken w.r.t. the queues and another product w.r.t. the customer classes. We refrain from giving details, for which we refer to [6, 13, 17].

## 7.8 Mean value analysis

In an open product-form network, an extension of PASTA has been shown to hold: An arriving customer (external arrival, or customer coming from another queue) sees time averages, i.e., sees the network in equilibrium. Hence it is easy to obtain the mean number of customers seen at $Q_i$ by an arriving customer: Just take the steady-state mean number of customers. Furthermore, the mean sojourn time at $Q_i$ follows by an application of Little's formula.

It is more difficult to determine the mean number of customers and mean sojourn time in a queue of a *closed* product-form network. Let us first consider the steady-state number of customers $EL_i$ in $Q_i$. Formally, $EL_i$ can be obtained from (7.11), but then not only $G(N, K)$ must be determined, but evaluating $\sum k_i p(k_1, \ldots, k_i, \ldots, k_N)$ requires summing over a possibly very large number of terms. Fortunately, there is a very efficient way of determining $EL_1, \ldots, EL_N$ in the closed product-form network of Section 7.6, *without evaluating the actual distributions*: The MVA algorithm (Mean Value Analysis algorithm) of Reiser and Lavenberg [23]. The MVA algorithm is based upon the following *arrival theorem* of Lavenberg and Reiser [18], see also Sevcik and Mitrani [29]:

**Arrival theorem in closed product-form networks**
*When a customer leaves a station in the closed product-form network of Section 7.6, then the joint distribution of the numbers of customers in all stations at this jump epoch equals the steady-state distribution of the numbers of customers in the same network but with* $K - 1$ *customers.*

So the jumping customer sees the remainder of the network in equilibrium, but not counting himself in; this should be compared with the PASTA result for open product-form networks. The arrival theorem is remarkable and even counterintuitive. Although the jumping customer has been in the system forever, and has thus influenced the past system behavior, at the moment of its jump the remainder of the system behaves as if the jumping customer has never been there.

**Remark 7.14**
In the computer - terminal model of Section 7.3, the correctness of the arrival theorem can readily be verified, using the equivalence of that model with the $M/M/K/K$ Erlang loss model. In the $M/M/K/K$ model, PASTA implies that

$$P(\text{arriving customer meets n customers}) \;=\; \frac{(\frac{\mu_2}{\mu_1})^n/n!}{\sum_{j=0}^{K}(\frac{\mu_2}{\mu_1})^j/j!}, \qquad n = 0, 1, \ldots, K. \tag{7.15}$$

Hence

$$P(\text{arriving customer is admitted}) \;=\; \frac{\sum_{j=0}^{K-1}(\frac{\mu_2}{\mu_1})^j/j!}{\sum_{j=0}^{K}(\frac{\mu_2}{\mu_1})^j/j!}, \tag{7.16}$$

and therefor

$$P(\text{arriving customer meets n customers} \mid \text{he is admitted}) \;=\; \frac{(\frac{\mu_2}{\mu_1})^n/n!}{\sum_{j=0}^{K-1}(\frac{\mu_2}{\mu_1})^j/j!}, \quad n = 0, 1, \ldots, K-1. \tag{7.17}$$

The latter probability also is the probability that a jumping customer in the computer - terminal model sees $n$ thinking terminals (here we use the equivalence of the computer - terminal model and the Erlang loss model).

The arrival theorem forms the basis for the MVA algorithm. Denote the mean number of customers, throughput and mean sojourn time at $Q_i$ in a closed network with $K$ customers by $EL_i(k)$, $\Lambda_i(K)$ and $ES_i(K)$, respectively. The MVA algorithm consists of an iterative evaluation of the following three steps (once more we remind the reader that we restrict ourself to the model of Section 7.6):

1. Little's formula for $Q_i$:

$$EL_i(K) = \Lambda_i(K) ES_i(K). \tag{7.18}$$

2. Arrival theorem:

$$ES_i(K) = \frac{1}{\mu_i}[EL_i(K-1) + 1]. \tag{7.19}$$

3. Little's formula for all $N$ queues:

$$\begin{aligned} K &= \sum_{j=1}^{N} EL_j(K) = \sum_{j=1}^{N} \Lambda_j(K) ES_j(K) \\ &= \Lambda_i(K) \sum_{j=1}^{N} \frac{\lambda_j}{\lambda_i} ES_j(K), \end{aligned} \tag{7.20}$$

(here $\lambda_j/\lambda_i = \Lambda_j/\Lambda_i$ denotes the fixed ratio of throughputs at $Q_j$ and $Q_i$, which we determine once and for all from (7.12)), so that

$$\Lambda_i(K) \;=\; \frac{K}{\sum_{j=1}^{N} \frac{\lambda_j}{\lambda_i} ES_j(K)}. \tag{7.21}$$

Starting with $ES_i(1) = 1/\mu_i$, we successively find $\Lambda_i(1)$ from (7.21), $EL_i(1)$ from (7.18), then $ES_i(2)$ from (7.19), etc. Question: What is the computational complexity of the MVA algorithm?

It should be noted that (1) the MVA algorithm determines the actual values of the throughputs in a closed product-form network with $K$ customers, $K = 1, 2, \ldots,$ while (7.12) only gave their proportions; and (2) the MVA algorithm neither gives the normalizing constant $G(N, K)$ nor the *distribution* of the numbers of customers.

# Bibliography


[1] F. Baskett, K.M. Chandy, R.R. Muntz and F.G. Palacios (1975). *Open, closed and mixed networks of queues with different classes of customers*. J. Assoc. Comput. Mach. **22**, 248-260.

[2] P.J. Burke (1956). *The output of a queueing system*. Oper. Res. **4**, 699-704.

[3] J.A. Buzacott, J.G. Shanthikumar (1993). *Stochastic Models of Manufacturing Systems*, Prentice Hall, Englewood Cliffs.

[4] J.W. Cohen (1976). *On Regenerative Processes in Queueing Theory*, Springer, Berlin.

[5] J.W. Cohen (2012). *The Single Server Queue*, North-Holland, Amsterdam.

[6] N.M. van Dijk (1993). *Queueing Networks and Product Forms*, Wiley, New York.

[7] D. Gross, C.M. Harris (1985). *Fundamentals of Queueing Theory*, Wiley, Chichester.

[8] M. Harchol-Balter (2013). *Performance Modeling and Design of Computer Systems: Queueing Theory in Action*, Cambridge University Press, New York.

[9] M. Haviv (2013). *A Course in Queueing Theory*, Springer, New York.

[10] J.R. Jackson (1957). *Networks of waiting lines*. Oper. Res. **5**, 518-521.

[11] J.R. Jackson (1963). *Jobshop-like queueing systems*. Management Science **10**, 131-142.

[12] R.R.P. Jackson (1954). *Queueing systems with phase-type service*. Operational Research Quarterly **5**, 109-120.

[13] F.P. Kelly (2011). *Reversibility and Stochastic Networks*, Cambridge University Press, New York.

[14] F.P. Kelly and E. Yudovina (2014). *Lecture Notes on Stochastic Networks*, Cambridge University Press, New York.

[15] L. Kleinrock (1975). *Queueing Systems, Vol. I: Theory*, Wiley, New York.

[16] L. Kleinrock (1976). *Queueing Systems, Vol. II: Computer Applications*, Wiley, New York.

[17] S.S. Lavenberg (ed.) (1983). *Computer Performance Modeling Handbook*, Academic Press, New York.

[18] S.S. Lavenberg and M. Reiser (1980). *Stationary state probabilities at arrival instants for closed queueing networks with multiple types of customers*. J. Appl. Probab. **17**, 1048-1061.

[19] J.D. Little (1961). *A proof of the queueing formula* $L = \lambda W$. Oper. Res. **9**, 383-387.

[20] P. Van Mieghem (2009). *Performance Analysis of Communication Systems and Networks*, Cambridge University Press, New York.

[21] E. Reich (1957). *Waiting times when queues are in tandem*. Ann. Math. Statist. **28**, 768-773.

[22] M. Reiser (1979). *A queueing network analysis of computer communication networks with window flow control*. IEEE Trans. Commun. **27**, 1199-1209.

[23] M. Reiser and S.S. Lavenberg (1980). *Mean value analysis of closed multichain queueing networks*. J. Assoc. Comput. Mach. **27**, 313-322.

[24] S.I. Resnick (2002). *Adventures in Stochastic Processes*, Birkhäuser Verlag, Boston.

[25] S.M. Ross (1996). *Stochastic Processes*, 2nd ed., Wiley, New York.

[26] S.M. Ross (2014). *Introduction to Probability Models*, 11th ed., Academic Press, London.

[27] R. Schassberger (1970). *On the waiting time in the queueing system* $GI/G/1$. Ann. Math. Statist. **41**, 182-187.

[28] R. Schassberger (1973). *Warteschlangen*, Springer-Verlag, Berlin.

[29] K.C. Sevcik and I. Mitrani (1981). *The distribution of queueing network states at input and output instants*. J. Assoc. Comput. Mach. **28**, 358-371.

[30] S. Stidham (1974). *A last word on* $L = \lambda W$. Oper. Res. **22**, 417-421.

[31] L. Takács (1962). *Introduction to the Theory of Queues*, Oxford University Press, New York.

[32] H.C. Tijms (1990). *Stochastic Modelling and Analysis: a Computational Approach*, John Wiley & Sons, Chichester.

[33] H.C. Tijms (1994). *Stochastic Models: an Algorithmic Approach*, John Wiley & Sons, Chichester.

[34] J. Walrand (1988). *An Introduction to Queueing Networks*, Prentice Hall, Englewood Cliffs (NJ).

[35] W. Whitt (1986). *Approximating a point process by a renewal process I: two basic methods*. Oper. Res. **30**, 125-147.

[36] E.T. Whittaker, G.N. Watson (1946). *Modern Analysis*, Cambridge University Press, London.

[37] R.W. Wolff (1982). *Poisson arrivals see time averages*. Oper. Res. **30**, 223-231.

[38] R.W. Wolff (1989). *Stochastic Modeling and the Theory of Queues*, Prentice-Hall, London.