UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering,
Mathematics & Computer Science

Overflow loss systems

Product forms, blocking probabilities, insensitivity and an application to health care modelling

Barteld Schilstra

MSc Thesis Applied Mathematics
June 2021

Supervisor:
Prof. dr. N.M. van Dijk

Graduation committee:
Prof. dr. R.J. Boucherie
Prof. dr. N.M. van Dijk
Prof. dr. A.A. Stoorvogel

Stochastic Operations Research
Department of Applied Mathematics
Faculty of EEMCS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Abstract

Background: Queueing systems with overflow can be found in a variety of settings, such as telecommunications, call centers and health care. In overflow systems, jobs are routed to another, secondary station if the primary station is fully occupied. This report mainly focuses on a two-station overflow loss system with two job types. In this system, type 1 jobs arrive at (primary) station 1 and are overflowed to (secondary) station 2 if all servers at station 1 are occupied. Besides that, direct arrivals of type 2 jobs at (secondary) station 2 are incorporated. The overflow system has the following distinguishing characteristics. First of all, it has a serial structure, which means that jobs that complete service at the primary station are allowed to be transferred to the secondary station. Secondly, it allows the service parameters to be dependent on the job type, the station at which the job is served and whether or not the job is overflowed. Thirdly, it includes both overflow and jump-over blocking. Finally, it assumes that the allowed number of jobs at the secondary station is restricted to a so-called coordinate convex set.

The two-station overflow system with serial structure is studied under two different assumptions. The first assumption that is considered is that overflowed jobs always complete service at the secondary station, even if a server at the primary station becomes available. Secondly, the system is also studied under the assumption that overflowed jobs switch from the secondary station to the primary station as soon as a server at the primary station becomes available. The latter assumption has been known in teletraffic theory for a long time under the terms of (immediate) repacking or call packing. In the sequel, the overflow system under this assumption is therefore referred to as the system with call packing, while the overflow system under the former assumption is referred to as the system without call packing.

Objectives: In this report, the following objectives are aimed for:

- Determine the joint steady-state distribution of the number of jobs in the overflow system. This is done both for the system with call packing and the system without call packing.
- Determine which blocking probabilities can be of interest, study how these can be computed, and examine how these are affected by the assumption of call packing.
- Examine whether the overflow system is insensitive to the service time distributions. If this is the case, the steady-state distribution does not depend on the service time distributions other than through their means.
- Illustrate a possible application of the overflow system by describing how (an adapted version of) the overflow system could be useful to model the interaction between an intensive care unit (ICU) and a step-down unit (SDU).

Results: First of all, the joint steady-state distribution of the number of jobs in the overflow system is determined. For the system with call packing, a product-form solution for the steady-state distribution is obtained. For the system without call packing, the Gauss-Seidel method and Grassmann-Taksar-Heyman (GTH) algorithm are applied to find the steady-state distribution.

From these steady-state distributions, in turn, the blocking probabilities of interest can be computed. This can be done by using the Poisson Arrivals See Time Averages (PASTA) property of Poisson arrivals or by computation of a Palm probability. Numerical results of the blocking probabilities are given, which illustrate how the blocking probabilities for the system with call packing compare to those for the system without call packing.

Subsequently, discrete-event simulation is used to study whether the overflow system can be expected to be insensitive to the service time distributions. It appears that both the system with call packing and the system without call packing are sensitive. However, the simulation results indicate that the system with call packing might be insensitive if the service rates of non-overflowed type 1 jobs at (primary) station 1 and overflowed type 1 jobs at (secondary) station 2 are assumed to be equal, and service is preemptively resumed after an overflowed job switches to (primary) station 1. It is then shown that, under these conditions, the product-form solution for the steady-state distribution remains valid when each of the service time distributions is a mixture of Erlang distributions, by which a non-negative distribution can be arbitrarily closely approximated.

As a final point of interest, a possible application to ICU-SDU modelling is studied. Literature regarding ICUs and SDUs is described in order to study which assumptions are reasonable to make. It is then found that the overflow system, in adapted form, could be useful to model the interaction between an ICU and SDU if it can be assumed that ICU patients may be overflowed to the SDU.

Conclusion: This report is concerned with a two-station overflow system with a serial structure. For this overflow system, several analytical and numerical results regarding the steady-state distribution, blocking probabilities and insensitivity are obtained. These results can be of interest from a theoretical point of view. Furthermore, as illustrated by the application to ICU-SDU modelling, a practical usefulness is also conceivable.

Acknowledgements

I would like to express my gratitude to the members of the graduation committee: prof. dr. Nico van Dijk, prof. dr. Richard Boucherie and prof. dr. Anton Stoorvogel. First of all, I am grateful to Nico van Dijk for his supervision and useful advice and feedback, but also for awakening my interest in overflow systems and providing the opportunity to work on the paper [60], which ultimately led to the writing of this report. Secondly, I would like to thank Richard Boucherie for taking on the role of chair of the graduation committee. Besides that, the monthly check-ins via Teams were appreciated. Thirdly, I am thankful to Anton Stoorvogel for joining the graduation committee and reading and evaluating my work.

Besides that, I owe a debt of gratitude to my family and friends, in particular Henk, Cindy, Tjomme, Annelore and Rikke, for their support.

Contents

List of Algorithms, Definitions, Examples, Figures, Remarks, Tables and Theorems

List of Algorithms

List of Definitions

List of Examples

List of Figures

List of Remarks

List of Tables

List of Theorems (and a Corollary)

Chapter 1

Introduction

The structure of this chapter is as follows. First of all, in Section 1.1, some background information is provided. Next, Section 1.2 discusses the motivation and objectives of the research project. Finally, Section 1.3 contains the outline of the report.

1.1 Background

1.1.1 Queueing theory

In daily life, we come across many service systems, such as supermarkets, call centers, hospitals and airlines. In order to provide a high level of service, it is often of interest to gain insight into the (expected) performance of these systems. This can be done, for example, by obtaining a prediction of levels of congestion or an estimate of how much capacity is required to reach a desired level of service. However, because of the unpredictability of arrivals and service times, such service systems are often difficult to analyze. For this purpose, queueing theory can be of great help.

Queueing theory finds its origin at the beginning of the 20th century when A.K. Erlang applied methods of the theory of probability in the field of telephony (see [15]). Since then, it has found an application in a variety of settings, including, among others, the areas of health care (e.g. hospitals) and emergency services (e.g. ambulances, fire brigade). In its essence, queueing theory is concerned with the mathematical study of queueing systems. A queueing system consists of one or multiple stations (or server groups or queues, etc.), which each have a number of servers (or machines, etc.).

First of all, if there is just one station, the queueing system is often referred to as a single-station queueing system. Such systems have in common that jobs (or customers or calls, etc.) arrive at the station to receive some sort of service or treatment. However, there are numerous variations that can be thought of. For example, if an arriving job finds all servers occupied, it could be blocked (or rejected), wait until a server becomes available or replace a job that is in service.

Secondly, a queueing system that consists of multiple stations is commonly called a queueing network (or network of queues). In such systems, a job may pass through a number of stations instead of only visiting a single station. The network could have a (completely) serial structure as in an assembly line, but also a more general structure. In this report, the main focus is on queueing networks with overflow. This means that jobs are served at another, secondary station if the primary station is fully occupied. In Sections 1.1.2 and 2.7, overflow systems are further discussed.

The following example illustrates an application of queueing theory in practice.

Example 1 (Intensive care unit)**.** Intensive care units (ICUs) provide intensive care and treatment for patients with a critical condition. It is thus important that sufficient ICU beds and personnel are available in order to care for and treat these ICU patients. In order to obtain insight into how much capacity is necessary, queueing theory can be useful. As a consequence, ICUs have frequently been studied from a queueing perspective (see e.g. [2, 16, 19, 20, 23, 25, 39, 41, 42, 59]).

First of all, the ICU can be seen as a single-station queueing system (see e.g. [19, 20, 23, 25, 42]). In this case, patients (i.e. customers) arrive to the ICU (i.e. the station) to receive intensive care and treatment (i.e. service) in one of the operational ICU beds (i.e. at one of the servers). Secondly, the ICU can also be modelled as part of a queueing network. For example, the queueing network may consist of multiple wards or departments at a hospital, among which the ICU (see e.g. [2, 41, 59]). Besides that, a queueing network with multiple ICUs at different hospitals in a region can also be considered (see e.g. [16, 39]).

The queueing systems that have been studied also rely on different assumptions regarding what occurs if an arriving patient finds the ICU fully occupied. These include the following:

- The arriving patient is rejected and leaves the system (see e.g. [19, 42]).
- The arriving patient waits for an available bed (see e.g. [23, 25, 41]).
- The ICU patient in relatively the best condition is bumped from the ICU (see e.g. [20]).
- Depending on the patient type, the arriving patient is rejected or waits for an available bed (see e.g. [59]).
- Depending on the situation, the arriving patient balks (i.e. is rejected and leaves), is off-placed in the step-down unit (SDU), waits for an available bed elsewhere or is admitted to the ICU by bumping a patient who is present in the ICU (see e.g. [2]).
- Depending on the patient type, the arriving patient is admitted to the ICU by creating an over-bed (i.e. an ICU bed that was not staffed), is rejected or is overflowed to another ICU in the region that is also part of the queueing system (see e.g. [16, 39]).

In reality, it depends on the situation what solution is applicable when all ICU beds are occupied upon arrival of a new ICU patient. A further discussion is contained in Chapter 4, which studies a queueing network with an ICU and SDU.

Finally, it is noted that the information that is provided in this section is mainly based on [24, 32]. These sources can also be consulted for a more extensive discussion of queueing theory. Moreover, Chapter 2 discusses some aspects from queueing theory that are relevant for the research project.

1.1.2 Overflow systems

In queueing networks with finite capacities, it may occur that an arriving job cannot immediately be served at the primary station, because all servers are occupied. In this case, it is often assumed that the job waits until a server becomes available or is rejected and leaves the system. Instead, it may also be an option that the service request is handled by a server at a secondary station. This mechanism is known as overflow and is further discussed in this section, where the main focus is on the two overflow systems that are depicted in Figure 1.

First of all, Figure 1a depicts a simple and generic overflow system, which has been studied under different assumptions (e.g. regarding the service parameters) in [27, 28, 60, 61] (see also Remark 6 in Section 2.7). Arriving type 1 jobs that find (primary) station 1 fully occupied are overflowed to (secondary) station 2. If all servers at station 2 are also occupied, arriving type 1 jobs are rejected and leave the system. Next to arrivals of overflowed type 1 jobs, direct arrivals of type 2 jobs at station 2 can also be observed. These jobs are immediately rejected and lost if they find station 2 congested upon arrival.

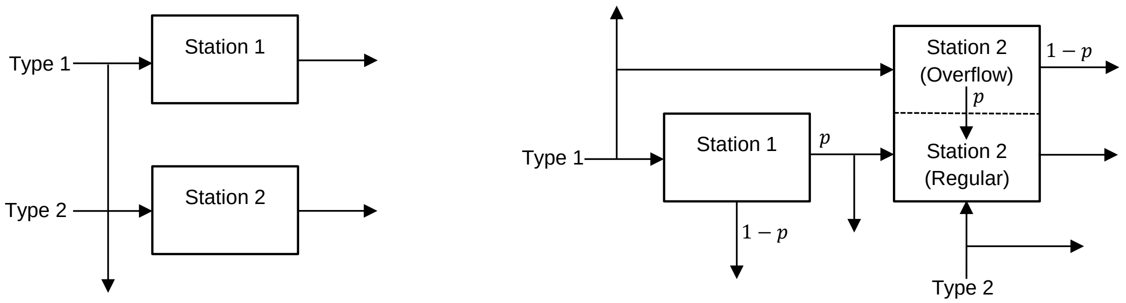
Secondly, Figure 1b depicts another overflow system, which can be seen as similar to the overflow system in Figure 1a. The main difference is that a type 1 job might also require service at station 2 after service completion at station 1 (or station 2 in case of overflow). More specifically, after a type 1 job at station 1 (or overflowed type 1 job at station 2) finishes service, it goes to (or stays at) station 2 with probability $p \in (0, 1]$, while it leaves the system with probability $1 - p$ (note that the overflow systems are equivalent if $p = 0$). In this sense, the overflow system in Figure 1b can be said to have a serial structure. Similarly, the system in Figure 1a can be said to have a parallel structure, since station 2 is only visited by type 1 jobs if they are overflowed.

When analyzing the overflow systems in Figure 1, it is desired to determine certain performance measures, such as blocking (or loss or rejection) probabilities or the average number of jobs in service. For this purpose, it would be useful if the joint steady-state distribution of the number of jobs in service can (efficiently) be computed. To this end, it would be of particular interest if the steady-state distribution has a so-called product-form solution, which is a particular closed-form solution (see also Section 2.4.1).

In this respect, it is useful to make a distinction between two possibilities when the service of overflowed type 1 jobs at station 2 is considered:

- When overflowed type 1 jobs are served at (secondary) station 2, they also complete the service at this station, even if a server at (primary) station 1 becomes available.
- An overflowed type 1 job at (secondary) station 2 is switched to (primary) station 1 as soon as a place at station 1 becomes available.

The latter assumption, which is known from teletraffic theory under the names of (immediate) repacking or call packing, has already been associated with product-form results (see e.g. [10, 11, 26, 60, 61]). Under the former assumption, in contrast, no product-form solution for the joint steady-state distribution of the number of jobs in the system can be expected (see also Example 12 in Section 3.4.2). Instead, a numerical algorithm, simulation or an approximation method could be



(a) Parallel structure

(b) Serial structure

Figure 1: Two queueing networks with overflow

used to determine the steady-state distribution and/or related performance measures.

Finally, two comments regarding the service times are made:

- In order to derive analytical results, it is often desirable to assume that the service times are exponentially distributed. However, this is often not the most realistic assumption, since many service time distributions are not very well approximated by the exponential distribution. Therefore, it would be of interest if the steady-state distribution does not depend on the service time distributions other than through their means. This appealing property can be referred to as insensitivity (see also Section 2.6).
- Typically, the service requirements of type 1 and type 2 jobs are not the same. Therefore, type 1 jobs may have a different mean service time than type 2 jobs. Moreover, it could well be that one of the stations is more suited to provide a specific service. This means that overflowed jobs at station 2 may have a different mean service time than non-overflowed jobs at station 1. Finally, in the system with serial structure, the mean service time of non-overflowed type 1 jobs at station 2 may also differ from that of other type 1 jobs. Using the notation that is introduced in Section 3.2, this can be summarized as $1/\mu^1_{11} \neq 1/\mu^1_{12} \neq 1/\mu^1_{22} \neq 1/\mu^2_{22}$.

1.2 Objectives

Overflow systems can be found in a variety of settings, such as telecommunications, call centers and health care. As a consequence, there is a large amount of literature that is concerned with overflow systems (see e.g. [3, 9, 10, 11, 16, 26, 27, 28, 29, 39, 49, 52, 60, 61, 62, 63] and references therein). Most of these studies consider overflow systems with a parallel structure, which means that transfers from the primary station to the secondary (overflow) station are not incorporated.

In this report, the main focus is on overflow systems with a serial structure, and in particular the overflow system that is depicted in Figure 1b. This system is studied both with and without the assumption of call packing. The following objectives are then aimed for:

- Determine the joint steady-state distribution of the number of jobs that are present in the overflow system, either by deriving a product-form solution or by using a numerical algorithm.

- Determine which blocking probabilities can be of interest, study how these can be computed, and examine what effect the assumption of call packing has on these.
- Examine whether (and under which conditions) the overflow system is insensitive to the service time distributions.
- Illustrate a possible application of the overflow system by describing how the overflow system (in adapted form) can be used to analyze the interaction between an ICU and SDU.

1.3 Outline of report

This report consists of three main parts, which are each covered in a separate chapter. The structure of this report is therefore as follows:

- Chapter 2 contains the theoretical background. This chapter discusses some concepts, models and methods from queueing theory that are relevant for the research project. These are, among others, stochastic processes, Markov chains, queueing networks, product forms, numerical algorithms to determine the steady-state distribution of a Markov chain, three approaches that can be taken to compute blocking probabilities (using the steady-state distribution, simulation and approximation methods), insensitivity and overflow systems.
- In Chapter 3, a two-station overflow system with serial structure is studied. First of all, a formal model description is provided. Besides that, the steady-state distribution is determined, and blocking probabilities are computed. Finally, the feature of insensitivity is studied.
- In Chapter 4, a possible application to ICU-SDU modelling is studied. Literature regarding ICUs and SDUs is described, and a product-form solution for the steady-state distribution is provided. Moreover, it is discussed how this steady-state distribution can be useful to obtain insight into the blocking probabilities and other related performance measures.
- Chapter 5 contains a conclusion, which summarizes the main findings of the research project.

Finally, it is noted that the appendix contains some supplementary material. More specifically, additional information regarding the proofs, blocking probabilities and simulation model is provided in Appendix A, Appendix B and Appendix C, respectively. Besides that, Appendix D contains the Matlab code that is used.

Chapter 2

Theoretical background

In this chapter, some concepts, models and methods from queueing theory that are relevant for the research project are discussed. For a more thorough introduction to queueing theory, the reader can consult, for example, [1, 32, 53, 55, 57].

2.1 Outline of chapter

This chapter is organized as follows:

- Section 2.2 discusses stochastic processes, and in particular Markov chains, which are frequently encountered in queueing theory.
- In Section 2.3, a brief description of queueing networks is provided. In particular, the characteristics of the queueing networks that are studied in this report are mentioned.
- Section 2.4 describes how the joint steady-state distribution of the number of jobs in a queueing network could be determined. Two approaches are discussed: deriving a product-form solution and using a numerical algorithm.
- Section 2.5 focuses on the determination of blocking probabilities. It is described how blocking probabilities can be computed when the steady-state distribution is known. Besides that, simulation and approximation methods are briefly discussed.
- In Section 2.6, the feature of insensitivity is discussed.
- In Section 2.7, an overflow system with parallel structure is considered. For this system, it is described how the steady-state distribution and blocking probabilities can be determined and under which conditions the insensitivity property holds. Besides that, a brief discussion of literature regarding overflow systems is included in this section.

Finally, it is noted that throughout this chapter several examples are provided. In these examples, the following queueing systems are discussed: the Erlang loss system (Examples 2, 3, 7 and 10), a two-station tandem queue with jump-over blocking (Examples 4, 6, 8 and 9) and the Jackson network (Example 5).

2.2 Stochastic processes and Markov chains

This section discusses stochastic processes, and in particular Markov chains, since they play an important part in the analysis of queueing systems. The information that is provided is mainly based on the book of Stewart [53] (chapter 9) and the book of Karlin and Taylor [33] (chapters 1 and 2), although slightly different notation is used. For a more extensive discussion of the topics that are discussed in this section, the reader is also referred to these sources.

2.2.1 Formal definitions

First of all, a stochastic process is defined as follows:

Definition 1 (Stochastic process, [53], p. 253)**.** A stochastic process is a family of random variables $\{X(t),\ t \in T\}$. Here, T is called the index set or parameter space. It can be either discrete (e.g. $T = \{0, 1, ...\}$) or continuous (e.g. $(T = [0, \infty) = \{t \mid t \geq 0\})$.

The parameter t can often be interpreted as time. Hence, $X(t)$ then denotes the value that the random variable assumes at time t. The values of $X(t)$ are called states and can be one-dimensional, but also multi-dimensional. The set of all possible states that can occur is referred to as the state space and is commonly denoted by $\boldsymbol{S}$ or S.

An important type of stochastic process is a Markov process, and in particular a Markov chain. A stochastic process $\{X(t),\ t \in T\}$ is a Markov process if it possesses the so-called Markov property. In words, this property states that the future behaviour of the process only depends on the current state and is not altered by additional information concerning its past history. A Markov process that has a finite or denumerable state space is then called a Markov chain.

When Markov chains are defined, a distinction is generally made between discrete-time Markov chains, where $T = \{0, 1, ...\}$, and continuous-time Markov chains, where $T = [0, \infty)$. Below, formal definitions are given.

First of all, the definition of a discrete-time Markov chain is as follows:

Definition 2 (Discrete-time Markov chain, [53], p. 195)**.** A stochastic process $\{X_n,\ n = 0, 1, ...\}$ is a discrete-time Markov chain if it satisfies the following relationship for all natural numbers n and all states $x_0, ..., x_{n+1} \in \boldsymbol{S}$:

$$P[X_{n+1} = x_{n+1} \mid X_n = x_n,\ X_{n-1} = x_{n-1}, ...,\ X_0 = x_0] = P[X_{n+1} = x_{n+1} \mid X_n = x_n] \tag{1}$$

Secondly, a continuous-time Markov chain is defined as follows:

Definition 3 (Continuous-time Markov chain, [53], p. 253)**.** A stochastic process $\{X(t),\ t \geq 0\}$ is a continuous-time Markov chain if it satisfies the following relationship for all integers $n \geq 1$, all states $x_0, ..., x_{n+1} \in \boldsymbol{S}$ and for any sequence $t_0, t_1, ..., t_n, t_{n+1}$ such that $t_0 < t_1 < ... < t_n < t_{n+1}$:

$$\begin{aligned} P[X(t_{n+1}) = x_{n+1} \mid X(t_n) = x_n,\ X(t_{n-1}) = x_{n-1}, ...,\ X(t_0) = x_0]& \\ = P[X(t_{n+1}) = x_{n+1} \mid X(t_n) = x_n]& \end{aligned} \tag{2}$$

In the sequel, the main focus lies on continuous-time Markov chains with a finite state space, as these are of particular interest for this research project. Moreover, the continuous-time Markov chains are assumed to be (time-)homogeneous and irreducible as defined below. To this end, let $p(\mathbf{n}, \mathbf{n}'; s, t)$ denote the conditional probability that the Markov chain is in state $\mathbf{n}'$ at time t given that it is in state $\mathbf{n}$ at time s (for $\mathbf{n}, \mathbf{n}' \in \boldsymbol{S}$, $s, t \in [0, \infty)$ and $t \geq s$), that is:

$$p(\mathbf{n}, \mathbf{n}'; s, t) = P[X(t) = \mathbf{n}' \mid X(s) = \mathbf{n}] \tag{3}$$

A homogeneous continuous-time Markov chain is then defined as follows:

Definition 4 (Homogeneous continuous-time Markov chain, [53], pp. 194, 253-254)**.** A continuous-time Markov chain is said to be homogeneous or time-homogeneous when the transitions are independent of the elapsed time, that is, when the transition probability $p(\mathbf{n}, \mathbf{n}'; s, t)$ does not depend on the values of t and s, but only on their difference $\tau = t - s$.

In order to simplify notation, the transition probability of a homogeneous continuous-time Markov chain can therefore be written as $p(\mathbf{n}, \mathbf{n}'; \tau)$ (for $\mathbf{n}, \mathbf{n}' \in \boldsymbol{S}$ and $\tau \in [0, \infty)$), where:

$$p(\mathbf{n}, \mathbf{n}'; \tau) = P[X(s + \tau) = \mathbf{n}' \mid X(s) = \mathbf{n}] \qquad \text{for all } s \geq 0 \tag{4}$$

Subsequently, an irreducible continuous-time Markov chain is defined as follows:

Definition 5 (Irreducibility, [53], p. 260)**.** A homogeneous, continuous-time Markov chain is said to be irreducible if, for any two states $\mathbf{n}$ and $\mathbf{n}'$ ($\mathbf{n}, \mathbf{n}' \in \boldsymbol{S}$), there exists real numbers $\tau_1 \geq 0$ and $\tau_2 \geq 0$ such that $p(\mathbf{n}, \mathbf{n}'; \tau_1) > 0$ and $p(\mathbf{n}', \mathbf{n}; \tau_2) > 0$.

The interactions between the states in a continuous-time Markov chain are specified by means of the transition rates. For a homogeneous continuous-time Markov chain, these transition rates (denoted by $q(\mathbf{n}, \mathbf{n}')$, $\mathbf{n}, \mathbf{n}' \in \boldsymbol{S}$) are defined as follows:

$$q(\mathbf{n}, \mathbf{n}') = \lim_{\Delta t \to 0} \left(\frac{p(\mathbf{n}, \mathbf{n}'; \Delta t)}{\Delta t} \right), \qquad \mathbf{n} \neq \mathbf{n}' \tag{5}$$

$$q(\mathbf{n}, \mathbf{n}) = \lim_{\Delta t \to 0} \left(\frac{p(\mathbf{n}, \mathbf{n}; \Delta t) - 1}{\Delta t} \right) \tag{6}$$

For $\mathbf{n} \neq \mathbf{n}'$, $q(\mathbf{n}, \mathbf{n}')$ can be interpreted as the rate at which transitions occur from state $\mathbf{n}$ to state $\mathbf{n}'$. Besides that, $q(\mathbf{n}, \mathbf{n})$ is equal to the negative of the sum of the transition rates from $\mathbf{n}$ to all other states $\mathbf{n}'$ (i.e. $\mathbf{n}' \neq \mathbf{n}$), that is, $q(\mathbf{n}, \mathbf{n}) = -\sum_{\mathbf{n}' \in \boldsymbol{S} \setminus \mathbf{n}} q(\mathbf{n}, \mathbf{n}')$. It is also noted that the values of the transition rates are often arranged in a matrix. This matrix is known as the infinitesimal generator matrix or transition rate matrix and is generally denoted by Q.

Finally, as an illustration, the following example describes a single-station queueing system with N servers that can be represented by a homogeneous, irreducible, continuous-time Markov chain. This queueing system is known as Erlang loss system or, using Kendall's notation (see e.g. [1], p. 24, for a description of this notation), $M|M|N|N$ queue (see e.g. [1, 32, 47, 54]).

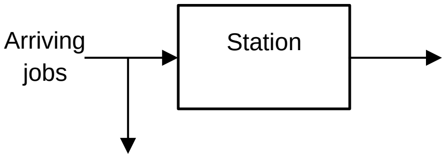


Figure 2: Erlang loss system

Example 2 (Erlang loss system: Model description)**.** Consider the queueing system that is depicted in Figure 2. The system consists of one station with N servers. Jobs arrive to this station according to a Poisson process with rate λ, which means that the time between arrivals is exponentially distributed with mean $1/\lambda$. If an arriving job finds all N servers occupied upon arrival, it is rejected and lost. Otherwise, the arriving job is served by one of the servers, where the service times are assumed to be exponential with rate μ.

Now, let the random variable $n(t)$ denote the number of jobs in the system at time t. The process $\{n(t),\ t \geq 0\}$ is then a continuous-time Markov chain with state space $\boldsymbol{S_{Er}} = \{0, 1, ..., N\}$ and transition rates $q_{Er}(n, n')$, $n, n' \in \boldsymbol{S_{Er}}$, equal to:

$$q_{Er}(n, n') = \begin{cases} \lambda & n' = n+1 \\ n\mu & n' = n-1 \\ -(\lambda + n\mu) & n' = n < N \\ -n\mu & n' = n = N \\ 0 & \text{else} \end{cases} \tag{7}$$

It is noted that the Erlang loss system is a special case of a birth-death process. More specifically, for each state $n \in \boldsymbol{S_{Er}}$, there is either an arrival (i.e. a birth) with rate $\lambda 1_{\{n<N\}}$, which leads to state $n+1$, or a departure (i.e. a death) with rate $n\mu$, which leads to state $n-1$. See, for example, [32] for a wider discussion on birth-death processes.

In Examples 3, 7 and 10, the Erlang loss system system is further discussed.

2.2.2 Steady-state distribution

When a Markov chain is analyzed, it is often of interest to determine the steady-state distribution $\pi = (\pi(\mathbf{n}),\ \mathbf{n} \in \boldsymbol{S})$, which is also known as long-run or equilibrium distribution. For $\mathbf{n} \in \boldsymbol{S}$, the steady-state probability $\pi(\mathbf{n})$ generally has the following two interpretations ([53], p. 238):

- $\pi(\mathbf{n})$: The probability that a random observer sees the Markov chain in state $\mathbf{n}$ after the process has evolved over a long period of time.
- $\pi(\mathbf{n})$: The long-run proportion of time the Markov chain spends in state $\mathbf{n}$.

Moreover, Section 2.5.1 discusses another interpretation that can be thought of when the Markov chain represents a queueing system with Poisson arrivals.

The Markov chains that are considered in this report are homogeneous, irreducible, continuous-time Markov chains with a finite state space $\boldsymbol{S}$ and transition rates $q(\mathbf{n}, \mathbf{n}')$, $\mathbf{n}, \mathbf{n}' \in \boldsymbol{S}$ (see also Section 2.2.1). For such Markov chains, there exists a unique steady-state distribution, which has strictly positive elements (i.e. $\pi(\mathbf{n}) > 0$ for all $\mathbf{n} \in \boldsymbol{S}$) and may be obtained as the solution to the following global balance equations:

$$\sum_{\mathbf{n}' \in \boldsymbol{S} \setminus \mathbf{n}} \pi(\mathbf{n}) q(\mathbf{n}, \mathbf{n}') = \sum_{\mathbf{n}' \in \boldsymbol{S} \setminus \mathbf{n}} \pi(\mathbf{n}') q(\mathbf{n}', \mathbf{n}), \qquad \mathbf{n} \in \boldsymbol{S}, \tag{8}$$

subject to the condition that the probabilities sum to one (i.e. $\sum_{\mathbf{n} \in \boldsymbol{S}} \pi(\mathbf{n}) = 1$).

It is noted that the global balance equations (8) have the interpretation that for each $\mathbf{n} \in \boldsymbol{S}$ the flow out of state $\mathbf{n}$ is equal to the flow into state $\mathbf{n}$. More specifically, the left-hand side of (8) represents the flow out of state $\mathbf{n}$, while the right-hand side represents the flow into state $\mathbf{n}$.

Finally, the section ends with a continuation of Example 2.

Example 3 (Erlang loss system: Steady-state distribution)**.** Consider the Erlang loss system as introduced in Example 2. It can then be of interest to find the steady-state distribution of the number of jobs that are present in the system, denoted by $\pi_{Er} = (\pi_{Er}(n),\ n \in \boldsymbol{S_{Er}})$. As discussed above, the steady-state distribution should satisfy the global balance equations. For $n \in \boldsymbol{S_{Er}} = \{0, 1, ..., N\}$, these global balance equations are given by:

$$\lambda \pi_{Er}(n) 1_{\{n<N\}} + n\mu \pi_{Er}(n) 1_{\{n>0\}} = (n+1)\mu \pi_{Er}(n+1) 1_{\{n<N\}} + \lambda \pi_{Er}(n-1) 1_{\{n>0\}} \tag{9}$$

In fact, in this case, an explicit closed-form expression for the steady-state distribution can be obtained. To this end, it is noted that from the global balance equations (9) it can be seen that the following more detailed equalities must be satisfied (see e.g. [32]):

$$\lambda \pi_{Er}(n-1) = n\mu \pi_{Er}(n), \quad n \in \{1, ..., N\} \tag{10}$$

From these equations, in turn, the following expression for the steady-state distribution is readily verified (see e.g. [1]):

$$\pi_{Er}(n) = \left(\sum_{k=0}^{N} \frac{1}{k!} \left(\frac{\lambda}{\mu} \right)^k \right)^{-1} \frac{1}{n!} \left(\frac{\lambda}{\mu} \right)^n, \qquad n \in \boldsymbol{S_{Er}} = \{0, 1, ..., N\} \tag{11}$$

2.3 Queueing networks

Instead of a single-station queueing system (e.g. as in Example 2 in Section 2.2.1), it can also be of interest to consider a queueing network consisting of multiple stations that are in some sense connected. An advantage of this is that it enables us to take into account the interaction and interdependence between the stations. For example, this could be useful when modelling hospital wards, since patients often visit more than one ward during a hospital stay. Another example of an application is an assembly line, in which a product visits several workstations in sequence. At each workstation, one or more operations are executed in order to assemble the product (see e.g. [32]).

It can be noted that many different assumptions can be made when describing a queueing network. To name a few, the queueing network may be closed (i.e. no arrivals from and departures to the outside) or open (i.e. arrivals from and departures to the outside are possible), one or multiple job types may be distinguished, capacities may be limited or unlimited, and different assumptions regarding the arrival process and service time distributions can be made. Moreover, if a job finds a station fully occupied upon arrival, it can, among others, be rejected and leave the system, wait until it can be served or replace a job in service. Another possibility is that the job is overflowed, which is defined as follows:

Definition 6 (Overflow)**.** Jobs are said to be overflowed when they go to another, secondary station for the same service if the primary station is fully occupied.

As mentioned above, there is a large variety of queueing networks that can be studied. In this report, the main focus is on queueing networks with the following characteristics:

- The queueing network consists of two stations, which each have a finite number of servers.
- Two types of jobs arrive to the network. Type 1 jobs arrive at station 1 according to a Poisson process, while type 2 jobs arrive at station 2 according to a Poisson process.
- At a station, jobs are served by one of the servers, where the service times are assumed to be exponential. It is also studied, though, to what extent the exponentiality assumption is necessary (see also Section 2.6).
- After service completion at station i $(i = 1, 2)$, type t jobs $(t = 1, 2)$ leave the system with probability $p^t_{i,0} \in [0, 1]$, while they are routed to the other station with probability $1 - p^t_{i,0}$ (provided that a server is available).
- Jobs that find all servers at a station occupied upon a service request are overflowed to the other station or they are blocked and leave the system (or 'jump over' the station as in Example 4 below).
- The service rates may depend on the job type, the station at which the job is served and whether or not the job is overflowed.

It is noted that queueing networks with the aforementioned characteristics can be represented by a homogeneous, irreducible, finite, continuous-time Markov chain. This is illustrated by the following example that describes a two-station tandem queue with jump-over blocking. In Remark 2 below, it is discussed how this system is related to the overflow system that is studied in Chapter 3.

Example 4 (Tandem queue with jump-over blocking: Model description)**.** Consider the queueing network that is depicted in Figure 3. The network consists of two stations: station 1 with a capacity of N_1 servers and station 2 with N_2 servers. Type 1 jobs arrive at station 1

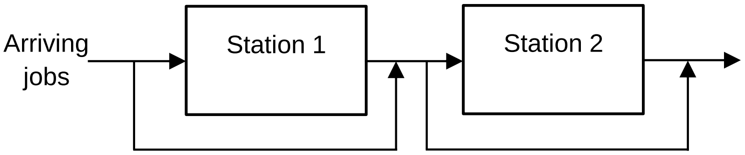


Figure 3: A two-station tandem queue with jump-over blocking mechanism

according to a Poisson process with rate λ_1. For simplicity, arrivals of type 2 jobs at station 2 are left out of account (i.e. the arrival rate of type 2 jobs λ_2 is assumed to be zero).

If station 1 is not fully occupied, arriving type 1 jobs are served by one of the N_1 servers, where the service times are assumed to be exponential with rate μ_1. After service completion, the jobs then go to station 2 (i.e. $p^1_{1,0} = 0$ and $p^1_{1,2} = 1$). Here, the jobs receive a service from one of the N_2 servers, provided that they are accepted. After a service time, which is exponentially distributed with rate μ_2, the jobs leave the system (i.e. $p^1_{2,0} = 1$ and $p^1_{2,1} = 0$).

It might also occur that a job that arrives at station 1 or station 2 finds all servers occupied. In this case, a jump-over or skipping blocking mechanism is assumed (see also Remark 1 below). More specifically, an arriving job at station 1 'jumps over' station 1 and immediately goes to station 2 if all N_1 servers are occupied. Similarly, a job that arrives at station 2 after completing service at station 1 (or jumping over station 1) jumps over station 2 to the outside if station 2 is fully occupied.

Now, let $\mathbf{n}(t) = (n_1(t), n_2(t))$ denote the state of the system at time t, where $n_i(t)$ denotes the number of jobs at station i at time t $(i = 1, 2)$. The process $\{\mathbf{n}(t),\ t \geq 0\}$ is then a continuous-time Markov chain with state space $\boldsymbol{S_{tq}}$, which is as follows:

$$\boldsymbol{S_{tq}} = \{(n_1, n_2) \mid 0 \leq n_i \leq N_i,\ i = 1, 2\} \tag{12}$$

Moreover, the transition rates $q_{tq}(\mathbf{n}, \mathbf{n}')$, $\mathbf{n}, \mathbf{n}' \in \boldsymbol{S_{tq}}$, are given by:

$$q_{tq}(\mathbf{n}, \mathbf{n}') = \begin{cases} \lambda_1 & (n_1, n_2)' = (n_1 + 1, n_2) \\ n_1\mu_1 & (n_1, n_2)' = (n_1 - 1, n_2 + 1) \\ n_1\mu_1 1_{\{n_2 = N_2\}} & (n_1, n_2)' = (n_1 - 1, n_2) \\ \lambda_1 1_{\{n_1 = N_1\}} & (n_1, n_2)' = (n_1, n_2 + 1) \\ n_2\mu_2 & (n_1, n_2)' = (n_1, n_2 - 1) \\ 0 & \text{else} \end{cases} , \qquad \mathbf{n} \neq \mathbf{n}' \tag{13}$$

$$q_{tq}(\mathbf{n}, \mathbf{n}) = -\sum_{\mathbf{n}' \in \boldsymbol{S_{tq}} \setminus \mathbf{n}} q_{tq}(\mathbf{n}, \mathbf{n}') \tag{14}$$

The tandem queue with jump-over blocking is further studied in Examples 6, 8 and 9.

Finally, two remarks regarding Example 4 are made.

Remark 1 (Jump-over blocking in literature). The description of the system in Example 4 is based on the two-station tandem queue with jump-over blocking that is described in [58] (pp. 61-62). Next to this reference, the jump-over blocking mechanism has found several other applications in literature. For example, in [56], a closed queueing network with jump-over blocking is studied. More precisely, a job that requests service at a station, but finds all servers at this station occupied, is immediately routed to another station according to the routing probabilities, that is, the job 'jumps over' the station as if it is served with infinite speed. Besides that, [5] also considers a queueing network with jump-over blocking, although instead of jumping over the term skipping is used. This queueing network is more general than the two-station tandem queue in Example 4, since it allows for possibly more than two stations and direct arrivals at each of the stations. Other references in which a queueing network with jump-over or skipping blocking mechanism can be found include [14, 43].

Remark 2 (Example 4: Relation to overflow system in Chapter 3). In a sense, the tandem queue that is described in Example 4 can be seen as similar to the overflow system that is studied in Chapter 3. More specifically, in Chapter 3, it is assumed that jobs that complete service at station 1 and are routed to station 2 are rejected if there is no server at station 2 available. This can also be seen as if these jobs jump over station 2. However, jobs that arrive at station 1 do not jump over station 1 if the station is fully occupied. Instead, these jobs are overflowed to station 2.

2.4 Determining the steady-state distribution

In this report, we consider queueing networks that can be represented by a homogeneous, irreducible, finite, continuous-time Markov chain. For these networks, we are interested in determining the joint steady-state distribution of the number of jobs that are present, from which related performance measures (e.g. blocking probabilities) can be computed. Therefore, this section discusses two approaches that can be thought of when determining the joint steady-state distribution of the number of jobs in a queueing network that is represented by a homogeneous, irreducible, finite, continuous-time Markov chain.

First of all, Section 2.4.1 discusses steady-state distributions that have a product-form solution. Secondly, Section 2.4.2 describes numerical algorithms that can be used to find the steady-state distribution.

2.4.1 Product-form solution

Over the past decades, a great deal of attention has been devoted to so-called product-form solutions for the joint steady-state distributions of the number of jobs in a queueing network. The first product-form results are generally attributed to R.R.P. Jackson [31] and J.R. Jackson [30]. In the latter reference, the queueing network that is described in Example 5 below is considered.

Example 5 (Jackson network: Product form)**.** Consider a queueing network that consists of multiple, say K, stations, where the ith station has N_i servers and infinite waiting room. Jobs from outside the system arrive at station i according to a Poisson process with rate λ_i. Besides that, arrivals at a station may also come from one of the stations. More specifically, when a job finishes service at station i, it is routed to station j with probability p_{ij}, while it leaves the system with probability $1 - \sum_{k=1}^{K} p_{ik}$. At station i, jobs are served by one of the N_i servers, where the service times are assumed to be exponentially distributed with mean $1/\mu_i$. Arriving jobs that find all servers at a station occupied join a queue and are served in order of arrival (i.e. first come, first served). The state of the system is then given by $(n_1, ..., n_K)$, where n_i denotes the number of jobs that are present at station i (either in service or waiting).

It is then shown by J.R. Jackson [30] that the joint steady-state distribution of the number of jobs at the stations, denoted by $\pi(n_1, ..., n_K)$, is given by the product of the steady-state distributions of the number of jobs at the individual stations, denoted by $\pi_i(n_i)$, $i = 1, ..., K$. This means that the joint steady-state distribution can be written as follows:

$$\pi(n_1, ..., n_K) = \prod_{i=1}^{K} \pi_i(n_i) \tag{15}$$

In this case, the joint steady-state distribution can thus be factorized into the steady-state distributions for the individual stations as if they were in isolation. This is reflected in the name product-form solution or, in short, product form (see also e.g. [57], p. 65).

Since the product-form result in [30], many queueing networks have been shown to exhibit a product form. These include, for example, Gordon-Newell networks, which are studied by Gordon and Newell [22], and BCMP networks, which are introduced by Baskett et al. [8].

Here, it is noted that the term product form is also used when a factorization to marginal probabilities as in Example 5 is not possible (see e.g. the descriptions of product forms in [5, 8, 12, 17, 18, 57]). For example, dependence between the components may generally still be included into a normalizing constant. Besides that, more general expressions are sometimes also referred to as product-form solutions. For example, in [12] (p. 29), it is mentioned that a “closed-form distribution that can be obtained from the transition rates” may also be called a product-form distribution.

The expressions for the steady-state distributions that are derived in this report (in particular, those in Theorems 1, 2 and 4) are of product form in the sense that the joint steady-state distribution factorizes into functions for the individual components, up to normalization. More specifically, we consider queueing networks for which the state description is given by a multi-dimensional vector $\mathbf{n}$, say $\mathbf{n} = (n_1, ..., n_R)$ for some positive integer R. The expressions for the steady-state distributions are then of the following form:

$$\pi(n_1, ..., n_R) = c \prod_{i=1}^{R} \pi_i(n_i) \tag{16}$$

Here, π is the joint steady-state distribution, π_i the function for component i, $i = 1, ..., R$, and c a normalizing constant, which is such that the steady-state probabilities sum to one.

As discussed in Section 2.2.2, the steady-state distribution may be found as the unique solution to the global balance equations (8) subject to the condition that the probabilities sum to one. However, it is often difficult, if not impossible, to obtain a product-form solution by solving the global balance equations directly. As a consequence, for most of the product-form solutions that are available, more detailed subequations of the global balance equations are also satisfied. Examples of such subequations include detailed balance equations, which are related to reversibility (see e.g. [34, 36]), job-local-balance equations, which are related to insensitivity (see Section 2.6) and station balance equations, which are defined as follows:

Definition 7 (Station balance, [57], p. 63)**.** For all states $\mathbf{n} \in \boldsymbol{S}$, the rate out of state $\mathbf{n}$ due to a departure from a station is equal to the rate into state $\mathbf{n}$ due to an arrival at that station. Here, the outside of the system is also seen as a station, which is referred to as station 0.

Station balance thus has the natural interpretation that the physical outrate and inrate at a station are balanced. If multiple job types and/or overflowed and non-overflowed are distinguished, it might also be possible to verify a balance between physical outrate and inrate for each of these job classes separately. In line with [58] (p. 11), this leads to the following notion of balance, which is referred to as class balance:

Definition 8 (Class balance)**.** For all states $\mathbf{n} \in \boldsymbol{S}$, the rate out of state $\mathbf{n}$ due to a departure of a job that belongs to some class is equal to the rate into state $\mathbf{n}$ due to an arrival of a job of the same class.

In Sections 3.4.1 and 4.5, class balance is further illustrated. More specifically, it is shown that global balance is satisfied by verifying specific class balances.

Finally, the section ends with a continuation of Example 4. This example shows that the two-station tandem queue with jump-over blocking exhibits a product form.

Example 6 (Tandem queue with jump-over blocking: Product form)**.** For the two-station tandem queue with jump-over blocking as introduced in Example 4, there exists a product-form solution for the joint steady-state distribution of the number of jobs at station 1 and station 2, denoted by $\pi_{tq} = (\pi_{tq}(n_1, n_2),\ (n_1, n_2) \in \boldsymbol{S_{tq}})$. More specifically, with c_{tq} a normalizing constant, the following product-form solution for the steady-state distribution applies (see [58], p. 62):

$$\pi_{tq}(n_1, n_2) = c_{tq} \frac{1}{n_1!} \left(\frac{\lambda_1}{\mu_1}\right)^{n_1} \frac{1}{n_2!} \left(\frac{\lambda_1}{\mu_2}\right)^{n_2}, \quad (n_1, n_2) \in \boldsymbol{S_{tq}} \tag{17}$$

As shown in [58] (p. 62), the product form can be proven by verifying that the global balance equations are satisfied for all $(n_1, n_2) \in \boldsymbol{S_{tq}}$ when the product form (17) is substituted. For illustrative purposes, the global balance equations are also provided below. For $(n_1, n_2) \in \boldsymbol{S_{tq}}$, the global balance equations are as follows:

$$\left\{\begin{array}{l}\pi_{tq}(n_1,n_2)n_1\mu_1 1_{\{n_1>0\}}+\\ \pi_{tq}(n_1,n_2)n_2\mu_2 1_{\{n_2>0\}}+\\ \pi_{tq}(n_1,n_2)\lambda_1(1_{\{n_1<N_1\}}+1_{\{n_1=N_1\}}1_{\{n_2<N_2\}})\end{array}\right\} \qquad \begin{array}{r}(18.1)\\(18.2)\\(18.3)\end{array}$$

$$= \qquad (18)$$

$$\left\{\begin{array}{l}\pi_{tq}(n_1-1,n_2)\lambda_1 1_{\{n_1>0\}}+\\ \pi_{tq}(n_1+1,n_2-1)(n_1+1)\mu_1 1_{\{n_1<N_1\}}1_{\{n_2>0\}}+\pi_{tq}(n_1,n_2-1)\lambda_1 1_{\{n_1=N_1\}}1_{\{n_2>0\}}+\\ \pi_{tq}(n_1,n_2+1)(n_2+1)\mu_2 1_{\{n_2<N_2\}}+\pi_{tq}(n_1+1,n_2)(n_1+1)\mu_1\lambda_1 1_{\{n_1<N_1\}}1_{\{n_2=N_2\}}\end{array}\right\} \qquad \begin{array}{r}(18.1)'\\(18.2)'\\(18.3)'\end{array}$$

By substitution of the product form (17), it can now be verified that $(18.i) = (18.i)'$, $i = 1, 2, 3$. Here, it is noted that $(18.1) = (18.1)'$ represents station balance for station 1, $(18.2) = (18.2)'$ for station 2 and $(18.3) = (18.3)'$ for station 0 (i.e. the outside). From this, it then follows that $(18.1) + (18.2) + (18.3) = (18.1)' + (18.2)' + (18.3)'$, which means that the global balance equations (18) are satisfied.

Remark 3 (Jump-over blocking: Product forms). As illustrated by Example 6, it is possible to obtain a product-form solution for the joint steady-state distribution of the number of jobs in a two-station tandem queue with jump-over blocking. Moreover, the jump-over blocking mechanism has found several other product-form applications in literature as well, also for more general structures (see e.g. the references that are mentioned in Remark 1 in Section 2.3).

2.4.2 Numerical algorithm

In Section 2.4.1, it is discussed that in some cases the steady-state probabilities $\pi(\mathbf{n})$, $\mathbf{n} \in \boldsymbol{S}$, can be determined by obtaining a product-form solution. However, it is not always possible to derive a product-form solution. A suitable, though generally computationally less attractive, alternative could then be to use a numerical algorithm to obtain the steady-state distribution. This approach is discussed in this section. The information that is provided is mainly based on the book of Quarteroni et al. [45] (chapters 3 and 4) and the book of Stewart [53] (chapter 10).

As discussed in Section 2.2.2, the steady-state probabilities $\pi(\mathbf{n})$, $\mathbf{n} \in \boldsymbol{S}$, should be positive numbers that sum to one and satisfy the global balance equations (8). In matrix form, this is equivalent to the following:

$$\pi Q = \vec{0}, \quad \pi > \vec{0} \;\text{ and }\; \pi e = 1 \tag{19}$$

Here, π is a row vector that contains the steady-state probabilities, while Q is the infinitesimal generator matrix as introduced in Section 2.2.1. Moreover, $\vec{0}$ is a row vector with the same length as π that contains a zero in every field. Similarly, e is a column vector with the same length as π that contains a one in every field.

In order to solve the system of equations (19), a numerical algorithm can be applied (see also Remark 5 at the end of this section). When categorizing these algorithms, a distinction is generally made between iterative methods and direct methods.

Algorithm 1 GTH algorithm for continuous-time Markov chains ([53], p. 299)

```
Require: $s \times s$ infinitesimal generator matrix $Q$
Let $A = Q^T$ and $a_{ij} = A(i,j)$, $i,j = 1,...,s$ (i.e. $a_{ij}$ is the transition rate from state $j$ into state $i$).
Step 1: Reduction step
for $i = 1,...,s-1$ do
    $a_{ik} = a_{ik} / \sum_{j=i+1}^{s} a_{ji}$,        for all $k > i$.
    $a_{jk} = a_{jk} + a_{ji}a_{ik}$,        for all $j,k > i$, $k \neq j$.
end for
Step 2: Backsubstitution step
$x_s = 1$.
for $i = s-1, s-2, ..., 1$ do
    $x_i = \sum_{j=i+1}^{s} a_{ij}x_j$.
end for
Step 3: Normalization step
norm $= \sum_{j=1}^{s} x_j$.
for i = 1,...,n do
    $\pi_i = x_i/\text{norm}$ ($\pi_i$ is component $i$ of the steady-state probability vector, $i = 1,...,s$).
end for
return Steady-state probability vector $\pi$.
```

Algorithm 2 Gauss-Seidel method for continuous-time Markov chains (based on [53], pp. 309-313)

```
Require: $s \times s$ infinitesimal generator matrix $Q$, an initial approximation of the steady-state probability vector $\pi_0$ and information for determining when to stop, such as a maximum number of iterations $itmax$ or a tolerance for a stopping test $tol$ (see also Remark 4 below).
Let $A = Q^T$ and $x_0 = (\pi_0)^T$.
Determine a strictly lower triangular matrix $L$, a strictly upper triangular matrix $U$ and a diagonal matrix $D$, such that $A = D - L - U$.
Determine the iteration matrix $B$ as follows: $B = (D-L)^{-1}U$.
Initialize number of iterations $k = 0$ and initial vector $x^{(0)} = x_0$.
while $k < itmax$ && convergence test(s) not satisfied do
    $x^{(k+1)} = Bx^{(k)}$.
    $x^{(k+1)} = x^{(k+1)}/||x^{(k+1)}||_1$ (i.e. normalize).
    $k = k + 1$.
end while
Set $\pi = x^T$.
return A $1 \times s$ vector $\pi$ that contains the steady-state probabilities of the Markov chain.
```

First of all, an iterative method requires an initial approximation of the solution vector or, if an initial approximation is not available, a guess or arbitrarily chosen vector. The idea of an iterative method is then to successively improve this approximation in each iteration until it eventually converges to the solution. If an iterative method is used, it should therefore be determined when convergence has occurred and the iterative process can be halted. This is further discussed in Remark 4 at the end of this section. Examples of iterative methods include the method of Jacobi, the Gauss-Seidel method and the method of successive overrelaxation.

Secondly, as opposed to iterative methods, direct methods attempt to go directly to the solution by executing a finite number of steps. Direct methods are often based on Gaussian elimination and related LU factorization.

For the overflow system that is studied in Chapter 3, two algorithms are applied to obtain the steady-state distribution. These are the Grassmann-Taksar-Heyman (GTH) algorithm (see Algorithm 1), which is a direct method, and the Gauss-Seidel method (see Algorithm 2), which is an iterative method. These algorithms are further discussed and compared in Section 3.4.2.

Finally, this section ends with two remarks.

Remark 4 (Iterative method: Stopping criteria). As mentioned above, if an iterative method is applied, it should be decided when the iterative process is halted. Ideally, the iterative process is stopped if the iterate $x^{(k)}$ is such that $||x^{(k)} - x|| < tol$, where $x = \pi^T$ is the solution vector, $|| \cdot ||$ a suitable norm (e.g. the 2-norm) and tol some prespecified tolerance. However, the solution vector x is not known in advance, which means that other stopping criteria should be devised. For this purpose, several stopping criteria can be thought of (see e.g. [45, 53] for a discussion). In this report, when applying the Gauss-Seidel method as given in Algorithm 2, the following stopping criteria are used. First of all, a maximum number of iterations $itmax$ is specified. Besides that, the iterative process is also halted if the following relative convergence test is satisfied (see [53], p. 319):

$$\max_i \left\{ \frac{|x_i^{(k)} - x_i^{(k-m)}|}{|x_i^{(k)}|} \right\} < tol \tag{20}$$

Here, tol is some prespecified tolerance, $x_i^{(k)}$ the ith element of the iterate $x^{(k)}$ and $x_i^{(k-m)}$ the ith element of the iterate $x^{(k-m)}$, where m can be chosen greater than one, so that iterates that are not successive can be compared. Finally, if the relative convergence test indicates that convergence has occurred, the following residual $resid$, which should be close to zero, is also checked:

$$resid = ||\pi Q||_2 \tag{21}$$

Remark 5 (Alternative approach). In this section, it is described how the steady-state distribution can be obtained by applying a numerical algorithm to solve the system of equations (19). Instead of solving $\pi Q = \vec{0}$, it may also be an option to discretize the continuous-time Markov chain. Subsequently, the steady-state distribution π can be obtained by solving $\pi P = \pi$, where P is the transition probability matrix of the discretized Markov chain. See, for example, [53] (pp. 285-287) for a further discussion.

2.5 Determining blocking probabilities

For queueing networks, several performance measures can be of interest, such as the utilization, throughput and expected number of busy servers. In this report, the performance measure of primary interest is the blocking probability (i.e. the probability that an arriving job finds all servers occupied and is rejected). Therefore, this section discusses several methods that may be applied to obtain (an approximation of) the blocking probability.

First of all, Section 2.5.1 describes how the blocking probabilities can be computed from the steady-state distribution. Next, Section 2.5.2 contains a short description of simulation. Finally, approximation methods are briefly discussed in Section 2.5.3.

2.5.1 Using the steady-state distribution

In this section, it is discussed how blocking probabilities can be determined when the steady-state distribution $\pi = (\pi(\mathbf{n}),\ \mathbf{n} \in \boldsymbol{S})$ is known. For example, this can occur when there exists a product-form solution for the steady-state distribution (see Section 2.4.1) or when the steady-state distribution is determined by using a numerical algorithm (see Section 2.4.2). Below, two cases are distinguished:

(i) Jobs arrive according to a Poisson process.

(ii) Jobs do not arrive according to a Poisson process.

Case (i): In this case, the Poisson Arrivals See Time Averages (PASTA) property of Poisson arrivals can be used to determine the blocking probability. This property says that the fraction of jobs that find the system in state $\mathbf{n}$ upon arrival is the same as the fraction of time that the system spends in state $\mathbf{n}$, which is equal to $\pi(\mathbf{n})$ ([1], p. 27). As a consequence, the probability that an arriving job is blocked can be computed by summing the steady-state probabilities $\pi(\mathbf{n})$ over the states $\mathbf{n} \in \boldsymbol{S}$ that lead to blocking. More specifically, let $\boldsymbol{S_B} \subseteq \boldsymbol{S}$ denote the set of states that lead to blocking. The blocking probability, denoted by B, can then be calculated as follows:

$$B = \sum_{\mathbf{n} \in \boldsymbol{S_B}} \pi(\mathbf{n}) \tag{22}$$

This is illustrated by the following examples.

Example 7 (Erlang loss system: Blocking probability)**.** Again, consider the Erlang loss system as studied in Examples 2 and 3. Because of the PASTA property of Poisson arrivals, it can immediately be concluded from the steady-state distribution (11) that the blocking probability, denoted by B_{Er}, is as follows:

$$B_{Er}(\lambda, \mu, N) = \pi_{Er}(N) = \left(\sum_{k=0}^{N} \frac{1}{k!} \left(\frac{\lambda}{\mu} \right)^k \right)^{-1} \frac{1}{N!} \left(\frac{\lambda}{\mu} \right)^N \tag{23}$$

The expression (23) is commonly referred to as Erlang loss formula (see e.g. [1]).

Example 8 (Tandem queue with jump-over blocking: Blocking probability, 1/2)**.** Consider the tandem queue with jump-over blocking mechanism as described in Examples 4 and 6. In this system, arriving jobs at station 1 are only blocked if both station 1 and station 2 are fully occupied. Because of the PASTA property, the blocking probability of arriving jobs, denoted by $B_{1,tq}$, can therefore be computed from the steady-state distribution (17) as follows:

$$B_{1,tq}(\lambda_1, \mu_1, \mu_2, N_1, N_2) = \pi_{tq}(N_1, N_2) = c_{tq} \frac{1}{N_1!} \left(\frac{\lambda_1}{\mu_1}\right)^{N_1} \frac{1}{N_2!} \left(\frac{\lambda_1}{\mu_2}\right)^{N_2} \tag{24}$$

Case (ii): In this case, the PASTA property cannot be used to determine the blocking probability. Instead, if the queueing network is represented by a continuous-time Markov chain, the blocking probability may be determined by computation of a so-called Palm probability (see e.g. [4, 14, 51]). For example, suppose we are interested in determining the probability that a job that moves from some station, say station i, to another station, say station j, finds station j fully occupied and is blocked, denoted by $B_{i,j}$. This means that $B_{i,j}$ is the conditional probability that station j is fully occupied given that a job moves from station i to station j.

The blocking probability $B_{i,j}$ is then given by the Palm probability that is determined as follows. First of all, let A be the set of transitions in which a job moves from station i to station j. These transitions are then called A-transitions. Moreover, let $C \subseteq A$ be the event that a job that moves from station i to station j finds station j congested and is blocked. The Palm probability of the C-event given that an A-transition occurs, denoted by $P_A(C)$, is then as follows (see [14], p. 159):

$$P_A(C) = \frac{\sum_{(\mathbf{n},\mathbf{n}') \in C} \pi(\mathbf{n}) q(\mathbf{n}, \mathbf{n}')}{\sum_{(\mathbf{n},\mathbf{n}') \in A} \pi(\mathbf{n}) q(\mathbf{n}, \mathbf{n}')}, \quad C \subseteq A \tag{25}$$

The Palm probability $P_A(C)$ can be interpreted as the expected number of A-transitions at which a C-event occurs in a fixed time interval divided by the expected number of all A-transitions in the same interval (see [51], p. 300).

Again, for illustrative purposes, an example is provided.

Example 9 (Tandem queue with jump-over blocking: Blocking probability, 2/2)**.** Again, consider the tandem queue with jump-over blocking mechanism that is discussed in Examples 4, 6 and 8. In Example 8, it is already shown how the blocking probability of arriving jobs at station 1 can be determined using the PASTA property of Poisson arrivals. However, the PASTA property cannot be used to find the probability that a job that leaves station 1 finds all servers at station 2 occupied. Instead, this blocking probability, which is denoted by $B_{1,2,tq}$, can be determined by computation of a Palm probability. More specifically, using the product form (17), $B_{1,2,tq}$ can be determined as follows:

$$B_{1,2,tq}(\lambda, \mu_1, \mu_2, N_1, N_2) = \frac{\sum_{n_1=1}^{N_1} \pi_{tq}(n_1, N_2) n_1 \mu_1}{\sum_{n_1=1}^{N_1} \sum_{n_2=0}^{N_2} \pi_{tq}(n_1, n_2) n_1 \mu_1} \tag{26}$$

2.5.2 Simulation

An approach that is frequently taken to determine performance measures, among which blocking probabilities, is simulation. Simulation can be described as “the process of designing and creating a computerized model of a real or proposed system for the purpose of conducting numerical experiments to give us a better understanding of the behaviour of that system for a given set of conditions” ([35], p. 7). The main advantage of simulation is that it allows for the study of very complicated systems. Disadvantages are that the results are generally not exact and that a relatively long computation time is required. A particular type of simulation model is a discrete-event simulation model. In such a model, changes in the state of the system only occur at discrete points in time (see e.g. [35, 48]).

In this report, discrete-event simulation is used to analyze the overflow system of interest when the service times are not exponentially distributed. In Section 3.6.1, the simulation results are presented. Besides that, Appendix C provides more extensive information regarding the simulation model, the determination of the length of the warm-up period, run length and number of replications and the verification and validation.

2.5.3 Approximation methods

If it is not possible or undesirable to compute the steady-state distribution analytically or numerically (e.g. because of a long computation time), it may also be an option to apply an approximation method to approximate the blocking probabilities. Over the past decades, various methods to approximate blocking probabilities have therefore been developed. Below, two approximation methods are briefly mentioned.

First of all, a way to approximate blocking probabilities that can be thought of is by means of a product-form modification. The idea of this method is to modify the (unsolvable) system in order to obtain a product-form system, that is, a system that does exhibit product-form solution for the steady-state distribution. The blocking probabilities for the product-form system can then be determined as described in Section 2.5.1. These blocking probabilities, in turn, provide an approximation, or even a secure bound, of the blocking probabilities for the original system (see e.g. [27, 28, 60, 61]). Secondly, in the context of overflow systems, it is worthwhile to mention the Equivalent Random Method, which was originally developed by Wilkinson [62]. Since then, it has found several applications in literature (see e.g. [11, 39, 49, 52, 63]). In Section 2.7.2, these approximation methods are further discussed.

2.6 Insensitivity

When determining the joint steady-state distribution of the number of jobs in a queueing network, it is often most convenient to assume that the service times are exponentially distributed. For example, this could be desirable in order to represent the queueing network by a continuous-time Markov chain. However, in reality, it may well be that the exponential distribution does not provide a very accurate description of the service times. Therefore, it would be useful if the queueing network is insensitive to the service time distributions (see e.g. [27, 54, 58]). In that case, the steady-state distribution does not depend on the service time distributions other than through their means. This is thus an appealing property, since the steady-state distribution then remains valid, even if the service times are not exponentially distributed (provided that the means are the same).

Finally, the section ends with an example that further discusses the Erlang loss system, which is an insensitive single-station queueing system.

Example 10 (Erlang loss system: Insensitivity)**.** Again, consider the Erlang loss system that is studied in Examples 2, 3 and 7. This system is known to be insensitive to the service time distribution (see e.g. [1, 54]). This means that the steady-state distribution (11) is the same for other, non-exponential service time distributions, provided that the mean is equal to $1/\mu$. As a consequence, instead of $M|M|N|N$ queue, the system is also frequently referred to as $M|G|N|N$ queue, where the G in the second field denotes that the service time has a general distribution (instead of M for Markovian).

2.7 Overflow systems

This section contains a discussion on overflow systems. First of all, as an example, Section 2.7.1 considers an overflow system with parallel structure that is known from literature and related to the overflow system in Chapter 3. Secondly, in Section 2.7.2, some other literature concerning overflow systems is briefly discussed.

2.7.1 Example: An overflow system with parallel structure

This section considers the overflow system that is studied in [60], which can be seen as closely related to the overflow system that is subject of Chapter 3 (see Remark 8 below). The purpose of this section is twofold. First of all, it aims to illustrate how the theory that is discussed in this chapter can be applied to an overflow system. Secondly, it serves as an introduction to the overflow system with serial structure that is studied in Chapter 3.

This section has the following structure. First of all, a formal model description is given. Next, it is shown how the steady-state distribution and blocking probabilities can be determined. Finally, it is discussed under which conditions the insensitivity property holds.

Model description

Figure 4 depicts the system with parallel structure that is considered in this section. The overflow system consists of two stations: station 1 with a capacity of N_1 servers and station 2 with a capacity of N_2 servers. Type 1 jobs arrive at station 1 according to a Poisson process with rate λ_1. These arriving type 1 jobs are only accepted at station 1 if there is a server available. If all servers at

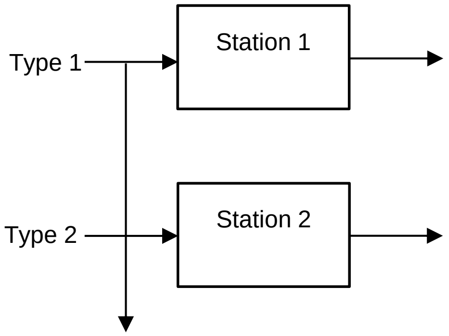


Figure 4: An overflow system with parallel structure

(primary) station 1 are occupied, the arriving type 1 jobs are overflowed to (secondary) station 2. Next to these arrivals, station 2 also receives direct arrivals of type 2 jobs, which arrive according to a Poisson process with rate λ_2.

Overflowed type 1 jobs and arriving type 2 jobs are then only admitted to station 2 if they can immediately be served, that is, if there is a server available. This may depend not only on the total number of type 1 and type 2 jobs that are present, but also on the number of each job type. More specifically, let the state of the system be denoted by $\mathbf{n} = (n_1, n_2, m)$, where:

- n_1: The number of type 1 jobs at station 1.
- n_2: The number of type 2 jobs at station 2.
- m: The number of overflowed type 1 jobs at station 2.

The allowed number of type 1 and type 2 jobs at station 2 is then restricted to a so-called coordinate convex set $\boldsymbol{C}$, which is given by:

$$\boldsymbol{C} = \{(n_2, m) \mid 0 \leq n_2 + m \leq N_2,\ 0 \leq n_2 \leq L,\ 0 \leq m \leq M\}, \quad \text{where } L, M \in \{0, 1, ..., N_2\} \tag{27}$$

It can be noted that this set possesses the following property, which is known as the coordinate convex property (see e.g. [58], p. 7):

$$(n_2, m) \in \boldsymbol{C} \Rightarrow \begin{cases} (n_2 - 1, m) \in \boldsymbol{C} & (n_2 > 0) \\ (n_2, m - 1) \in \boldsymbol{C} & (m > 0) \end{cases} \tag{28}$$

Overflowed type 1 jobs are therefore accepted at station 2 if $(n_2, m + 1) \in \boldsymbol{C}$, while arriving type 2 jobs are accepted if $(n_2 + 1, m) \in \boldsymbol{C}$. If this is not the case, the arriving jobs are rejected and lost. An example of a coordinate convex set is the situation that each of the N_2 servers at station 2 can be occupied by both type 1 and type 2 jobs, which leads to $\boldsymbol{C} = \{(n_2, m) \mid 0 \leq n_2 + m \leq N_2\}$ (then, $L = M = N_2$). See, for example, [58, 60] and Example 11 in Section 3.3 for more examples of coordinate convex structures.

If an arriving job is accepted at station 1 or station 2, it is served by one of the available servers. Here, the service times are assumed to be exponentially distributed with rates μ_1 for type 1 jobs at station 1, μ_2 for type 2 jobs at station 2 and γ for overflowed type 1 jobs at station 2.

Finally, it is noted that it may occur that a type 1 job at (primary) station 1 completes service, while overflowed type 1 jobs are present at station 2. In this case, two possibilities are considered:

- When overflowed type 1 jobs are served at (secondary) station 2, they also complete service at this station, even if a server at (primary) station 1 becomes available.
- An overflowed type 1 job at (secondary) station 2 is switched to (primary) station 1 as soon as a server at station 1 becomes available.

The latter assumption is also known under the terms (immediate) repacking or call packing (see e.g. [10, 11, 26, 60, 61]). Therefore, the system under this assumption is referred to as the (parallel) overflow system with call packing. The system under the former assumption is referred to as the (parallel) overflow system without call packing.

Remark 6 (Overflow systems in literature). As mentioned before, this section is concerned with the overflow system that is considered in [60]. It is noted that this system has also been studied in literature under slightly different assumptions. For example, in [27, 28], the overflow system without call packing is studied under the assumptions that all servers at station 2 can be used by type 1 and type 2 jobs (i.e. $\boldsymbol{C} = \{(n_2, m) \mid 0 \leq n_2 + m \leq N_2\})$ and that overflowed type 1 jobs and type 2 jobs at station 2 have the same service parameter (i.e. $\mu_2 = \gamma$). Besides that, [61] considers both the overflow system with call packing and the overflow system without call packing under the assumptions that $\boldsymbol{C} = \{(n_2, m) \mid 0 \leq n_2 + m \leq N_2\}$ and $\mu_1 = \gamma$. Other related overflow systems can be found in, among others, [21] and [26] (Example 2). In Section 2.7.2, these and some other references regarding overflow systems are briefly discussed.

Remark 7 (Call packing: Modification). In [60], the overflow system without call packing that is described above is of main interest. In this reference, call packing is then solely considered as a modification of the system in order to obtain a product-form system, which may lead to an approximation, or even a bound, of the blocking probabilities for the system without call packing (see also Section 2.5.3). In this report, in contrast, call packing is not regarded as a modification, but rather as a natural part of the system.

Remark 8 (Overflow system in Chapter 3). It is noted that the overflow system that is studied in Chapter 3 can be seen as an extension of the overflow system that is described above. More specifically, in the system in Chapter 3, jobs that finish service at (primary) station 1 may also go to (secondary) station 2 (see also Remark 11 in Section 3.3).

Steady-state distribution

For the parallel overflow system (with or without call packing) as described above, it can be of interest to determine the joint steady-state distribution of the number of jobs in the system. Below, it is discussed how these steady-state distributions can be determined. To this end, the state space for the system with call packing, denoted by $\boldsymbol{S_{cp}}$, and the state space for the system without call packing, denoted by $\boldsymbol{S_{ncp}}$, are first given.

In the system with call packing, overflowed type 1 jobs at station 1 switch to station 1 as soon as a server becomes available. This means that overflowed type 1 jobs can only be present at station 2 if station 1 is fully occupied. As a consequence, the state space $\boldsymbol{S_{cp}}$ is given by:

$$\boldsymbol{S_{cp}} = \{\mathbf{n} \mid 0 \leq n_1 < N_1,\ m = 0,\ (n_2, 0) \in \boldsymbol{C} \ \text{ or } \ n_1 = N_1,\ m \geq 0,\ (n_2, m) \in \boldsymbol{C}\} \tag{29}$$

In the system without call packing, in contrast, there can also be overflowed type 1 jobs at station 2 if there are servers at station 1 available. Therefore, the state space $\boldsymbol{S_{ncp}}$ is as follows:

$$\boldsymbol{S_{ncp}} = \{\mathbf{n} \mid 0 \leq n_1 \leq N_1,\ (n_2, m) \in \boldsymbol{C}\} \tag{30}$$

For the system with call packing, a product-form solution for the joint steady-state distribution of the number of jobs in the system, denoted by $\pi_{cp} = (\pi_{cp}(\mathbf{n}),\ \mathbf{n} \in \boldsymbol{S_{cp}})$, can then be obtained (see Section 2.4.1 for a brief discussion on product forms).

Theorem 1 (Parallel overflow system with call packing: Product form)**.** *The parallel overflow system with call packing that is described above has the following steady-state distribution:*

$$\pi_{cp}(\mathbf{n}) = \begin{cases} c_{cp} \frac{1}{n_1!} \left(\frac{\lambda_1}{\mu_1}\right)^{n_1} \frac{1}{n_2!} \left(\frac{\lambda_2}{\mu_2}\right)^{n_2} & m = 0 \\ c_{cp} \frac{1}{n_1!} \left(\frac{\lambda_1}{\mu_1}\right)^{n_1} \frac{1}{n_2!} \left(\frac{\lambda_2}{\mu_2}\right)^{n_2} \frac{(\lambda_1)^m}{\prod_{k=1}^{m} (N_1\mu_1 + k\gamma)} & m > 0 \end{cases}, \quad \mathbf{n} \in \boldsymbol{S_{cp}} \tag{31}$$

Here, c_{cp} is a normalizing constant and $\mathbf{n} = (n_1, n_2, m)$ the state description.

Proof. There are different ways to prove the product form (31):

- The global balance equations can be written down, after which it can be verified that the product form (31) satisfies these for all $\mathbf{n} \in \boldsymbol{S_{cp}}$ (see [60], p. 10).
- The product-form result in [26] can be used to prove the product form (31) (see [60], pp. 28-29).
- The overflow system could also be modelled as competing Markov chains. Subsequently, the product form (31) can be concluded from the product-form result in [13] (see [60], pp. 29-31, and Appendix A.3). □

For the system without call packing, in contrast, a product-form solution for the steady-state distribution cannot be expected (see e.g. [61], p. 4, or Example 12 in Section 3.4.2). Instead, the steady-state distribution, denoted by $\pi_{ncp} = (\pi_{ncp}(\mathbf{n}),\ \mathbf{n} \in \boldsymbol{S_{ncp}})$, could be determined by a numerical algorithm, such as the GTH algorithm (see also Section 2.4.2). Besides that, when the steady-state distribution is determined in order to compute related performance measures (e.g. blocking probabilities), it could also be an option to approximate the desired performance measures (see also Section 2.5.3).

Blocking probabilities

For the parallel overflow system (with or without call packing) that is shown in Figure 4, one might be interested in finding the probability that a type 1 or type 2 job is rejected and lost. As discussed in Section 2.5.1, these blocking probabilities can be determined from the steady-state distribution.

More specifically, because of the PASTA property of Poisson arrivals, the blocking probabilities can be obtained by summing the steady-state probabilities over the states that lead to blocking.

For the system with call packing, the probability that a type 1 job is rejected (i.e. find station 1 fully occupied and can also not be overflowed to station 2), denoted by $B_{1,cp}$, and the probability that a type 2 job is rejected (i.e. find all N_2 servers occupied or at least M type 2 jobs in service), denoted by $B_{2,cp}$, can then be determined as follows:

$$B_{1,cp} = \sum_{n_2=0}^{L} \pi_{cp}(N_1, n_2, \min\{N_2 - n_2, M\}) \tag{32}$$

$$B_{2,cp} = \sum_{n_1=0}^{N_1-1} \pi_{cp}(n_1, L, 0) + \sum_{m=0}^{M} \pi_{cp}(N_1, \min\{N_2 - m, L\}, m) \tag{33}$$

Similarly, for the system without call packing, the probability that a type 1 job is rejected, denoted by $B_{1,ncp}$, and the probability that a type 2 job is rejected, denoted by $B_{2,ncp}$, are given by the following expressions:

$$B_{1,ncp} = \sum_{n_2=0}^{L} \pi_{ncp}(N_1, n_2, \min\{N_2 - n_2, M\}) \tag{34}$$

$$B_{2,ncp} = \sum_{n_1=0}^{N_1} \sum_{m=0}^{M} \pi_{ncp}(n_1, \min\{N_2 - m, L\}, m) \tag{35}$$

It can then be interesting to compare the blocking probabilities for the system with call packing (i.e. $B_{1,cp}$ and $B_{2,cp}$) and the blocking probabilities for the system without call packing (i.e. $B_{1,ncp}$ and $B_{2,ncp}$). In particular, it can be studied whether the system with call packing or the system without call packing performs better in terms of blocking probabilities. In [60], this is done for the blocking probability of type 1 jobs (i.e. $B_{1,cp}$ and $B_{1,ncp}$). It is argued that $B_{1,ncp} \leq B_{1,cp}$ for all situations with $\mu_1 \leq \gamma$, while no ordering can be expected if $\mu_1 > \gamma$. This is also supported by numerical experiments that are performed.

(In)sensitivity

Up to now, it is assumed that the service times of jobs are exponentially distributed. The question then arises if the results also remain valid if the service times follow another, non-exponential distribution. However, it appears that the parallel overflow system is not insensitive to the service time distributions, which can be shown by simulation (see e.g. [60], pp. 17-19). This holds for the overflow system with call packing as well as the overflow system without call packing. It is noted, though, that the overflow system with call packing is insensitive if it is assumed that the mean service time of type 1 jobs at station 1 and overflowed type 1 jobs at station 2 are equal (i.e. $\mu_1 = \gamma$) and service is preemptively resumed after call packing (see [60], pp. 26-27).

2.7.2 Brief discussion of literature

Section 2.7.1 contains a description of the overflow system that is considered in [60], which is closely related to the overflow system that is studied in Chapter 3. In this section, some other literature regarding overflow systems is briefly discussed as well. See, for example, the papers that are mentioned in this section and references therein for a more extensive discussion.

First of all, several product-form results have already been established for overflow systems with call packing (see e.g. [10, 26, 61]). Here, it is noted that call packing is not seen as a natural part of the system in [61]. Instead, similar as in [60] (see Remark 7 in Section 2.7.1), call packing is considered as a modification in order to obtain a product-form system. In this way, an upper bound for the blocking probability is obtained. Next, [11, 63] consider an overflow system with multiple primary stations and one secondary (overflow) station. Overflowed jobs are then allowed to switch to their primary station if a server at this station becomes available. In these references, however, it is assumed that switching to primary station i occurs with a rate β_i, where $0 \leq \beta_i \leq \infty$. It is then mentioned that an exact expression for the steady-state distribution can be obtained if switching occurs immediately (i.e. $\beta_i = \infty$). On the other hand, for cases with no switching (i.e. $\beta_i = 0$) or switching with a finite rate (i.e. $0 < \beta_i < \infty$), the blocking probability is approximated using the Equivalent Random Method (see also Section 2.5.3).

Secondly, several analytical (product-form) results have also been obtained for overflow systems without call packing. For example, in [21], an overflow system that consists of a primary station and a secondary (overflow) station is considered. This system is similar to the overflow system without call packing that is described in Section 2.7.1, although there are some differences. For example, direct arrivals at the secondary station are not incorporated, and overflowed jobs are assumed to have the same service parameter. An explicit expression for the joint probabilities of the number of busy servers at the primary station and secondary station is then derived. From these probabilities, related performance measures, such as overflow probabilities, can then be determined.

However, especially in case of overflow systems without call packing, an explicit expression for the steady-state distribution cannot always be (easily) derived. Therefore, several methods to approximate performance measures for overflow systems, such as blocking probabilities, have also been developed over the last decades (see also Section 2.5.3). For example, in [27, 28, 60, 61], the blocking probabilities for an overflow system are approximated by means of a product-form modification. More specifically, the overflow system is modified in order to obtain a product-form system, that is, a system for which a product-form solution for the steady-state distribution can be obtained. The blocking probabilities for the product-form system can then be computed, and lead to an approximation, or even a bound, of the blocking probabilities for the original system. Another method to approximate the blocking probabilities, which has been frequently applied, is the Equivalent Random Method (see e.g. [11, 39, 49, 52, 62, 63]).

Finally, it is noted that most of the studies regarding overflow systems have in common that they focus on a system with a parallel structure, which means that jobs that finish service at the primary station cannot go the secondary station. This is in contrast with overflow systems with a serial structure, such as the system that is studied in Chapter 3.

Chapter 3

Overflow system with serial structure

In this chapter, a two-station overflow system with serial structure is studied. This means that jobs could also go to the secondary (overflow) station after they complete service at the primary station. This is in contrast with the overflow system with parallel structure that is discussed in Section 2.7.1, which assumes that jobs always leave the system after finishing service at the primary station.

3.1 Outline of chapter

The structure of this chapter is as follows:

- Section 3.2 introduces the notation that is used in this chapter.
- Section 3.3 describes the overflow system with serial structure. As in Section 2.7.1 for the parallel case, the system is considered both with and without the assumption of call packing.
- In Section 3.4, it is discussed how the joint steady-state distribution of the number of jobs in the system can be determined. For the overflow system with call packing, a product-form solution for the steady-state distribution is derived. For the overflow system without call packing, it is discussed how the steady-state distribution can be determined by using a numerical algorithm.
- In Section 3.5, it is discussed which blocking probabilities can be of interest and how these can be calculated. Moreover, for illustrative purposes and in order to study the effect of call packing on the blocking probabilities, numerical results are provided.
- In Section 3.6, the feature of insensitivity is studied. Simulation experiments are performed in order to study under which conditions on the parameters an insensitivity result might be established. Moreover, the overflow system with call packing is shown to be insensitive when non-overflowed and overflowed jobs have the same service parameter.
- In Section 3.7, the chapter concludes with a summary of the results.

Finally, some additional information is provided in the appendix. More specifically, some proof details are given in Appendices A.1 and A.2. Moreover, Appendix B contains additional information regarding the blocking probabilities. Finally, the simulation procedure is discussed in Appendix C, and Matlab code is provided in Appendix D.

3.2 Notation

In this section, the notation that is used in this chapter is introduced. It is noted that most of the notation is also mentioned in Section 3.3, which provides the model description. For ease of reading, a complete description of the notation is given in this section.

Beforehand, it is noted that a superscript is used to refer to the job type. Similarly, a subscript is used when a station is referred to. Here, it is possible that a subscript contains more than one character, such as μ_{ij}^t, $p_{i,k}^t$ and n_{ij}^t (see also Figure 5). For example, in order to indicate at which station a job is served, two characters are used. In this case, the first subscript, say i, refers to the primary station (i.e. the preferred station, at which the job is served if possible), while the second subscript, say j, refers to the actual station (i.e. the station at which the job actually receives service). It can thus be noted that $i = j$ for non-overflowed jobs, while $i \neq j$ if the jobs are overflowed.

The notation for the input parameters is then as follows:

Table 1: Notation: Input parameters

λ_1:	Arrival rate at station 1 (of type 1 jobs), $\lambda_1 \geq 0$.
λ_2:	Arrival rate at station 2 (of type 2 jobs), $\lambda_2 \geq 0$.
μ_{11}^1:	Service rate of type 1 jobs with primary station 1 that are present at station 1, $\mu_{11}^1 > 0$.
μ_{12}^1:	Service rate of type 1 jobs with primary station 1 that are present at station 2, $\mu_{12}^1 > 0$.
μ_{22}^1:	Service rate of type 1 jobs with primary station 2 that are present at station 2, $\mu_{22}^1 > 0$.
μ_{22}^2:	Service rate of type 2 jobs with primary station 2 that are present at station 2, $\mu_{22}^2 > 0$.
$p_{1,2}^1$:	Probability that a type 1 job with primary station 1 goes to station 2 after finishing service, $p_{1,2}^1 \in [0, 1]$.$^{(a)}$
$p_{1,0}^1$:	Probability that a type 1 job with primary station 1 leaves the system after finishing service, $p_{1,0}^1 = 1 - p_{1,2}^1$.$^{(a)}$
N_1:	Number of servers at station 1, $N_1 \in \mathbb{N}$.
N_2:	Number of servers at station 2, $N_2 \in \mathbb{N}$.
M_2^1:	Maximum number of type 1 jobs that can be present at station 2, $M_2^1 \in \{0, ..., N_2\}$.
M_2^2:	Maximum number of type 2 jobs that can be present at station 2, $M_2^2 \in \{0, ..., N_2\}$.
$\boldsymbol{C_2}$:	Coordinate convex set at station 2.

$^{(a)}$The routing probabilities for overflowed and non-overflowed jobs are assumed to be equal. Hence, they only depend on the primary station (and not on the actual station).

Next, the following notation is used to describe the underlying continuous-time Markov chain:

Table 2: Notation: Continuous-time Markov chain

n_{11}^1:	Number of (non-overflowed) type 1 jobs with primary station 1 that are present at station 1, $n_{11}^1 \in \mathbb{N}$.
n_{12}^1:	Number of (overflowed) type 1 jobs with primary station 1 that are present at station 2, $n_{12}^1 \in \mathbb{N}$.
n_{22}^1:	Number of (non-overflowed) type 1 jobs with primary station 2 that are present at station 2, $n_{22}^1 \in \mathbb{N}$.
n_{22}^2:	Number of (non-overflowed) type 2 jobs with primary station 2 that are present at station 2, $n_{22}^2 \in \mathbb{N}$.
$\mathbf{n}$:	State of the system, $\mathbf{n} = (n_{11}^1, n_{12}^1, n_{22}^1, n_{22}^2)$.
$\boldsymbol{S}$:	Set of admissible states.
$q(\mathbf{n}, \mathbf{n}')$:	Transition rate from state $\mathbf{n}$ to state $\mathbf{n}'$, where $\mathbf{n}, \mathbf{n}' \in \boldsymbol{S}$.
Q:	Infinitesimal generator matrix.
$\pi(\mathbf{n})$:	Steady-state probability to observe state $\mathbf{n}$, where $\mathbf{n} \in \boldsymbol{S}$.

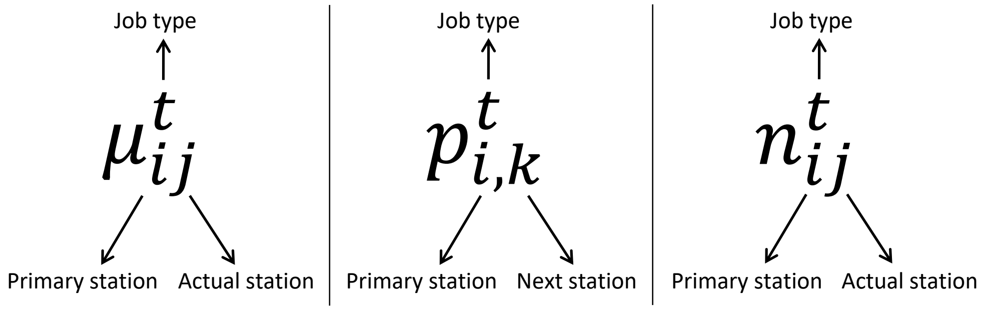


Figure 5: Illustration of the notation of μ^t_{ij}, $p^t_{i,k}$ and n^t_{ij} (see also Tables 1 and 2)

Subsequently, the following notation also comes in handy:

Table 3: Notation: Some additional notation

$\mathbf{n} - e^1_{11}$:	The same state with n^1_{11} decreased by one, that is, $\mathbf{n} - e^1_{11} = (n^1_{11} - 1, n^1_{12}, n^1_{22}, n^2_{22})$.
$\mathbf{n} - e^1_{12}$:	The same state with n^1_{12} decreased by one, that is, $\mathbf{n} - e^1_{12} = (n^1_{11}, n^1_{12} - 1, n^1_{22}, n^2_{22})$.
$\mathbf{n} - e^1_{22}$:	The same state with n^1_{22} decreased by one, that is, $\mathbf{n} - e^1_{22} = (n^1_{11}, n^1_{12}, n^1_{22} - 1, n^2_{22})$.
$\mathbf{n} - e^2_{22}$:	The same state with n^2_{22} decreased by one, that is, $\mathbf{n} - e^2_{22} = (n^1_{11}, n^1_{12}, n^1_{22}, n^2_{22} - 1)$.
$1_{\{C\}}$:	Indicator that is 1 if condition C is true and 0 otherwise.
$[E]^+$:	Expression that is 0 if $E \leq 0$ and E otherwise.

Then, as further discussed in Section 3.5.1, the following notation is used in order to denote the blocking probabilities:

Table 4: Notation: Blocking probabilities

b_1:	Probability that a job that arrives from outside at station 1 is blocked and lost or overflowed to station 2.
B_1:	Probability that a job that arrives from outside at station 1 is blocked and lost.
O_1:	Probability that a job that arrives from outside at station 1 is overflowed to station 2, $O_1 = b_1 - B_1$.
B_2:	Probability that a job that arrives from outside at station 2 is blocked and lost.
$B^1_{11,2}$:	Probability that a type 1 job with primary station 1 that completes service at station 1 and goes to station 2 is blocked and lost.
$B^1_{12,2}$:	Probability that a type 1 job with primary station 1 that completes service at station 2 and stays at station 2 is blocked and lost.
$B^1_{1,2}$:	Probability that a type 1 job with primary station 1 that completes service and goes to or stays at station 2 is blocked and lost.

Finally, it is noted that Section 3.6.2 introduces some additional notation in order to prove an insensitivity result. This notation is not discussed in this section.

3.3 Model description

In this section, the overflow system with serial structure is described. The notation that is used follows the notation that is introduced in Section 3.2 (see also Remark 9 at the end of this section).

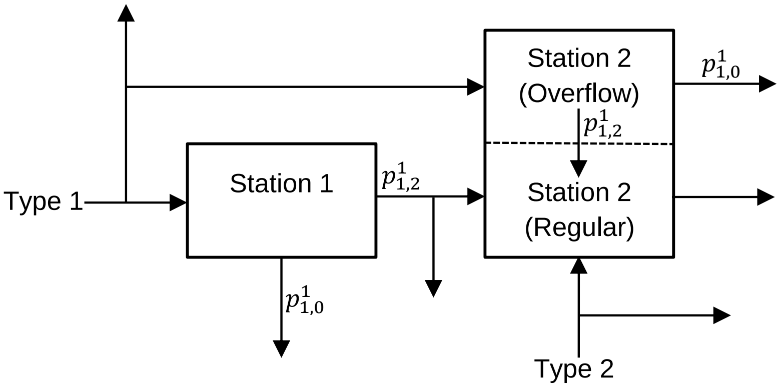


Figure 6: The serial overflow system that is studied in this chapter. Routing probabilities ($p^1_{1,k}$, $k = 0, 2$) are mentioned next to the applicable arrows

Figure 6 depicts the overflow system of interest. The system consists of two stations. Station 1 has a capacity of N_1 servers, while N_2 servers are present at station 2.

Arrivals at station 1 solely consist of arrivals of type 1 jobs that arrive from outside the system according to a Poisson process with rate λ_1. Arriving type 1 jobs are then served by one of the N_1 servers at station 1 if there is one available. Here, the service times are assumed to be exponentially distributed with rate μ^1_{11} (see also Remark 10). However, it may also occur that a type 1 job finds all N_1 servers at (primary) station 1 occupied upon arrival. In this case, arriving type 1 jobs are overflowed to (secondary) station 2.

Next to arrivals of these overflowed type 1 jobs, other arrival streams at station 2 can also be observed. First of all, type 2 jobs arrive from outside at station 2 according to a Poisson process with rate λ_2. Secondly, it is assumed that type 1 jobs require service at station 2 after completing service at station 1 with probability $p^1_{1,2} \in [0.1]$. This means that type 1 jobs that finish service at station 1 go to station 2 with probability $p^1_{1,2}$, while they leave the system with probability $p^1_{1,0}$, where $p^1_{1,0} = 1 - p^1_{1,2}$ (see also Remark 11). Similarly, it is also assumed that overflowed type 1 jobs that complete service at station 2 (instead of station 1) require a 'regular' service at station 2 with probability $p^1_{1,2}$ as well. This means that finished overflowed type 1 jobs stay at station 2 for a 'regular' service with probability $p^1_{1,2}$, while they leave the system with probability $p^1_{1,0}$.

The jobs at station 2 are therefore classified into three groups:

- Overflowed type 1 jobs that found station 1 fully occupied upon arrival.
- Non-overflowed type 1 jobs that require a 'regular' service at station 2. These jobs either come from station 1 or were already present at station 2, because they were overflowed.
- Type 2 jobs that arrive from outside the system.

If these jobs are accepted at station 2, they are served by one of the N_2 servers, where the service times are assumed to be exponentially distributed with rates μ^1_{12} for overflowed type 1 jobs, μ^1_{22} for non-overflowed type 1 jobs and μ^2_{22} for type 2 jobs (see also Remark 10). However, arriving jobs are only accepted at station 2 if they can immediately be served, that is, if there is a server available. Here, whether or not a server is available may depend not only on the total number of jobs that are present, but also on the number of present jobs of each job type.

In order to make this more formal, the notation for the state of the system is first introduced. To this end, let the state of the system be denoted by $\mathbf{n} = (n_{11}^1, n_{12}^1, n_{22}^1, n_{22}^2)$, where:

- n_{11}^1: The number of type 1 jobs at station 1.
- n_{12}^1: The number of overflowed type 1 jobs at station 2.
- n_{22}^1: The number of non-overflowed type 1 jobs at station 2.
- n_{22}^2: The number of type 2 jobs at station 2.

The allowed number of type 1 and type 2 jobs at station 2 is then restricted to a coordinate convex set $\boldsymbol{C_2}$, which is as follows:

$$\boldsymbol{C_2} = \{(n_{12}^1 + n_{22}^1, n_{22}^2) \mid 0 \leq n_{12}^1 + n_{22}^1 + n_{22}^2 \leq N_2,\ 0 \leq n_{12}^1 + n_{22}^1 \leq M_2^1,\ 0 \leq n_{22}^2 \leq M_2^2\}, \tag{36}$$

where $M_2^t \in \{0, 1, ..., N_2\}$, $t = 1, 2$. It can be noted that the following so-called coordinate convex property is then satisfied (see also Section 2.7.1):

$$(n_{12}^1 + n_{22}^1, n_{22}^2) \in \boldsymbol{C_2} \Rightarrow \begin{cases} (n_{12}^1 + n_{22}^1 - 1, n_{22}^2) \in \boldsymbol{C_2} & (n_{12}^1 + n_{22}^1 > 0) \\ (n_{12}^1 + n_{22}^1, n_{22}^2 - 1) \in \boldsymbol{C_2} & (n_{22}^2 > 0) \end{cases} \tag{37}$$

Overflowed type 1 jobs that found all servers at station 1 occupied are thus accepted at station 2 if $(n_{12}^1 + 1 + n_{22}^1, n_{22}^2) \in \boldsymbol{C_2}$. Similarly, type 1 jobs that complete service at station 1 and are routed to station 2 are accepted if $(n_{12}^1 + n_{22}^1 + 1, n_{22}^2) \in \boldsymbol{C_2}$. On the other hand, overflowed type 1 jobs that also require a 'regular' service at station 2 are always accepted, since they already occupy a server at station 2 (see also Remark 12). Finally, type 2 jobs that arrive from outside are accepted if $(n_{12}^1 + n_{22}^1, n_{22}^2 + 1) \in \boldsymbol{C_2}$. Otherwise, arriving jobs are rejected and leave the system.

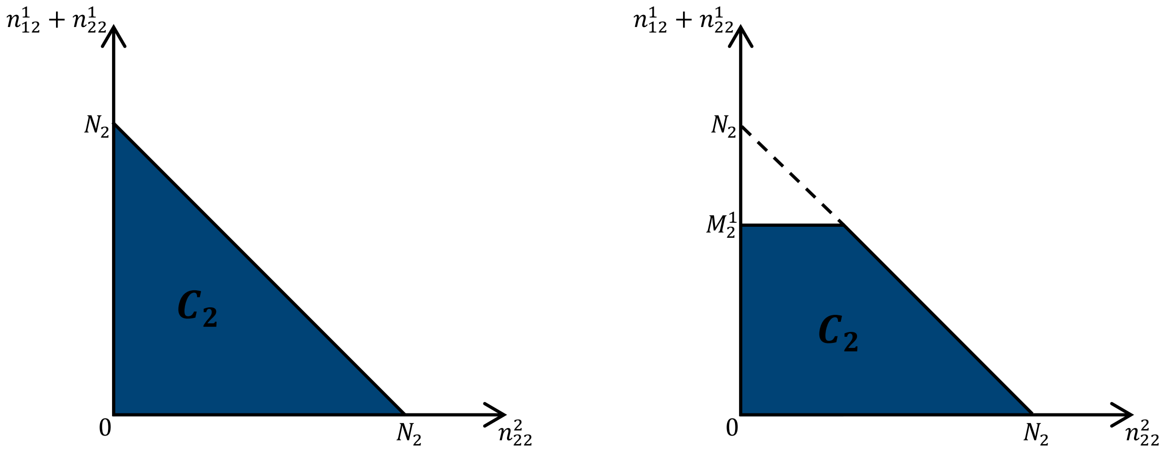
There are different possibilities for the choice of the coordinate convex structure $\boldsymbol{C_2}$, as illustrated by the following example.

Example 11 (Coordinate convex set $\boldsymbol{C_2}$)**.** Examples of the coordinate convex structure $\boldsymbol{C_2}$ include the following:

(i) $\boldsymbol{C_2} = \{(n_{12}^1 + n_{22}^1, n_{22}^2) \mid 0 \leq n_{12}^1 + n_{22}^1 + n_{22}^2 \leq N_2\}$. This corresponds to the natural case that there are N_2 servers at station 2 that can be occupied by both type 1 and type 2 jobs.

(ii) $\boldsymbol{C_2} = \{(n_{12}^1 + n_{22}^1, n_{22}^2) \mid 0 \leq n_{12}^1 + n_{22}^1 + n_{22}^2 \leq N_2,\ 0 \leq n_{12}^1 + n_{22}^1 \leq M_2^1\}$, where $M_2^1 \in \{0, 1, ..., N_2 - 1\}$. In this case, the total number of jobs present at station 2 is again at most N_2. Besides that, there is a maximum number of type 1 jobs that can be present at station 2, which is equal to M_2^1. As a consequence, $N_2 - M_2^1$ servers are kept exclusively available for type 2 jobs.

Examples (i) and (ii) are schematically depicted in Figures 7a and 7b, respectively.



(a) Example 11.(i)

(b) Example 11.(ii)

Figure 7: Two examples of the coordinate convex set $\boldsymbol{C_2}$ (see Example 11)

Subsequently, it is noted that two possibilities are considered when a type 1 job at station 1 finishes service, while one or more overflowed type 1 jobs are present at station 2:

- When overflowed type 1 jobs are served at (secondary) station 2, they also complete the service at this station, even if a server at (primary) station 1 becomes available.
- An overflowed type 1 job at (secondary) station 2 is switched to (primary) station 1 as soon as a place at station 1 becomes available (see also Remark 13).

The latter assumption is also known as (immediate) repacking or call packing (see also Section 2.7.1). For this reason, the system under this assumption is referred to as the serial overflow system with call packing. The system under the former assumption is referred to as the serial overflow system without call packing.

Finally, some remarks regarding the model description are made.

Remark 9 (Notation). Regarding the notation, it is noted that a subscript is used to refer to the station, while the job type is referred to with a superscript. The current notation is used, since it can easily be generalized, for example, when more overflow streams added, such as in Chapter 4. See Section 3.2 for a further discussion of the notation that is used to describe the overflow system.

Remark 10 (Non-exponential service times). At first, the service times are assumed to be exponentially distributed, since this enables us to describe the behaviour of the overflow system by a continuous-time Markov chain. The case of non-exponential service times is then considered in Section 3.6, which studies to what extent the results remain valid if the service times follow another, non-exponential distribution.

Remark 11 (Overflow system in Section 2.7.1). It is noted that the overflow system with parallel structure that is subject of Section 2.7.1 is a special version of the system that is described in this section. More specifically, by setting $p_{1,2}^1 = 0$, the overflow system with parallel structure is obtained. On the other hand, if $p^1_{1,2} > 0$, the overflow system has a serial structure in the sense that the service at (primary) station 1 might be followed by a service at (secondary) station 2.

Remark 12 (Blocking of non-overflowed type 1 jobs at station 2). As mentioned before, non-overflowed type 1 jobs at station 2 either come from station 1 or are already present at station 2 after being overflowed. In the latter case, the jobs already occupy a server at station 2, which means that they cannot be rejected (note that $n^1_{12} + n^1_{22}$ then remains unchanged). This holds for the system with call packing as well as the system without call packing. In the former case, a distinction is made between the system with call packing and the system without call packing.

- In the overflow system with call packing, type 1 jobs that go to station 2 after service completion at station 1 are always accepted if there are overflowed type 1 jobs present (i.e. $n^1_{12} > 0$). Because of the assumption of call packing, one of the overflowed type 1 jobs then goes to station 1, which means that there is always a server at station 2 available (note that $n^1_{12}+n^1_{22}$ then stays the same). On the other hand, if there are no overflowed type 1 jobs present (i.e. $n^1_{12} = 0$), type 1 jobs that come from station 1 may be rejected at station 2 (note that $n^1_{12}+n^1_{22}$ would then increase by one).
- In the overflow system without call packing, type 1 jobs that come from station 1 may be rejected at station 2 regardless of the presence of overflowed type 1 jobs (note that $n^1_{12} + n^1_{22}$ would then always be incremented by one).

If non-overflowed type 1 jobs are rejected at station 2, they are lost and leave the system. This can also be seen as if they skip or jump over station 2 when there is no server available. See Example 4 in Section 2.3 for a more extensive discussion of the jump-over blocking mechanism.

Remark 13 (Call packing and (non-)exponential service times). In the overflow system with call packing, an overflowed type 1 job, if present, switches from station 2 to station 1 as soon as a server at station 1 becomes available. It is noted, though, that it is not specified which overflowed type 1 job switches to station 1 if there is more than one present. For example, this could be the job that has been in service for the longest time (first in, first out, FIFO) or the shortest time (last in, first out, LIFO) or an arbitrary job (random). Moreover, it is also not specified whether the service of the job that switches to station 1 is preemptively resumed (resume) or completely restarted (resample). However, because of the memoryless property of the exponential distribution, the joint steady-state distribution of the number of present jobs is not affected by these choices. It can be noted that this is not the case when the service times are non-exponential (see also Section 3.6).

3.4 Steady-state distribution

3.4.1 Overflow system with call packing

For the serial overflow system with call packing, a product-form solution for the joint steady-state distribution of the number of jobs in the system can be derived. To this end, it is first noted that under the assumption of call packing there can only be overflowed jobs present ($n^1_{12} > 0$) if station 1 is fully occupied ($n_{11}^1 = N_1$). As a consequence, the state space $\boldsymbol{S}$ is as follows:

$$\boldsymbol{S} = \{\mathbf{n} \mid 0 \le n_{11}^1 < N_1,\ n_{12}^1 = 0,\ (n_{22}^1, n_{22}^2) \in \boldsymbol{C_2} \ \ \text{or} \ \ n_{11}^1 = N_1,\ (n_{12}^1 + n_{22}^1, n_{22}^2) \in \boldsymbol{C_2}\} \tag{38}$$

Next, the product-form solution for the steady-state distribution can be given.

Theorem 2 (Serial overflow system with call packing: Product form)**.** *The serial overflow system with call packing has the following steady-state distribution* $\pi = (\pi(\mathbf{n}), \mathbf{n} \in \boldsymbol{S})$*:*

$$\pi(\mathbf{n}) = cF(n_{12}^1)\frac{1}{n_{11}^1!}\left(\frac{\lambda_1}{\mu_{11}^1}\right)^{n_{11}^1} \frac{1}{n_{22}^1!}\left(\frac{p_{1,2}^1\lambda_1}{\mu_{22}^1}\right)^{n_{22}^1} \frac{1}{n_{22}^2!}\left(\frac{\lambda_2}{\mu_{22}^2}\right)^{n_{22}^2},\ \ \mathbf{n} = (n_{11}^1, n_{12}^1, n_{22}^1, n_{22}^2) \in \boldsymbol{S} \tag{39}$$

Here, c is a normalizing constant, $\boldsymbol{S}$ *the state space as defined in* (38) *and F a function, which is defined as follows:*

$$F(n) = \begin{cases} (\lambda_1)^n / \prod_{k=1}^{n}(N_1\mu_{11}^1 + k\mu_{12}^1) & n > 0 \\ 1 & n = 0 \end{cases} \tag{40}$$

Proof. In order to prove the result, it must be verified that the product form (39) satisfies the global balance equations for each $\mathbf{n} \in \boldsymbol{S}$. The global balance equations, in turn, are given by (for $\mathbf{n} \in \boldsymbol{S}$):

$$\left\{\begin{array}{l}
\pi(\mathbf{n})n_{11}^1\mu_{11}^1 1_{\{n_{11}^1>0\}}1_{\{n_{12}^1=0\}}+ \\
\pi(\mathbf{n})(N_1\mu_{11}^1 + n_{12}^1\mu_{12}^1)1_{\{n_{11}^1=N_1\}}1_{\{n_{12}^1>0\}}+ \\
\pi(\mathbf{n})n_{22}^1\mu_{22}^1 1_{\{n_{22}^1>0\}}+ \\
\pi(\mathbf{n})n_{22}^2\mu_{22}^2 1_{\{n_{22}^2>0\}}+ \\
\pi(\mathbf{n})\lambda_1 1_{\{n_{11}^1<N_1\}}+ \\
\pi(\mathbf{n})\lambda_1 1_{\{n_{11}^1=N_1\}}1_{\{(n_{12}^1+n_{22}^1+1,n_{22}^2)\in\boldsymbol{C_2}\}}+ \\
\pi(\mathbf{n})\lambda_2 1_{\{(n_{12}^1+n_{22}^1,n_{22}^2+1)\in\boldsymbol{C_2}\}}
\end{array}\right\}
\qquad
\begin{array}{r}
(41.1)\\ (41.2)\\ (41.3)\\ (41.4)\\ (41.5)\\ (41.6)\\ (41.7)
\end{array}$$

$$= \tag{41}$$

$$\left\{\begin{array}{l}
\pi(\mathbf{n}-e_{11}^1)\lambda_1 1_{\{n_{11}^1>0\}}1_{\{n_{12}^1=0\}}+ \\
\pi(\mathbf{n}-e_{12}^1)\lambda_1 1_{\{n_{11}^1=N_1\}}1_{\{n_{12}^1>0\}}+ \\
\pi(\mathbf{n}+e_{11}^1-e_{22}^1)p_{1,2}^1(n_{11}^1+1)\mu_{11}^1 1_{\{n_{11}^1<N_1\}}1_{\{n_{22}^1>0\}}+ \\
\pi(\mathbf{n}+e_{12}^1-e_{22}^1)p_{1,2}^1(N_1\mu_{11}^1+(n_{12}^1+1)\mu_{12}^1)1_{\{n_{11}^1=N_1\}}1_{\{n_{22}^1>0\}}+ \\
\pi(\mathbf{n}-e_{22}^2)\lambda_2 1_{\{n_{22}^2>0\}}+ \\
\pi(\mathbf{n}+e_{11}^1)p_{1,0}^1(n_{11}^1+1)\mu_{11}^1 1_{\{n_{11}^1<N_1\}}+ \\
\pi(\mathbf{n}+e_{12}^1)p_{1,0}^1(N_1\mu_{11}^1+(n_{12}^1+1)\mu_{12}^1)1_{\{n_{11}^1=N_1\}}1_{\{(n_{12}^1+n_{22}^1+1,n_{22}^2)\in\boldsymbol{C_2}\}}+ \\
\pi(\mathbf{n}+e_{11}^1)p_{1,2}^1(n_{11}^1+1)\mu_{11}^1 1_{\{n_{11}^1<N_1\}}1_{\{(n_{12}^1+n_{22}^1+1,n_{22}^2)\notin\boldsymbol{C_2}\}}+ \\
\pi(\mathbf{n}+e_{22}^1)(n_{22}^1+1)\mu_{22}^1 1_{\{(n_{12}^1+n_{22}^1+1,n_{22}^2)\in\boldsymbol{C_2}\}}+ \\
\pi(\mathbf{n}+e_{22}^2)(n_{22}^2+1)\mu_{22}^2 1_{\{(n_{12}^1+n_{22}^1,n_{22}^2+1)\in\boldsymbol{C_2}\}}
\end{array}\right\}
\qquad
\begin{array}{r}
(41.8)\\ (41.9)\\ (41.10)\\ (41.11)\\ (41.12)\\ (41.13)\\ (41.14)\\ (41.15)\\ (41.16)\\ (41.17)
\end{array}$$

Here, $1_{\{C\}}$ is an indicator that is equal to one if condition C is true and equal to zero otherwise. Moreover, the vectors e_{ij}^t contain a one at the specified index and zeros in the other fields (e.g. $e_{22}^1 = (0,0,1,0)$). See also Table 3 in Section 3.2.

Table 5: Verification of the global balance equations in (41)

Job class	Class balance
Type 1 jobs at station 1	(41.1) = (41.8)
Overflowed type 1 jobs at station 2	(41.2) = (41.9)
Non-overflowed type 1 jobs at station 2	(41.3) = (41.10) + (41.11)
Type 2 jobs at station 2	(41.4) = (41.12)
Type 1 jobs at the outside	(41.5) + (41.6) = (41.13) + (41.14) + (41.15) + (41.16)
Type 2 jobs at the outside	(41.7) = (41.17)

It is also noted that both the left-hand side and right-hand side of the global balance equations (41) are divided into several parts, which each have an interpretation of either a flow out of state $\mathbf{n}$ (left-hand side) or a flow into state $\mathbf{n}$ (right-hand side). For example, (41.15) represents the flow into state $\mathbf{n}$ due to a type 1 job that completes service at station 1 and jumps over station 2 to the outside because no server at station 2 is available.

Now, the proof can be completed by verifying that the global balance equations (41) are satisfied when the product form (39) is substituted. This can be done by verifying the class balances that are mentioned in Table 5. As discussed in Section 2.4.1, class balance can be read as that for all $\mathbf{n} \in \boldsymbol{S}$ the rate out of state $\mathbf{n}$ due to a departure of a job that belongs to some class (e.g. overflowed type 1 jobs at station 2) is equal to the rate into state $\mathbf{n}$ due to an arrival of a job of the same class. In Appendix A.1, it is described how the class balance equations in Table 5 can be verified for all $\mathbf{n} \in \boldsymbol{S}$. Since the product form (39) satisfies the class balances in Table 5, it follows that the global balance equations (41) are also satisfied for all $\mathbf{n} \in \boldsymbol{S}$. This completes the proof. □

Remark 14 (Computation normalizing constant). As mentioned above, the product form (39) contains a normalizing constant, which is such that the steady-state probabilities sum to one (see also Section 2.4.1). Hence, the normalizing constant, denoted by c, is given by the following expression:

$$c = \left(\sum_{\mathbf{n} \in \boldsymbol{S}} F(n^1_{12}) \frac{1}{n^1_{11}!} \left(\frac{\lambda_1}{\mu^1_{11}} \right)^{n^1_{11}} \frac{1}{n^1_{22}!} \left(\frac{p^1_{1,2}\lambda_1}{\mu^1_{22}} \right)^{n^1_{22}} \frac{1}{n^2_{22}!} \left(\frac{\lambda_2}{\mu^2_{22}} \right)^{n^2_{22}} \right)^{-1} \tag{42}$$

For situations with a relatively small number of servers at station 1 and station 2 (i.e. N_1 and N_2), which are considered in this report, the normalizing constant can be computed by evaluating the expression in (42). However, the evaluation of (42) may take a lot of computation time if N_1 and/or N_2, and consequently the number of states in the state space $\boldsymbol{S}$, become very large. In such cases, it may be necessary to search for an alternative method to obtain (an approximation of) the normalizing constant (see e.g. [47] for a further discussion).

3.4.2 Overflow system without call packing

In this section, the serial overflow system without call packing is considered. This means that overflowed type 1 jobs at station 2 do not switch to station 1, even if a server at station 1 becomes

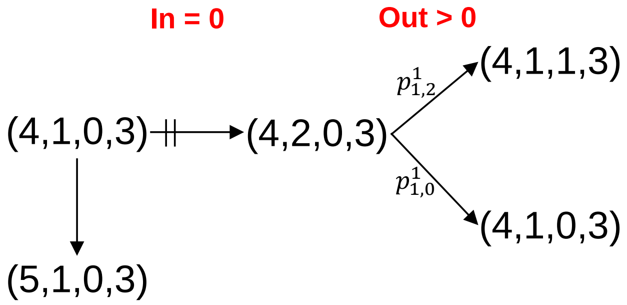


Figure 8: Example that illustrates why no product form can be expected

available. Therefore, the set of admissible states $\boldsymbol{S}$ is given by:

$$\boldsymbol{S} = \{\mathbf{n} \mid 0 \leq n^1_{11} \leq N_1,\ (n^1_{12} + n^1_{22}, n^2_{22}) \in \boldsymbol{C_2}\} \tag{43}$$

Next, it would be of interest if a product-form solution for the joint steady-state distribution of the number of jobs in the system could be derived. Unfortunately, however, a product form cannot be expected when overflowed type 1 jobs do not switch to station 1 when a server becomes available. This is illustrated by the following example.

Example 12 (No product-form solution)**.** Consider the serial overflow system without call packing, where it is assumed that station 1 and station 2 both have a capacity of five servers (i.e. $N_1 = N_2 = 5$). Moreover, let the coordinate convex set $\boldsymbol{C_2}$ be as in Example 11.(i) in Section 3.3. Then, consider the state $\mathbf{n} = (4, 2, 0, 3)$ (note that $\mathbf{n} \in \boldsymbol{S}$), that is, there are four type 1 jobs at station 1, two overflowed type 1 jobs at station 2, no non-overflowed type 1 jobs at station 2 and three type 2 jobs at station 2 present. It then follows that class balance for overflowed type 1 jobs at station 2 is violated. More specifically, as schematically depicted in Figure 8:

> The rate out of state $(4, 2, 0, 3)$ due to a departure of an overflowed type 1 job is positive. This would lead to state $(4, 1, 1, 3)$ (with probability $p^1_{1,2}$) or state $(4, 1, 0, 3)$ (with probability $p^1_{1,0}$).
>
> The rate into state $(4, 2, 0, 3)$ due to an arrival of an overflowed type 1 job is zero. This should then have occurred because of an arrival in state $(4, 1, 0, 3)$, However, an arrival of a type 1 job would lead to state $(5, 1, 0, 3)$, since the arriving type 1 job would go to station 1.

As a consequence, a product-form solution for the steady-state distribution of the number of jobs in the serial overflow system without call packing cannot be expected (cf. [61]). It is noted, though, that class balance is not a necessary condition for a product-form solution, which means that this does not prove that a product form is not available.

Instead, a numerical algorithm could be used to find the steady-state distribution. In this report, the GTH algorithm and Gauss-Seidel method are used for this purpose (see Section 2.4.2). In order to apply these algorithms, the infinitesimal generator matrix Q of the underlying continuous-time Markov chain is required. To this end, the transition rates $q = \{q(\mathbf{n}, \mathbf{n}'), \mathbf{n}, \mathbf{n}' \in \boldsymbol{S}\}$ are first given:

$$q(\mathbf{n}, \mathbf{n}') = \begin{cases} \lambda_1 & \mathbf{n}' = \mathbf{n} + e^1_{11} \\ p^1_{1,2} n^1_{11} \mu^1_{11} & \mathbf{n}' = \mathbf{n} - e^1_{11} + e^1_{22} \\ p^1_{1,0} n^1_{11} \mu^1_{11} 1_{\{(n^1_{12}+n^1_{22}+1, n^2_{22}) \in \boldsymbol{C_2}\}} & \mathbf{n}' = \mathbf{n} - e^1_{11} \\ n^1_{11} \mu^1_{11} 1_{\{(n^1_{12}+n^1_{22}+1, n^2_{22}) \notin \boldsymbol{C_2}\}} & \mathbf{n}' = \mathbf{n} - e^1_{11} \\ \lambda_1 1_{\{n^1_{11} = N_1\}} & \mathbf{n}' = \mathbf{n} + e^1_{12} \\ p^1_{1,2} n^1_{12} \mu^1_{12} & \mathbf{n}' = \mathbf{n} - e^1_{12} + e^1_{22} \ , \\ p^1_{1,0} n^1_{12} \mu^1_{12} & \mathbf{n}' = \mathbf{n} - e^1_{12} \\ n^1_{22} \mu^1_{22} & \mathbf{n}' = \mathbf{n} - e^1_{22} \\ \lambda_2 & \mathbf{n}' = \mathbf{n} + e^2_{22} \\ n^2_{22} \mu^2_{22} & \mathbf{n}' = \mathbf{n} - e^2_{22} \\ 0 & \text{else} \end{cases} \qquad \mathbf{n} \neq \mathbf{n}' \tag{44}$$

$$q(\mathbf{n}, \mathbf{n}) = - \sum_{\mathbf{n}' \in \boldsymbol{S} \setminus \mathbf{n}} q(\mathbf{n}, \mathbf{n}') \tag{45}$$

Let s then denote the number of states in the state space $\boldsymbol{S}$. Moreover, let each state $\mathbf{n} \in \boldsymbol{S}$ correspond to a unique number $i \in \{1, ..., s\}$ and denote the state corresponding to i by $\mathbf{n}(i)$. The infinitesimal generator matrix Q is then an $s \times s$ matrix with entries q_{ij}, where:

$$q_{ij} = q(\mathbf{n}(i), \mathbf{n}(j)), \qquad i, j \in \{1, ..., s\} \tag{46}$$

Now that the infinitesimal generator matrix Q is defined, the GTH algorithm (see Algorithm 1 in Section 2.4.2) or Gauss-Seidel method (see Algorithm 2 in Section 2.4.2) can be used to determine the steady-state distribution. Here, it is noted that, next to the infinitesimal generator matrix Q, an initial approximation of the steady-state vector π_0, a maximum number of iterations *itmax* and a tolerance for the stopping test *tol* should also be provided if the Gauss-Seidel method is applied (see Section 2.4.2). Unless specified otherwise, π_0 is a $1 \times s$ vector with entries $1/s$, and *itmax* and *tol* are set equal to 1000 and 10^{-8}, respectively.

It then appears that both the GTH algorithm and Gauss-Seidel method lead to a satisfactory runtime and accuracy if the number of admissible states is relatively small (i.e. a few hundred). However, if the state space gets larger, the Gauss-Seidel method vastly outperforms the GTH algorithm in terms of runtime. This is illustrated by the following example.

Example 13 (Comparison of GTH algorithm and Gauss-Seidel method)**.** Consider the serial overflow system without call packing, and let the parameter values be as given in Table 6. It can thus be noted that the number of servers at station 1 N_1 is varied, so that the number

Table 6: Comparison of GTH algorithm and Gauss-Seidel method: Parameter values

λ_1	λ_2	μ_{11}^1	μ_{12}^1	μ_{22}^1	μ_{22}^2	$p_{1,2}^1$	N_1	N_2	M_2^1	M_2^2
20	10	4	7	5	6	1	-	8	8	8

Table 7: Comparison of GTH algorithm and Gauss-Seidel method: Probability that station 2 is congested

Algorithm	$N_1 = 1$	$N_1 = 2$	$N_1 = 3$	$N_1 = 4$	$N_1 = 5$	$N_1 = 6$	$N_1 = 7$	$N_1 = 8$
GTH algorithm	0.2394	0.2154	0.1927	0.1718	0.1533	0.1377	0.1254	0.1165
Gauss-Seidel	0.2394	0.2154	0.1927	0.1718	0.1533	0.1377	0.1254	0.1165

See Table 6 for the parameter values.

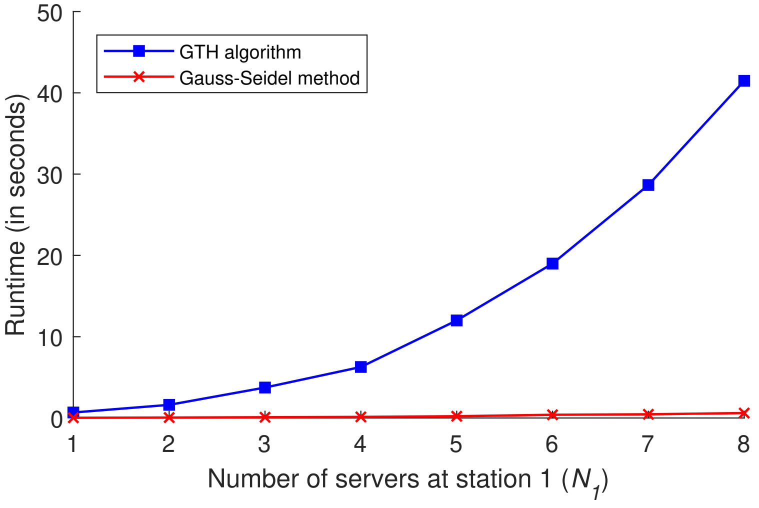


Figure 9: Comparison of GTH algorithm and Gauss-Seidel method: Runtime

of admissible states ranges from 330 ($N_1 = 1$) to 1485 ($N_1 = 8$). The GTH algorithm as well as the Gauss-Seidel method are then used to find the steady-state distribution.

Both algorithms indeed lead to very similar steady-state probabilities. This is illustrated by the results in Table 7. This table contains the probability that station 2 is fully occupied, which is computed from the resulting steady-state probabilities. However, the Gauss-Seidel method turns out to be considerably faster than the GTH algorithm. This can be seen from Figure 9, in which the time that it takes to compute the steady-state distribution from the infinitesimal generator matrix Q is shown.

Hence, for the serial overflow system without call packing that is considered in this chapter, the Gauss-Seidel method is preferred over the GTH algorithm when determining the steady-state distribution. It is also concluded that the Gauss-Seidel method as implemented suffices for the application in this report, since it leads to a satisfactory runtime and accuracy for the instances that are considered. It is noted, though, that it might be desirable to search for a more efficient implementation of a numerical algorithm if the number of admissible states gets much larger.

Besides that, next to using a numerical algorithm, other methods to obtain the steady-state distribution and/or (an approximation of) related performance measures can then also be thought of. For example, as discussed in Section 2.5.2, discrete-event simulation could be used to determine performance measures, such as the blocking probability. Moreover, it can be studied to what extent approximation methods, such as those mentioned in Section 2.5.3, could be of help to approximate performance measures, and in particular the blocking probabilities. This remains an interesting point for future research.

3.5 Blocking probabilities

3.5.1 Blocking probabilities of interest

In order to analyze the serial overflow system, several performance measures can be obtained, such as blocking (or loss or rejection) probabilities, the mean number of busy servers and the throughput. In this report, the main focus lies on the blocking probabilities. Three different 'types' of blocking probabilities are distinguished:

- The probability that an arriving job finds all servers at the primary station occupied and either leaves the system or is served at an overflow station. This probability is denoted by b.
- The probability that an arriving job finds all servers at the primary station occupied, can also not be served at an overflow station and leaves the system. This probability is denoted by B.
- The probability that an arriving job finds all servers at the primary station occupied, but can be served at an overflow station. This probability is denoted by O.

Moreover, these probabilities can be determined for different job types and stations. Therefore, subscripts and/or superscripts are also included when the blocking probabilities are referred to (see Table 4 in Section 3.2 for explanation of the notation). For the serial overflow system, the following blocking probabilities are then of interest (see also Figure 10):

$$b_1 = P[\text{A type 1 job arriving from outside at station 1 is lost or overflowed to station 2}] \tag{47}$$

$$B_1 = P[\text{A type 1 job that arrives from outside at station 1 is rejected and lost}] \tag{48}$$

$$O_1 = P[\text{A type 1 job arriving from outside at station 1 is overflowed to station 2}] \tag{49}$$

$$B_2 = P[\text{A type 2 job that arrives from outside at station 2 is rejected and lost}] \tag{50}$$

$$B^1_{11,2} = P[\text{A type 1 job that goes from station 1 to station 2 is blocked}] \tag{51}$$

$$B^1_{12,2} = P[\text{An overflowed type 1 job that stays at station 2 after finishing service is blocked}] \tag{52}$$

$$B^1_{1,2} = P[\text{A non-overflowed type 1 job at station 2 (coming from station 1 or 2) is blocked}] \tag{53}$$

Here, it is noted that $B^1_{12,2}$ in (52) is equal to zero, since an overflowed type 1 job already occupies a server at station 2. As a consequence, these jobs cannot be rejected when they stay at station 2 for a 'regular' service (see also Remark 12 in Section 3.3).

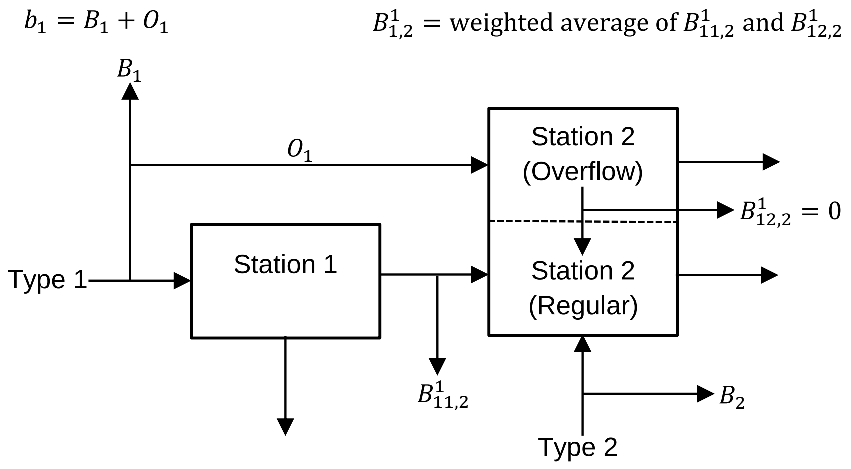


Figure 10: The blocking probabilities of interest for the serial overflow system. The routing probabilities that are mentioned in Figure 6 are omitted

3.5.2 Calculation of the blocking probabilities

In Section 3.4.1, the product-form solution for the joint steady-state distribution of the number of jobs in the overflow system with call packing is derived. Moreover, Section 3.4.2 describes how the steady-state distribution of the number of jobs in the overflow system without call packing can be found by applying the Gauss-Seidel method or GTH algorithm. This section briefly describes how the blocking probabilities of interest, which are mentioned in Section 3.5.1, can be determined from these steady-state distributions. In Appendix B.1, a more detailed description of the calculation of the blocking probabilities is given.

First of all, if the arrivals originate from outside the system, the PASTA property of Poisson arrivals can be used to determine the blocking probabilities (see Section 2.5.1). Because of this property, the blocking probabilities b_1, B_1, O_1 and B_2 can be computed by summing $\pi(\mathbf{n})$ over the appropriate states $\mathbf{n} = (n^1_{11}, n^1_{12}, n^1_{22}, n^2_{22})$. For example, the probability that a type 1 job that arrives from outside at station 1 is blocked and leaves the system (denoted by B_1) can be calculated as follows:

$$B_1 = \sum_{n^1_{12}=0}^{M^1_2} \sum_{n^2_{22}=0}^{\min\{M^2_2, N_2 - n^1_{12}\}} \pi(N_1, n^1_{12}, \min\{M^1_2 - n^1_{12}, N_2 - n^1_{12} - n^2_{22}\}, n^2_{22}) \tag{54}$$

Besides the steady-state distribution π, this expression is the same for both the system with call packing and the system without call packing. It is noted that this does not hold true for all blocking probabilities, as is illustrated in Appendix B.1.

Secondly, if the arrivals originate from one of the stations, the PASTA property can no longer be used. Instead, the blocking probabilities $B^1_{11,2}$ and $B^1_{1,2}$ can be determined with the use of Palm probabilities (see Section 2.5.1). This is explained more detailed in Appendix B.1.

3.5.3 Numerical results

In this section, some numerical results of the blocking probabilities are given for illustrative purposes. This also makes it possible to provide some insight into how the blocking probabilities for the system with call packing compare to those for the system without call packing. Below, the results of three numerical experiments are then discussed. Moreover, Appendix B.2 contains some additional numerical results.

Numerical experiment 1 (B_1 for different values of μ_{12}^1)

Table 8 contains the parameter values for numerical experiment 1. This experiment considers the probability that a type 1 job that arrives at station 1 is rejected and lost (i.e. B_1) for varying service rates of overflowed type 1 jobs at station 2 (i.e. μ_{12}^1). The results are then shown in Figure 11. It can be observed that the blocking probability B_1 decreases when μ_{12}^1 increases, and consequently the mean service time of overflowed type 1 jobs at station 2 (i.e. $1/\mu_{12}^1$) decreases. Moreover, it can be seen that B_1 for the system with call packing is smaller than B_1 for the system without call packing if $\mu_{12}^1 \leq 2$, while this order is reversed if $\mu_{12}^1 \geq 3$. In order to explain this, it is noted that, in line with overflow systems with a parallel structure (see e.g. [60, 61]), two effects of call packing on the blocking probability B_1 can be thought of:

(i) In the system with call packing, an overflowed type 1 job, if present, switches from a server at station 2 to a server at station 1 as soon as one becomes available. Hence, instead of the server at station 1, the server at station 2 then becomes available, but this server can also be taken by a non-overflowed type 1 job that comes from station 1 or a type 2 job that arrives from outside the system. In the system without call packing, in contrast, overflowed type 1 jobs may occupy a server at station 2, even if servers at station 1 are available. This means that these servers at station 1 are kept exclusively available for arriving type 1 jobs at station 1.

(ii) If $\mu_{11}^1 \neq \mu_{12}^1$, call packing also has another effect on the blocking probabilities. This is due to the change of service speed when an overflowed type 1 jobs switches from station 2 to station 1. More specifically, if $\mu_{11}^1 > \mu_{12}^1$, overflowed jobs are served by a faster server at station 1 if they switch from station 2 to station 1. Therefore, on average, these jobs take a shorter time to leave the system than when they would not switch. As a consequence, the availability of both station 1 and station 2 may be enhanced by call packing if $\mu_{11}^1 > \mu_{12}^1$. Similarly, if $\mu_{11}^1 < \mu_{12}^1$, overflowed jobs are served by a slower server if they switch from station 2 to station 1. Hence, if $\mu_{11}^1 < \mu_{12}^1$, both station 1 and station 2 may be less often available when call packing is assumed.

Hence, apparently, if $\mu_{12}^1 \leq 2$, the positive impact of effect (ii) on B_1 outweighs the negative impact of effect (i) on B_1, which results in a blocking probability B_1 that is larger for the system with call packing than for the system without call packing. On the other hand, if $3 \leq \mu_{12}^1 < \mu_{11}^1 = 10$, the positive impact of effect (ii) on B_1 is not sufficient to overcome the negative impact of effect (i) on B_1. As a consequence, the blocking probability B_1 for the system with call packing is no longer smaller than that for the system without call packing. Finally, if $\mu_{12}^1 \geq \mu_{11}^1 = 10$, overflowed type 1

Table 8: Experiment 1

Parameter	Value
λ_1	50
λ_2	15
μ_{11}^1	10
μ_{12}^1	-
μ_{22}^1	20
μ_{22}^2	16
$p_{1,2}^1$	1
N_1	5
N_2	5
M_2^1	5
M_2^2	1

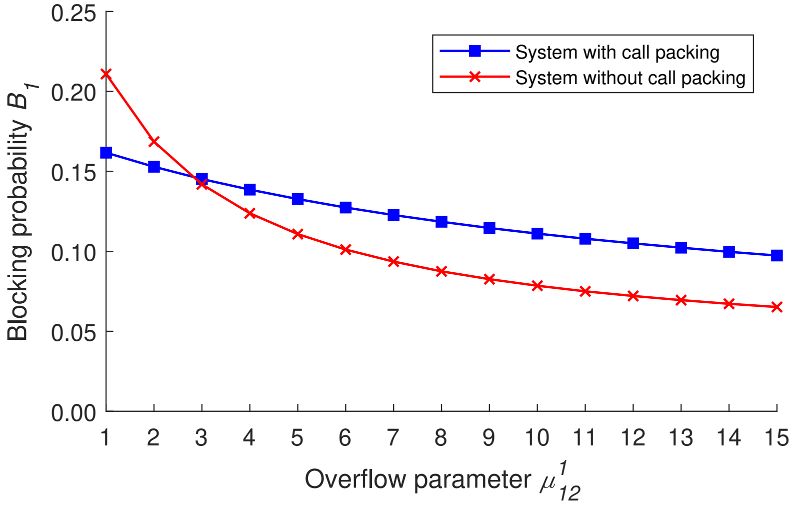


Figure 11: Results of experiment 1 (B_1 for different values of μ_{12}^1)

jobs are served by an equally fast or slower server if they switch from station 2 to station 1. This means that both effect (i) and effect (ii) have a negative (or no) impact on B_1. In this case, B_1 for the system with call packing is therefore expected to be larger than B_1 for the system without call packing, which indeed appears to be the case.

Numerical experiment 2 (B_1 for different values of $p_{1,2}^1$)

The parameter values for experiment 2 are given in Table 9. This experiment studies the probability that a type 1 job that arrives at station 1 is rejected and lost (i.e. B_1) for different values of the routing probability $p_{1,2}^1$. Figure 12 then shows the results. A first thing to note is that the blocking probability B_1 increases as the routing probability $p_{1,2}^1$ increases. This is as expected, because more non-overflowed type 1 jobs will be present at station 2 when $p_{1,2}^1$ gets larger, which leads to a larger probability to find station 2 congested. Besides that, for all values of $p_{1,2}^1$, the blocking probability B_1 for the system with call packing is found to be larger than that for the system without call packing. This is in line with the discussion for numerical experiment 1 above, since $\mu_{11}^1 < \mu_{12}^1$ in this example.

Numerical experiment 3 ($B_{11,2}^1$ for different values of M_2^1)

Table 10 shows the parameter values for numerical experiment 3. This experiment considers the probability that a type 1 job that goes from station 1 to station 2 is blocked (i.e. $B_{11,2}^1$) for different values of M_2^1. Here, as discussed in Section 3.3, M_2^1 denotes the maximum number of type 1 jobs that can be present at the same time at station 2. The results are then given in Figure 13. It can be seen that the blocking probability $B_{11,2}^1$ is indeed smaller when more type 1 jobs are allowed to be present at station 2. Moreover, $B_{11,2}^1$ for the system with call packing is found to be smaller than $B_{11,2}^1$ for the system without call packing (or equal if $M_2^1 = 0$). This does not come as a

Table 9: Experiment 2

Parameter	Value
λ_1	15
λ_2	30
μ_{11}^1	1
μ_{12}^1	2
μ_{22}^1	3
μ_{22}^2	4
$p_{1,2}^1$	-
N_1	15
N_2	10
M_2^1	10
M_2^2	10

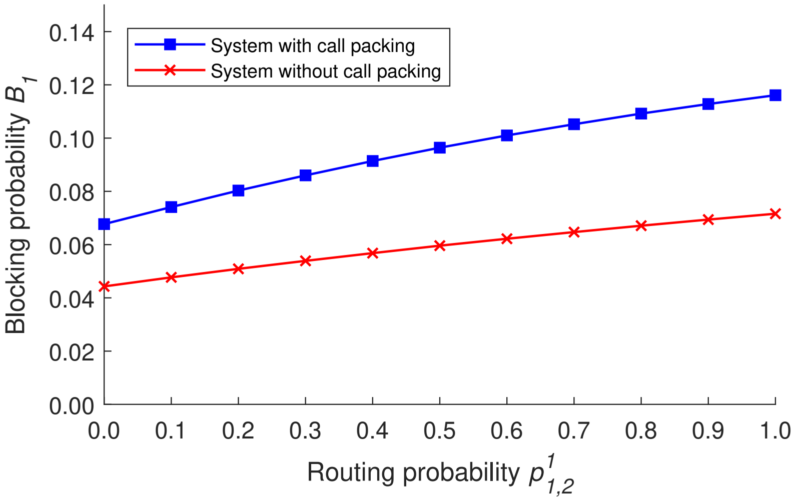


Figure 12: Results of experiment 2 (B_1 for different values of $p_{1,2}^1$)

Table 10: Experiment 3

Parameter	Value
λ_1	14
λ_2	16
μ_{11}^1	3
μ_{12}^1	2
μ_{22}^1	4
μ_{22}^2	6
$p_{1,2}^1$	1
N_1	7
N_2	10
M_2^1	-
M_2^2	10

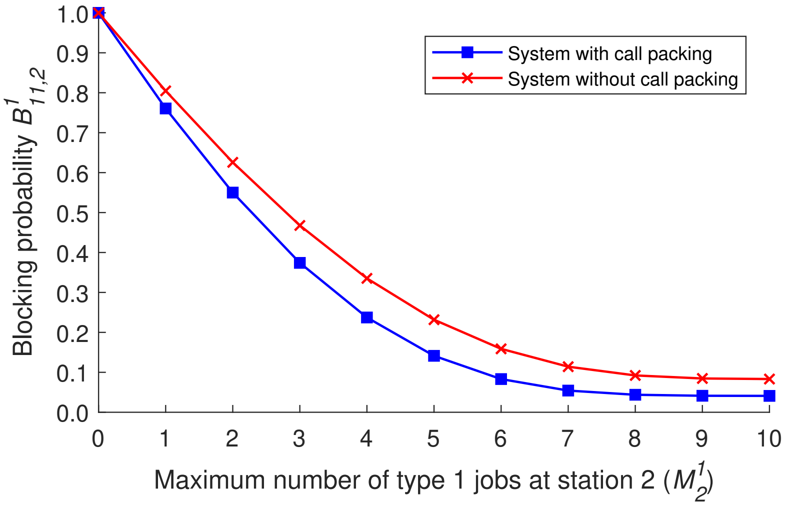


Figure 13: Results of experiment 3 ($B_{11,2}^1$ for different values of M_2^1)

surprise, mainly because of the following. In the system with call packing, an overflowed type 1 job, if present, immediately switches from station 2 to station 1 once a job at station 1 leaves. The servers at station 2 can therefore be expected to be less often occupied by overflowed type 1 jobs than in the system without call packing. Moreover, this also means that a type 1 job that goes from station 1 to station 2 cannot be blocked if at least one overflowed type 1 job is present at station 2, while this may occur in the system without call packing (see also Remark 12 in Section 3.3).

Concluding remarks

In this section and Appendix B.2, some numerical results of the blocking probabilities are given. These results illustrate how the blocking probabilities are affected by a change in the value of a certain input parameter (in particular, μ_{12}^1, $p_{1,2}^1$ and M_2^1). Moreover, they provide some insight into the effect of call packing on the blocking probabilities. For all numerical experiments that are performed, it is then concluded that the results coincide with the expectations based on intuition. Finally, it is noted that the discussion in this section does not take into account any additional time or costs that could be associated with call packing.

3.6 Insensitivity

In this section, it is studied whether the serial overflow system with call packing and serial overflow system without call packing are insensitive to the service time distributions. As discussed in Section 2.6, a queueing network is insensitive if the steady-state distribution does not depend on the service time distributions other than through their means.

First of all, in Section 3.6.1, discrete-event simulation is used to study whether the steady-state distributions are affected by the service time distribution. Secondly, in Section 3.6.2, the serial overflow system with call packing is shown to be insensitive if it is assumed that $\mu_{11}^1 = \mu_{12}^1$ and that the service is preemptively resumed after an overflowed type 1 job switches from station 2 to station 1.

3.6.1 Simulation

This section aims to determine whether the serial overflow system with call packing and serial overflow system without call packing can be expected to be insensitive to the service time distributions. Similar as in [60], it is therefore studied whether the steady-state distributions are affected when the service times are assumed to be lognormally distributed instead of exponentially distributed. To this end, we focus on a specific steady-state probability, which is the steady-state probability that the system is in a state $\mathbf{n} = (n_{11}^1, n_{12}^1, n_{22}^1, n_{22}^2)$ with $n_{11}^1 = N_1$ and $(n_{12}^1 + n_{22}^1 + 1, n_{22}^2) \notin \boldsymbol{C_2}$ (i.e. a state $\mathbf{n}$ for which, next to $n_{11}^1 = N_1$, at least one of $n_{12}^1 + n_{22}^1 = M_2^1$ and $n_{12}^1 + n_{22}^1 + n_{22}^2 = N_2$ holds). Because of the PASTA property of Poisson arrivals (see Section 2.5.1), this steady-state probability has the interpretation of the blocking probability of type 1 jobs at station 1 (i.e. B_1). In the sequel, this steady-state probability is therefore referred to as B_1.

If the serial overflow system with call packing and serial overflow system without call packing are insensitive, the steady-state probability B_1 should remain the same when the service times are lognormally distributed instead of exponentially distributed. In order to examine whether this is the case, two scenarios are considered in this section (see Table 11 for the parameter values). It is mentioned, though, that other scenarios are also studied and lead to similar conclusions.

First of all, if the service times are assumed to be exponential, the steady-state probability B_1 can be computed from one of the steady-state distributions that are obtained in Section 3.4. More specifically, for the system with call packing, B_1 can be determined from the product-form solution for the steady-state distribution in (39). Besides that, for the system without call packing, B_1 can be computed from the steady-state distribution that is obtained by applying the Gauss-Seidel method. This leads to the steady-state probabilities that are shown in Table 12.

Table 11: Insensitivity experiment: Parameter values

	λ_1	λ_2	μ_{11}^1	μ_{12}^1	μ_{22}^1	μ_{22}^2	$p_{1,2}^1$	N_1	N_2	M_2^1	M_2^2
Scenario 1	15	12	4	7	5	6	1	5	5	5	5
Scenario 2	15	12	4	4	5	6	1	5	5	5	5

Note: $\mu_{11}^1 \neq \mu_{12}^1$ for scenario 1, while $\mu_{11}^1 = \mu_{12}^1$ for scenario 2. Besides that, all times are assumed to be in hours.

Table 12: Insensitivity experiment: Steady-state probability B_1 (exponential case)

	System with call packing	System without call packing
Scenario 1	0.0829	0.0592
Scenario 2	0.0898	0.0636

Note: the steady-state probabilities for the system with call packing are determined from the product form (39). The steady-state probabilities for the system without call packing are computed from the steady-state distribution that is obtained by applying the Gauss-Seidel method.

Secondly, if the service times are assumed to be lognormally distributed, the steady-state probability B_1 can be determined by discrete-event simulation (see Appendix C for a description of the simulation procedure). Besides that, for verification purposes, discrete-event simulation is also used to determine B_1 under the assumption of exponential service times. The resulting steady-state probabilities are then given in Table 13. Moreover, 95% confidence intervals as based on the t-distribution are given between brackets (see Appendix C.2.3 for a discussion on the computation of these confidence intervals).

A first thing to note is that for the overflow system with call packing multiple cases can be considered (see also Remark 13 in Section 3.3):

- If an overflowed type 1 job switches from station 2 to station 1, several assumptions about the service time of this job can be made. For example, it can be assumed that the service has to be completely started over (resample) or can be preemptively resumed (resume). Because of the memoryless property of the exponential distribution, these two assumptions should lead to the same steady-state distribution if the service times are exponentially distributed. However, this does not hold true for lognormal service times. Therefore, both resampling and resuming are considered in this section. If the service is started over (resample), the service time is obtained by sampling from the service time distribution with mean $1/\mu_{11}^1$. Besides that, if the service is preemptively resumed (resume), the service time at station 1 is set equal to the residual service time multiplied by μ_{12}^1/μ_{11}^1 due to the difference in service speed. From the results in Table 13, it can then be seen that the steady-state probability B_1 is indeed similar for resume and resample if the service times follow the exponential distribution. On the other hand, this is not the case if the service times are lognormally distributed.
- If a server at station 1 becomes available and multiple overflowed type 1 jobs are present

Table 13: Insensitivity experiment: Steady-state probability B_1 according to simulation

Assumptions	Distribution	CV[a]	Scenario 1	Scenario 2
Call packing Resample FIFO	Exponential	1	0.0829 (0.0825-0.0833)[b]	0.0897 (0.0893-0.0901)
	Lognormal	0.2	0.0976 (0.0972-0.0979)	0.1009 (0.1005-0.1012)
	Lognormal	5	0.0699 (0.0696-0.0702)	0.0759 (0.0756-0.0762)
Call packing Resume FIFO	Exponential	1	0.0828 (0.0825-0.0832)	0.0895 (0.0892-0.0898)
	Lognormal	0.2	0.0820 (0.0817-0.0822)	0.0898 (0.0894-0.0902)
	Lognormal	5	0.0827 (0.0824-0.0830)	0.0898 (0.0894-0.0901)
Without call packing	Exponential	1	0.0592 (0.0589-0.0594)	0.0636 (0.0633-0.0639)
	Lognormal	0.2	0.0546 (0.0544-0.0548)	0.0585 (0.0583-0.0588)
	Lognormal	5	0.0627 (0.0624-0.0630)	0.0688 (0.0685-0.0691)

[a] Coefficient of variation. [b] 95% confidence interval as based on the t-distribution between brackets.

at station 2, it needs to be decided which of these jobs switches to station 1. For example, this job can be chosen to be the job that has been present for the longest time (first in, first out, FIFO) or the shortest time (last in, first out, LIFO). Another option could be to select an arbitrary job (random). Because of the memoryless property of the exponential distribution, this choice should not affect the steady-state distribution if the service times follow the exponential distribution. On the other hand, if the service times are lognormally distributed and $\mu_{11}^1 = \mu_{12}^1$, this may not be the case. In this section, it is assumed that the overflowed type 1 job that has been in service for the longest time switches to station 1 (i.e. FIFO). It is mentioned, though, that LIFO and random lead to similar conclusions.

Next, it can be seen that the steady-state probabilities in Table 13 do no significantly differ from the corresponding steady-state probabilities in Table 12 if the service times are exponentially distributed (see also Appendix C.3). By comparing these steady-state probabilities with those under the assumption of lognormal service times, it can then be studied whether the serial overflow system with call packing and serial overflow system without call packing can be expected to be insensitive.

First of all, the serial overflow system with call packing does not appear to be insensitive if the service is completely restarted after an overflowed type 1 job switches from station 2 to station 1 (resample). If the service times are lognormally distributed with a coefficient of variation of 0.2 or 5, the steady-state probability B_1 is significantly different from the corresponding steady-state probability in Table 12. This holds true for scenario 1 as well as scenario 2.

Besides that, the simulation results of this experiment as well as others seem to indicate that the serial overflow system with call packing is also not insensitive if the service is preemptively resumed after switching from station 2 to station 1 (resume) and $\mu_{11}^1 \neq \mu_{12}^1$ (see scenario 1). It is noted, though, that the steady-state probabilities do not appear to be much affected by the service time distribution. For example, if the service times are lognormally distributed with a coefficient of variation of 5, the steady-state probability B_1 is not found to be significantly different from B_1 for the exponential case. On the other hand, it appears that the serial overflow system with call packing might be insensitive if the service is preemptively resumed after switching from station 2 to station 1 (resume) and $\mu_{11}^1 = \mu_{12}^1$ (see scenario 2). In Section 3.6.2, this is formally proven.

Finally, the serial overflow system without call packing does not appear to possess the insensitivity property. For scenario 1 as well as scenario 2, the steady-state probability B_1 under the assumption of lognormal service times turns out to be significantly different from that under the assumption of exponential service times.

3.6.2 System with call packing: Proof of insensitivity (assuming $\mu_{11}^1 = \mu_{12}^1$)

In Section 3.6.1, it is studied whether the serial overflow system with call packing is insensitive, that is, whether the product form (39) is also valid if the service times are non-exponential. The simulation results then indicate that the system is not insensitive for all $\mu_{11}^1 \neq \mu_{12}^1$. On the other hand, there might be insensitivity if it is assumed that $\mu_{11}^1 = \mu_{12}^1$ and that the service is preemptively resumed after an overflowed type 1 job switches from station 2 to station 1. In this section, a proof for this last statement is provided. To this end, it is first shown that the product-form solution for the steady-state distribution remains valid if the service time distribution is given by a mixture of Erlang distributions. From this, the result can then be concluded for general service time distributions as well (see Remark 15 at the end of this section).

For the most part, the notation that is used in this section follows the notation that is given in Section 3.3. However, we also need some additional notation that is introduced below. The majority of this notation can be seen as an adapted version of the notation that is used in [58, 60].

First of all, if $\mu_{11}^1 = \mu_{12}^1$ and service is preemptively resumed after an overflowed type 1 job switches from station 2 to station 1, it is not necessary to distinguish between type 1 jobs that are served at station 1 and overflowed jobs that are served at station 2. Instead, we only need to keep track of the total number of type 1 jobs with primary station 1 (i.e. the type 1 jobs at station 1 and overflowed type 1 jobs at station 2). To this end, the following shorthand notation is introduced:

$$n_1^1 = n_{11}^1 + n_{12}^1; \qquad \mu_1^1 = \mu_{11}^1 = \mu_{12}^1 \tag{55}$$

Next, the distribution function of the service times is defined. As mentioned before, it is assumed that the service time distribution is given by a mixture of Erlang distributions. This means that the distribution function of the service times for type 1 jobs with primary station 1, denoted by G_1^1, is as follows:

$$G_1^1 = \sum_{k=1}^{\infty} q_1^1(k) E(k, \nu_1^1) \tag{56}$$

Here, $q_1^1(k)$ is the probability that the service distribution is an Erlang distribution of k exponential phases with parameter ν_1^1, which is denoted by $E(k, \nu_1^1)$.

Similarly, the distribution functions of non-overflowed type 1 and type 2 jobs at station 2 (denoted by G_{22}^1 and G_{22}^2, respectively) are given by:

$$G_{22}^t = \sum_{k=1}^{\infty} q_{22}^t(k) E(k, \nu_{22}^t), \qquad t = 1, 2 \tag{57}$$

Furthermore, the following notation is introduced for type 1 jobs with primary station 1:

$$\tau_1^1 = \sum_{k=1}^{\infty} q_1^1(k) \frac{k}{\nu_1^1} \quad \text{and} \quad \mu_1^1 = \frac{1}{\tau_1^1} \tag{58}$$

$$H_1^1(r) = \frac{\mu_1^1}{\nu_1^1} \sum_{k=r}^{\infty} q_1^1(k) \tag{59}$$

Here, τ_1^1 is the mean service requirement of type 1 jobs with primary station 1. Besides that, μ_1^1 is the mean service rate, which is similar to the parameter for the exponential case. Finally, the terms $H_1^1(\cdot)$ sum to one and can be interpreted as "steady-state probabilities for the number of residual exponential phases up to a next renewal in a discrete renewal process with (inter) renewal distribution function G_1^1" ([58], p. 19). From (59), it can also be verified that the following discrete renewal relation holds:

$$H_1^1(r) = H_1^1(r+1) + H_1^1(1) q_1^1(r) \tag{60}$$

Next, similar definitions can be given for non-overflowed type 1 and type 2 jobs at station 2:

$$\tau_{22}^t = \sum_{k=1}^{\infty} q_{22}^t(k) \frac{k}{\nu_{22}^t} \quad \text{and} \quad \mu_{22}^t = \frac{1}{\tau_{22}^t}, \qquad t = 1, 2 \tag{61}$$

$$H_{22}^t(r) = \frac{\mu_{22}^t}{\nu_{22}^t} \sum_{k=r}^{\infty} q_{22}^t(k), \qquad t = 1, 2 \tag{62}$$

$$H_{22}^t(r) = H_{22}^t(r+1) + H_{22}^t(1) q_{22}^t(r), \qquad t = 1, 2 \tag{63}$$

In order to describe the state of the system, the number of residual phases for each present job must be kept track of. For this purpose, each job is allocated a position or label l amongst the jobs in the same class, where three job classes are distinguished: type 1 jobs with primary station 1, non-overflowed type 1 jobs at station 2 and type 2 jobs at station 2. In order to refer to these job classes, superscripts and subscripts are added. More specifically, when there are n_1^1 type 1 jobs with primary station 1, n_{22}^1 non-overflowed type 1 jobs at station 2 and n_{22}^2 type 2 jobs at station 2 present, the following applies:

- $l_1^1 \in \{1, ..., n_1^1\}$ denotes the position of the l_1^1th type 1 job with primary station 1 (i.e. either a type 1 job at station 1 or overflowed type 1 job at station 2).
- $l_{22}^1 \in \{1, ..., n_{22}^1\}$ denotes the position of the l_{22}^1th non-overflowed type 1 job at station 2.
- $l_{22}^2 \in \{1, ..., n_{22}^2\}$ denotes the position of the l_{22}^2th type 2 job at station 2.

An arriving job is then randomly allocated a position amongst the jobs of the same class. Moreover, in order to make sure that the positions remain successive, a shift protocol is used. More specifically, if a job arrives and $n-1$ jobs in the same class are already present, the job is assigned position l amongst these jobs with probability $1/n$, for all $l = 1, ..., n$. The jobs that were previously at positions $l, ..., n-1$ then shift to positions $l+1, ..., n$. Besides that, if n jobs of a certain class are present and the job at position l amongst these jobs leaves, the jobs that were at positions $l+1, ..., n$ then shift to positions $l, ..., n-1$.

Now, the serial overflow system with call packing and mixtures of Erlang distributions as service time distributions can be represented by a continuous-time Markov chain. This Markov chain has the following state description:

$$\boldsymbol{R} = [\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}], \quad \text{where} \quad \boldsymbol{X} = (x_1, ..., x_{n_1^1}), \quad \boldsymbol{Y} = (y_1, ..., y_{n_{22}^1}) \quad \text{and} \quad \boldsymbol{Z} = (z_1, ..., z_{n_{22}^2}) \tag{64}$$

Here, $\boldsymbol{X}$ is a $1 \times n_1^1$ vector whose l_1^1th element denotes that the job at position l_1^1 amongst the jobs with primary station 1 has $x_{l_1^1}$ residual exponential phases, each with parameter ν_1^1 ($l_1^1 = 1, ..., n_1^1$). $\boldsymbol{Y}$ and $\boldsymbol{Z}$ have a similar interpretation.

Moreover, the state space of the Markov chain, denoted by $\boldsymbol{S_d}$, is equal to:

$$\begin{aligned} \boldsymbol{S_d} = \{ [\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}] \mid ([n_1^1 - N_1]^+ + n_{22}^1, n_{22}^2) \in \boldsymbol{C_2},\ & x_{l_1^1} = 1, 2, ... \quad (l_1^1 = 1, ..., n_1^1), \\ y_{l_{22}^1} = 1, 2, ... \quad (l_{22}^1 = 1, ..., n_{22}^1),\ & z_{l_{22}^2} = 1, 2, ... \quad (l_{22}^2 = 1, ..., n_{22}^2) \} \end{aligned} \tag{65}$$

Finally, some shorthand notation that is used in the proof of Theorem 3 is mentioned. It is noted that, although not explicitly introduced below, similar notation is also used for $\boldsymbol{Y}$ and $\boldsymbol{Z}$. The shorthand notation is then as follows:

$$\boldsymbol{X} - (x_{l_1^1})_{l_1^1} = (x_1, ..., x_{l_1^1 - 1}, x_{l_1^1 + 1}, ..., x_{n_1^1}) \tag{66}$$

$$\boldsymbol{X} - (x_{l_1^1})_{l_1^1} + (x_{l_1^1} + 1)_{l_1^1} = (x_1, ..., x_{l_1^1 - 1}, (x_{l_1^1} + 1), x_{l_1^1 + 1}, ..., x_{n_1^1}) \tag{67}$$

$$\boldsymbol{X} + (1)_{l_1^1} = (x_1, ..., x_{l_1^1 - 1}, 1, x_{l_1^1}, x_{l_1^1 + 1}, ..., x_{n_1^1}) \tag{68}$$

Here, in (66) the job at position l_1^1 is deleted, and the jobs that were at positions $l_1^1 + 1, ..., n_1^1$ are moved to positions $l_1^1, ..., n_1^1 - 1$. Besides that, in (67) the number of residual phases for the job at position l_1^1 is changed from $x_{l_1^1}$ to $x_{l_1^1} + 1$. Finally, in (68) a job with one residual phase is added at position l_1^1, and the jobs that were at positions $l_1^1 + 1, ..., n_1^1$ are moved to positions $l_1^1 + 1, ..., n_1^1 + 1$.

The following detailed product form can now be proven. From this detailed product form, in turn, the insensitivity result that is aimed for can be concluded (see Corollary 1).

Theorem 3 (A detailed product-form result)**.** *Consider the serial overflow system with call packing, where $\mu_{11}^1 = \mu_{12}^1 = \mu_1^1$ and each of the service time distributions is a mixture of Erlang distributions (i.e. the distribution functions are as in (56) and (57)). Let c be a normalizing constant and the state space $\boldsymbol{S_d}$ be as given in (65). For $\boldsymbol{R} = [\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}] \in \boldsymbol{S_d}$, the following detailed product form, denoted by π_d, then applies:*

$$\pi_d(\boldsymbol{R}) = c \frac{1}{n_1^1!} \prod_{l_1^1 = 1}^{n_1^1} \left\{ \frac{\lambda_1}{\mu_1^1} H_1^1(x_{l_1^1}) \right\} \frac{1}{n_{22}^1!} \prod_{l_{22}^1 = 1}^{n_{22}^1} \left\{ \frac{p_{1,2}^1 \lambda_1}{\mu_{22}^1} H_{22}^1(y_{l_{22}^1}) \right\} \frac{1}{n_{22}^2!} \prod_{l_{22}^2 = 1}^{n_{22}^2} \left\{ \frac{\lambda_2}{\mu_{22}^2} H_{22}^2(z_{l_{22}^2}) \right\} \tag{69}$$

Proof. In order to prove the detailed product form (69), the global balance equations are again formulated and verified.

First of all, the rate out of state $\boldsymbol{R} \in \boldsymbol{S_d}$ is given by (70.1) + ... + (70.5):

$$1_{\{n_1^1>0\}} \sum_{l_1^1=1}^{n_1^1} \pi_d([\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}])\nu_1^1 \tag{70.1}$$

$$1_{\{n_{22}^1>0\}} \sum_{l_{22}^1=1}^{n_{22}^1} \pi_d([\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}])\nu_{22}^1 \tag{70.2}$$

$$1_{\{n_{22}^2>0\}} \sum_{l_{22}^2=1}^{n_{22}^2} \pi_d([\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}])\nu_{22}^2 \tag{70.3}$$

$$\sum_{l_1^1=1}^{n_1^1+1} \pi_d([\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}])\frac{1}{n_1^1+1}\lambda_1 \left(1_{\{n_1^1<N_1\}} + 1_{\{n_1^1\geq N_1\}}1_{\{([n_1^1-N_1]+n_{22}^1+1, n_{22}^2)\in \boldsymbol{C_2}\}}\right) \tag{70.4}$$

$$\sum_{l_{22}^2=1}^{n_{22}^2+1} \pi_d([\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}])\frac{1}{n_{22}^2+1}\lambda_2 1_{\{([n_1^1-N_1]^+ + n_{22}^1, n_{22}^2+1)\in \boldsymbol{C_2}\}} \tag{70.5}$$

Similarly, the rate into state $\boldsymbol{R} \in \boldsymbol{S_d}$ is given by (70.1)′ + ... + (70.5)′ (note that (70.4)′ is divided into three parts):

$$\begin{aligned}1_{\{n_1^1>0\}} \sum_{l_1^1=1}^{n_1^1} \Big\{&\pi_d([\boldsymbol{X} - (x_{l_1^1})_{l_1^1}, \boldsymbol{Y}, \boldsymbol{Z}])\frac{1}{n_1^1}q_1^1(x_{l_1^1})\lambda_1 + \\ &\pi_d([\boldsymbol{X} - (x_{l_1^1})_{l_1^1} + (x_{l_1^1}+1)_{l_1^1}, \boldsymbol{Y}, \boldsymbol{Z}])\nu_1^1\Big\}\end{aligned} \tag{70.1$'$}$$

$$\begin{aligned}1_{\{n_{22}^1>0\}} \sum_{l_{22}^1=1}^{n_{22}^1} \left\{ \sum_{l_1^1=1}^{n_1^1+1} \left(\pi_d([\boldsymbol{X} + (1)_{l_1^1}, \boldsymbol{Y} - (y_{l_{22}^1})_{l_{22}^1}, \boldsymbol{Z}])p_{1,2}^1 \frac{1}{n_{22}^1} q_{22}^1(y_{l_{22}^1})\nu_1^1\right) + \right.\\ \left. \pi_d([\boldsymbol{X}, \boldsymbol{Y} - (y_{l_{22}^1})_{l_{22}^1} + (y_{l_{22}^1}+1)_{l_{22}^1}, \boldsymbol{Z}])\nu_{22}^1 \right\}\end{aligned} \tag{70.2$'$}$$

$$\begin{aligned}1_{\{n_{22}^2>0\}} \sum_{l_{22}^2=1}^{n_{22}^2} \Big\{&\pi_d([\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z} - (z_{l_{22}^2})_{l_{22}^2}])\frac{1}{n_{22}^2}q_{22}^2(z_{l_{22}^2})\lambda_2 + \\ &\pi_d([\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z} - (z_{l_{22}^2})_{l_{22}^2} + (z_{l_{22}^2}+1)_{l_{22}^2}])\nu_{22}^2\Big\}\end{aligned} \tag{70.3$'$}$$

$$\left\{\begin{aligned}&\sum_{l_1^1=1}^{n_1^1+1} \pi_d([\boldsymbol{X} + (1)_{l_1^1}, \boldsymbol{Y}, \boldsymbol{Z}])p_{1,0}^1\nu_1^1 \left(1_{\{n_1^1<N_1\}} + 1_{\{n_1^1\geq N_1\}}1_{\{([n_1^1-N_1]+n_{22}^1+1, n_{22}^2)\in \boldsymbol{C_2}\}}\right) + && \text{(70.4a)}'\\ &\sum_{l_1^1=1}^{n_1^1+1} \pi_d([\boldsymbol{X} + (1)_{l_1^1}, \boldsymbol{Y}, \boldsymbol{Z}])p_{1,2}^1\nu_1^1 1_{\{n_1^1<N_1\}}1_{\{(n_{22}^1+1, n_{22}^2)\notin \boldsymbol{C_2}\}} + && \text{(70.4b)}'\\ &\sum_{l_{22}^1=1}^{n_{22}^1+1} \pi_d([\boldsymbol{X}, \boldsymbol{Y} + (1)_{l_{22}^1}, \boldsymbol{Z}])\nu_{22}^1 1_{\{([n_1^1-N_1]^+ + n_{22}^1+1, n_{22}^2)\in \boldsymbol{C_2}\}} && \text{(70.4c)}'\end{aligned}\right.$$

$$\sum_{l^2_{22}=1}^{n^2_{22}+1} \pi_d([\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z}+(1)_{l^2_{22}}])\nu^2_{22} 1_{\{([n^1_1-N_1]^+ + n^1_{22}, n^2_{22}+1)\in \boldsymbol{C_2}\}} \tag{70.5)$'$}$$

Subsequently, it can be verified that $(70.i) = (70.i)'$, $i = 1, ..., 5$, by substitution of the detailed product form (69). See Appendix A.2 for technical details of this verification. Therefore, since $(70.1) + ... + (70.5) = (70.1)' + ... + (70.5)'$, it follows that the global balance equations are satisfied by substituting the detailed product form (69). This completes the proof. □

Next, the result in Theorem 3 can be used to show that the product form (39) is also valid when each of the service time distributions is a mixture of Erlang distributions.

Corollary 1 (Corollary of Theorem 3)**.** *Consider the serial overflow system with call packing, where $\mu^1_{11} = \mu^1_{12} = \mu^1_1$. Moreover, let the distribution functions of the service times be as given in* (56) *and* (57)*, that is, let each of the service time distributions be a mixture of Erlang distributions. The same product-form solution for the steady-state distribution then applies as in case of exponential service times. More specifically, the steady-state distribution is as given in* (39) *in Theorem 2. Therefore, as $\mu^1_{11} = \mu^1_{12} = \mu^1_1$, the steady-state distribution $\pi = (\pi(\mathbf{n}),\ \mathbf{n} \in \boldsymbol{S})$ is as follows:*

$$\pi(\mathbf{n}) = c\frac{1}{(n^1_{11}+n^1_{12})!}\left(\frac{\lambda_1}{\mu^1_1}\right)^{n^1_{11}}\left(\frac{\lambda_1}{\mu^1_1}\right)^{n^1_{12}}\frac{1}{n^1_{22}!}\left(\frac{p^1_{1,2}\lambda_1}{\mu^1_{22}}\right)^{n^1_{22}}\frac{1}{n^2_{22}!}\left(\frac{\lambda_2}{\mu^2_{22}}\right)^{n^2_{22}}, \quad \mathbf{n} \in \boldsymbol{S} \tag{71}$$

Here, c is a normalizing constant, $\mathbf{n} = (n^1_{11}, n^1_{12}, n^1_{22}, n^2_{22})$ and $\boldsymbol{S}$ the state space, which is:

$$\boldsymbol{S} = \{\mathbf{n} \mid n^1_{11} < N_1,\ n^1_{12} = 0,\ (n^1_{22}, n^2_{22}) \in \boldsymbol{C_2} \ \ \textit{or} \ \ n^1_{11} = N_1,\ (n^1_{12} + n^1_{22}, n^2_{22}) \in \boldsymbol{C_2}\} \tag{72}$$

Proof. Consider an arbitrary state $\mathbf{n} = (n^1_{11}, n^1_{12}, n^1_{22}, n^2_{22}) \in \boldsymbol{S}$, and let $n^1_1 = n^1_{11} + n^1_{12}$. In order to prove the result, we first determine the steady-state probability that there are n^1_1 type 1 jobs with primary station 1 (i.e. non-overflowed jobs at station 1 and overflowed jobs at station 2), n^1_{22} non-overflowed type 1 jobs at station 2 and n^2_{22} type 2 jobs at station 2 present, which is denoted by $P(n^1_1, n^1_{22}, n^2_{22})$. To this end, the detailed product form (69) is summed over all states with n^1_1 type 1 jobs with primary station 1, n^1_{22} non-overflowed type 1 jobs at station 2 and n^2_{22} type 2 jobs at station 2. It can be noted that this means that we have to sum over all possible phases $x_{l^1_1}$, $y_{l^1_{22}}$ and $z_{l^2_{22}}$ for all positions $l^1_1 = 1, ..., n^1_1$, $l^1_{22} = 1, ..., n^1_{22}$ and $l^2_{22} = 1, ..., n^2_{22}$. This yields the following:

$$\begin{aligned}P(n^1_1, n^1_{22}, n^2_{22}) = c\sum_{x_1=1}^{\infty}\cdots\sum_{x_{n^1_1}=1}^{\infty}\frac{1}{n^1_1!}\prod_{l^1_1=1}^{n^1_1}\left\{\frac{\lambda_1}{\mu^1_1}H^1_1(x_{l^1_1})\right\}\times \\ \sum_{y_1=1}^{\infty}\cdots\sum_{y_{n^1_{22}}=1}^{\infty}\frac{1}{n^1_{22}!}\prod_{l^1_{22}=1}^{n^1_{22}}\left\{\frac{p^1_{1,2}\lambda_1}{\mu^1_{22}}H^1_{22}(y_{l^1_{22}})\right\}\times \\ \sum_{z_1=1}^{\infty}\cdots\sum_{z_{n^2_{22}}=1}^{\infty}\frac{1}{n^2_{22}!}\prod_{l^2_{22}=1}^{n^2_{22}}\left\{\frac{\lambda_2}{\mu^2_{22}}H^2_{22}(z_{l^2_{22}})\right\}\end{aligned}$$

$$
\begin{aligned}
&= c\frac{1}{n_1^1!}\left(\frac{\lambda_1}{\mu_1^1}\right)^{n_1^1}\frac{1}{n_{22}^1!}\left(\frac{p_{1,2}^1\lambda_1}{\mu_{22}^1}\right)^{n_{22}^1}\frac{1}{n_{22}^2!}\left(\frac{\lambda_2}{\mu_{22}^2}\right)^{n_{22}^2}\sum_{x_1=1}^{\infty}\cdots\sum_{x_{n_1^1}=1}^{\infty}\prod_{l_1^1=1}^{n_1^1}\left\{H_1^1(x_{l_1^1})\right\}\times\\
&\qquad\sum_{y_1=1}^{\infty}\cdots\sum_{y_{n_{22}^1}=1}^{\infty}\prod_{l_{22}^1=1}^{n_{22}^1}\left\{H_{22}^1(y_{l_{22}^1})\right\}\sum_{z_1=1}^{\infty}\cdots\sum_{z_{n_{22}^2}=1}^{\infty}\prod_{l_{22}^2=1}^{n_{22}^2}\left\{H_{22}^2(z_{l_{22}^2})\right\}\\
&= c\frac{1}{n_1^1!}\left(\frac{\lambda_1}{\mu_1^1}\right)^{n_1^1}\frac{1}{n_{22}^1!}\left(\frac{p_{1,2}^1\lambda_1}{\mu_{22}^1}\right)^{n_{22}^1}\frac{1}{n_{22}^2!}\left(\frac{\lambda_2}{\mu_{22}^2}\right)^{n_{22}^2}\prod_{l_1^1=1}^{n_1^1}\left\{\sum_{x_{l_1^1}=1}^{\infty}H_1^1(x_{l_1^1})\right\}\times\\
&\qquad\prod_{l_{22}^1=1}^{n_{22}^1}\left\{\sum_{y_{l_{22}^1}=1}^{\infty}H_{22}^1(y_{l_{22}^1})\right\}\prod_{l_{22}^2=1}^{n_{22}^2}\left\{\sum_{z_{l_{22}^2}=1}^{\infty}H_{22}^2(z_{l_{22}^2})\right\}\\
&= c\frac{1}{n_1^1!}\left(\frac{\lambda_1}{\mu_1^1}\right)^{n_1^1}\frac{1}{n_{22}^1!}\left(\frac{p_{1,2}^1\lambda_1}{\mu_{22}^1}\right)^{n_{22}^1}\frac{1}{n_{22}^2!}\left(\frac{\lambda_2}{\mu_{22}^2}\right)^{n_{22}^2}
\end{aligned}
\tag{73}
$$

Here, the last step follows by noting that the terms $H_1^1(\cdot)$, $H_{22}^1(\cdot)$ and $H_{22}^2(\cdot)$ sum to one.

Next, it is noted that in the overflow system with call packing there is only one possibility to have n_1^1 type 1 jobs with primary station 1, which is if $n_{11}^1 = \min\{n_1^1, N_1\}$ and $n_{12}^1 = [n_1^1 - N_1]^+$. Therefore, the steady-state probability $\pi(\mathbf{n})$ is equal to the steady-state probability $P(n_1^1, n_{22}^1, n_{22}^2)$. Hence, by substitution of $n_{11}^1 + n_{12}^1$ for n_1^1 in (73), the following expression for π can be concluded:

$$
\begin{aligned}
\pi(\mathbf{n}) &= P(n_{11}^1 + n_{12}^1, n_{22}^1, n_{22}^2)\\
&= c\frac{1}{(n_{11}^1 + n_{12}^1)!}\left(\frac{\lambda_1}{\mu_1^1}\right)^{n_{11}^1}\left(\frac{\lambda_1}{\mu_1^1}\right)^{n_{12}^1}\frac{1}{n_{22}^1!}\left(\frac{p_{1,2}^1\lambda_1}{\mu_{22}^1}\right)^{n_{22}^1}\frac{1}{n_{22}^2!}\left(\frac{\lambda_2}{\mu_{22}^2}\right)^{n_{22}^2}, \qquad \mathbf{n}\in\boldsymbol{S}
\end{aligned}
\tag{74}
$$

This completes the proof. □

Remark 15 (General service time distributions). In Corollary 1, the product form (39) is shown to remain valid when each of the service time distributions is a mixture of Erlang distributions. It is then noted that an arbitrary, non-negative and continuous distribution can be arbitrarily closely approximated by a mixture of Erlang distributions (see e.g. [55, 58, 60] and references therein). By using similar arguments as in [58, 60], the result of Corollary 1 can therefore be expected to remain valid for general service time distributions. Hence, this implies that the overflow system with call packing is insensitive to the service time distributions if it is assumed that $\mu_{11}^1 = \mu_{12}^1$ and that the service is preemptively resumed after an overflowed type 1 job switches from station 2 to station 1.

3.7 Conclusions

This chapter considers a two-station overflow system that stands out in the following ways:

- It has a serial structure, which means that jobs that complete service at the primary station may also be routed to the secondary station.
- It allows for service rates that are dependent on the station at which the job is served, the job type and whether or not the job is overflowed (i.e. $\mu_{11}^1 \neq \mu_{12}^1 \neq \mu_{22}^1 \neq \mu_{22}^2$).

- It includes both overflow and jump-over blocking. More specifically, if (primary) station 1 is fully occupied, arriving type 1 jobs are overflowed to (secondary) station 2. On the other hand, if all servers at station 2 are occupied, arriving type 1 jobs that come from station 1 are rejected and leave the system, which can be seen as if they 'jump over' station 2.
- It assumes that the number of jobs that can be present at station 2 is restricted to a coordinate convex set $\boldsymbol{C_2}$.

This is in contrast with most studies regarding overflow systems in literature, since these mainly consider systems with a parallel structure. Moreover, some restrictions on the parameters are also not uncommon.

The overflow system is then studied under two different assumptions. The first assumption that is considered is that overflowed type 1 jobs immediately switch from (secondary) station 2 to (primary) station 1 once a server at this station becomes available (i.e. the system with call packing). Secondly, the system is also considered under the assumption that overflowed type 1 jobs finish service at (secondary) station 2, even if a server at (primary) station 1 becomes available (i.e. the system without call packing).

The following results are then obtained:

- The joint steady-state distribution of the number of jobs that are present in the overflow system is determined. For the system with call packing, a product-form solution for the steady-state distribution is obtained. For the system without call packing, the steady-state distribution is determined by applying the Gauss-Seidel method or GTH algorithm.
- For both the system with call packing and the system without call packing, the blocking probabilities of interest (b_1, B_1, O_1, B_2, $B^1_{11,2}$ and $B^1_{1,2}$) are determined from the obtained steady-state distribution. This is done either by exploiting the PASTA property of Poisson arrivals (b_1, B_1, O_1 and B_2) or by computation of a Palm probability ($B^1_{11,2}$ and $B^1_{1,2}$). Next, as an illustration, numerical results of the blocking probabilities are also given.
- It is studied by discrete-event simulation whether the overflow system can be expected to be insensitive to the service time distributions. If this is the case, the steady-state distribution is not dependent on the service time distributions other than through their means. It then appears that both the system with call packing and the system without call packing are not insensitive. The simulation results indicate, though, that the system with call packing is not very sensitive if service is preemptively resumed after an overflowed type 1 job switches from station 2 to station 1. Moreover, if additionally it is assumed that $\mu^1_{11} = \mu^1_{12}$, the system with call packing might even be insensitive. A proof for this statement is provided by showing that, under these conditions, the product-form solution for the steady-state distribution is also valid when each of the service time distributions is a mixture of Erlang distributions.

Chapter 4

An application: ICU-SDU modelling

In Chapter 3, an overflow system with serial structure is studied. This chapter aims to illustrate a possible application of such an overflow system. To this end, it is described how an adapted version of the overflow system in Chapter 3 could be useful to model the interaction between an intensive care unit (ICU) and a step-down unit (SDU).

4.1 Outline of chapter

This chapter is outlined as follows:

- Section 4.2 describes the notation that is used in this chapter.
- In Section 4.3, a description of ICUs and SDUs is provided. Besides that, some studies that consider ICUs and/or SDUs from a queueing perspective are described. Finally, a discussion of papers that are concerned with overflow in a health care system is also included.
- In Section 4.4, the ICU-SDU system is described. Moreover, it is discussed how the system relates to the overflow system that is studied in Chapter 3.
- Section 4.5 discusses how a product-form solution for the joint steady-state distribution of the number of patients in the ICU and SDU can be obtained.
- In Section 4.6, it is briefly discussed how the steady-state distribution can be of help to obtain insight into the (expected) performance of the ICU-SDU system. In particular, it is described how the probability to find the ICU and/or SDU fully occupied can be computed.
- In Section 4.7, the chapter concludes with a discussion and conclusions.

Finally, it is noted that an alternative proof of Theorem 4 is provided in Appendix A.4.

4.2 Notation

This section discusses the notation that is used in this chapter. It is noted that the notation is in line with the notation that is used in Chapter 3 (see Section 3.2), where station 1 corresponds to the ICU and station 2 to the SDU. For completeness, a complete description of the notation is given in this section. Beforehand, it is again mentioned that a subscript is used to refer to a station, while a superscript is used for referring to the patient type. Here, as discussed in Section 4.4 (see Assumption A2), the patient type depends on whether the patient arrives from outside the system at the ICU (type 1) or at the SDU (type 2).

The notation for the input parameters is then as follows:

Table 14: Notation: Input parameters

λ_1:	Arrival rate of patients at station 1 (ICU), $\lambda_1 \geq 0$.
λ_2:	Arrival rate of patients at station 2 (SDU), $\lambda_2 \geq 0$.
μ^1_{11}:	Service rate of type 1 patients with primary station 1 (ICU) that are present at station 1 (ICU), $\mu^1_{11} > 0$.
μ^1_{12}:	Service rate of type 1 patients with primary station 1 (ICU) that are present at station 2 (SDU), $\mu^1_{12} > 0$.
μ^1_2:	Service rate of type 1 patients with primary station 2 (SDU) that are present in the SDU or ICU, $\mu^1_2 > 0$.[a]
μ^2_2:	Service rate of type 2 patients with primary station 2 (SDU) that are present in the SDU or ICU, $\mu^2_2 > 0$.[a]
$p^1_{1,2}$:	Probability that a type 1 patients with primary station 1 (ICU) is routed to station 2 (SDU) after finishing service, $p^1_{1,2} \in [0, 1]$.[b]
$p^1_{1,0}$:	Probability that a type 1 patient with primary station 1 (ICU) leaves the system after finishing service, $p^1_{1,0} = 1 - p^1_{1,2}$.[b]
N_1:	Number of operational beds at station 1 (ICU), $N_1 \in \mathbb{N}$.
N_2:	Number of operational beds at station 2 (SDU), $N_2 \in \mathbb{N}$.

[a] The service rates of overflowed and non-overflowed patients with primary station 2 (SDU) are assumed to be the same. Therefore, they are only dependent on the primary station (and not the actual station).

[b] The routing probabilities for overflowed and non-overflowed patients are assumed to be equal. Hence, they only depend on the primary station (and not on the actual station).

Besides that, in order to denote the state of the system, the following notation is used:

Table 15: Notation: State of the system

n^1_{11}:	Number of type 1 patients with primary station 1 (ICU) that are present at station 1 (ICU), $n^1_{11} \in \mathbb{N}$.
n^1_{12}:	Number of type 1 patients with primary station 1 (ICU) that are present at station 2 (SDU), $n^1_{12} \in \mathbb{N}$.
n^1_2:	Number of present (non-overflowed and overflowed) type 1 patients with primary station 2 (SDU), $n^1_2 \in \mathbb{N}$.[a]
n^2_2:	Number of present (non-overflowed and overflowed) type 2 patients with primary station 2 (SDU), $n^2_2 \in \mathbb{N}$.[a]
$\mathbf{n}$:	State of the system, $\mathbf{n} = (n^1_{11}, n^1_{12}, n^1_2, n^2_2)$.
$\boldsymbol{S}$:	Set of admissible states.
$\pi(\mathbf{n})$:	Steady-state probability to observe state $\mathbf{n}$, where $\mathbf{n} \in \boldsymbol{S}$.

[a] As the service rates of overflowed and non-overflowed patients with primary station 2 (SDU) are assumed to be equal, only the total number of patients with primary station 2 (SDU) is kept track of (see also Section 4.4).

Moreover, the following notation also comes in handy:

Table 16: Notation: Some additional notation

$\mathbf{n} - e^1_{11}$:	The same state with n^1_{11} decreased by one, that is, $\mathbf{n} - e^1_{11} = (n^1_{11} - 1, n^1_{12}, n^1_2, n^2_2)$.
$\mathbf{n} - e^1_{12}$:	The same state with n^1_{12} decreased by one, that is, $\mathbf{n} - e^1_{12} = (n^1_{11}, n^1_{12} - 1, n^1_2, n^2_2)$.
$\mathbf{n} - e^1_2$:	The same state with n^1_2 decreased by one, that is, $\mathbf{n} - e^1_2 = (n^1_{11}, n^1_{12}, n^1_2 - 1, n^2_2)$.
$\mathbf{n} - e^2_2$:	The same state with n^2_2 decreased by one, that is, $\mathbf{n} - e^2_2 = (n^1_{11}, n^1_{12}, n^1_2, n^2_2 - 1)$.
$1_{\{C\}}$:	Indicator that is 1 if condition C is true and 0 otherwise.
$[E]^+$:	Expression that is 0 if $E \leq 0$ and E otherwise.

Finally, in Section 4.6, the computation of the blocking probabilities is illustrated by two examples. These blocking probabilities are denoted as follows:

Table 17: Notation: Blocking probabilities

b_1:	Probability that a patient who arrives from outside at station 1 (ICU) is blocked and lost or overflowed to station 2 (SDU).
B_1:	Probability that a patient who arrives from outside at station 1 (ICU) is blocked and lost.

4.3 Literature

In this section, some literature related to ICUs and SDUs is discussed. First of all, Section 4.3.1 provides a description of ICUs and SDUs. Secondly, in order to provide some insight into mathematical modelling of ICUs and SDUs, Section 4.3.2 discusses several studies that consider ICUs and/or SDUs from the viewpoint of queueing theory. Finally, Section 4.3.3 contains a brief discussion of papers that are concerned with overflow in health care systems.

4.3.1 ICUs and SDUs

Patients are admitted to an ICU or SDU if they require more intensive monitoring, treatment and care than that is possible in a general ward. These patients are either emergency patients, whose arrivals are unscheduled, or elective patients, whose admissions are scheduled (see e.g. [19, 39, 59]).

In this section, a brief description of ICUs and SDUs is then given. See, for example, [44, 64] and references therein for a more extensive discussion.

Intensive care units

In an ICU, monitoring, treatment and care is provided for patients who are critically ill. More precisely, according to [64] (p. 8, translated from Dutch), an ICU patient can be defined as "a patient with one or more acutely threatened or disturbed vital functions for which continuous monitoring is necessary and treatment of a potentially reversible condition can lead to recovery of vital functions". This means that specialized medical staff, such as intensivists and intensive care (IC) nurses, and specialized medical equipment, such as ventilators, should be present in an ICU, so that the required care can be provided (see e.g. [2, 64]).

Arrivals of ICU patients can occur in a variety of ways. For example, arriving ICU patients can be patients with a deteriorating condition who were already present in a general ward, patients from the operating theatre who require ICU care after a major surgery or patients who arrive at the ICU after having a serious accident. However, because ICUs are costly to operate, it may occur that no staffed ICU bed is available upon arrival of an ICU patient. Depending on the situation, this can have several consequences (see e.g. [39]). For example, the surgery of an elective patient may be postponed, a patient may be transferred to an ICU at another hospital or an over-bed may be created (see also Example 1 in Section 1.1.1).

Step-down units

An SDU provides a level of monitoring and care that is intermediate between the level that is provided in an ICU and the level that is provided in a general ward. This means that the number of nurses per patient (i.e. the nurse-to-patient ratio) in an SDU is generally smaller than in an ICU. Besides that, an SDU may not be able to provide specific organ support, such as invasive ventilation (see e.g. [2, 44]). Most SDU patients can be categorized into three groups (see [44], p. 1212). First of all, SDU patients could be "step-down" patients, who come from the ICU and no longer have full intensive care requirements, but still need more intensive monitoring and care than that can be provided in a general ward. Secondly, SDU patients could also be "step-up" patients, who come from a general ward or the Emergency Department (ED) and require a higher level of care. Lastly, the third main group of SDU patients consists of postoperative patients, who are admitted to the SDU from the operating theatre.

Finally, it is noted that various terminology is used to describe units in which an intermediate level of monitoring and care is provided (see e.g. [2, 44] and references therein). For example, SDUs (or related units) are also known as intermediate care units, transitional care units and high dependency care units. Besides that, different definitions of SDUs (or related units) can also be found (see e.g. [44], pp. 1210-1212), which implies that the exact use of step-down units may differ across hospitals and countries. For example, this also appears from [2], in which it is mentioned that in many hospitals the SDU is staffed by critical-care nurses, but that this is not the case for all hospitals.

4.3.2 Mathematical modelling of ICUs and SDUs

Hospital wards have frequently been studied from the viewpoint of queueing theory. This section briefly describes some of these studies, where the main focus lies on papers that are related to ICUs and/or SDUs. First of all, several studies that consider the ICU or SDU in isolation are discussed. Next, some papers that study an ICU-SDU system are described.

Modelling the ICU and/or SDU in isolation

There are many studies that use a mathematical (queueing) model to analyze the ICU and/or SDU in isolation. Below, some of these studies are briefly mentioned.

There are several studies that model the ICU and/or SDU as an Erlang loss system or, using Kendall's notation, $M|G|N|N$ queue, where N is the number of beds in the unit (see also Example 2 in Section 2.2.1). For example, in [19], several hospital wards (including a medical ICU, surgical ICU and Medium Care Unit, which can be seen as similar to an SDU) are modelled as an Erlang loss system. Furthermore, in [42], the $M|G|N|N$ queue is used to study a medical-surgical ICU. Another paper that is worthwhile to mention in this context is [59]. In this reference, a tandem queue with an ICU and operating theatre is studied. It is then shown that the blocking probability for an $M|G|N|N$ queue can be regarded as a (reasonably accurate) lower bound for the ICU rejection probability when taking into account the interaction between the ICU and operating theatre.

If the ICU and/or SDU are modelled as an Erlang loss system, it is assumed that patients are blocked if all beds in the unit are occupied. Instead, it could also be assumed that patients wait until a bed becomes available. As a consequence, there are also several studies that model a hospital ward as an (Erlang) delay system. For example, this includes [23], in which the $M|M|N$ queue (with N again the number of beds in the unit) is used to predict ICU congestion and delays. Another example is [25]. In this reference, an ICU is modelled as an $M|H|N$ queue, where the H in the second field means that the service times are assumed to be hyperexponential.

Finally, it is noted that many other studies that consider a mathematical (queueing) model of an ICU in isolation can be found. See, for example, the references in the papers that are mentioned above for a further discussion.

Modelling ICU-SDU systems

Instead of modelling the ICU and/or SDU in isolation, it can also be an option to consider an ICU-SDU system. This would make it possible to take into account the interaction between the units, but it has as disadvantage that it usually leads to a more complicated model. Below, two papers that consider an ICU-SDU system are discussed.

First of all, [41] considers a system that consists of a medical ICU and SDU at a tertiary-care hospital. Historical data are used to estimate the arrival rate, length of stay, which consists of a "service time" and "time to transfer", and acuity level, which is either "acute" or "subacute". Acute patients must be admitted to the ICU, while subacute patients are assigned an SDU bed if possible. However, if the SDU is fully occupied, subacute patients could also be admitted to the ICU. If no suitable bed is available upon arrival of a patient, the patient waits in a priority queue (based on the acuity level and whether the patient arrives from the ED or not). In order to study the performance of this ICU-SDU system, a discrete-event simulation model is then developed. In this way, the effect of several hypothetical scenarios (e.g. reserving ICU beds for acute patients or expansion of the ICU) on the wait times and occupancy is studied.

Secondly, in [2], an ICU-SDU system is modelled as a queueing network that incorporates waiting, abandonment, balking and bumping. Similar as in [41], a distinction is made between two patient groups: "critical patients", who arrive at the ICU, and "semi-critical patients", who can be treated in the SDU. The ICU-SDU system is then examined by using fluid and diffusion analysis. Furthermore, simulation is used to study a more complex ICU-SDU system. This makes it possible to include several additional features, such as critical patients who are "off-placed" in the SDU if the ICU is congested, direct SDU arrivals (next to arrivals of step-down patients from the ICU) and readmissions.

4.3.3 Overflow in health care systems

In this chapter, it is assumed that SDU patients can be overflowed to the ICU if all SDU beds are occupied, and vice versa (see Assumptions A4 and A5 in Section 4.4). The question that then arises is under which conditions these assumptions can be made. Therefore, this section contains a discussion on ICUs, SDUs and overflow. Furthermore, some studies that consider an overflow system in a hospital setting are also discussed.

ICU, SDU and overflow

In this chapter, an ICU-SDU system is modelled as an overflow system. Below, a brief discussion on ICUs, SDUs and overflow is therefore included.

First of all, it appears reasonable to assume that SDU patients can be overflowed to the ICU. For example, in [2] (pp. 859-860), it is mentioned that ICU beds can be regarded as "flexible servers", since ICUs can provide care for critically ill patients, but also for semi-critical patients. In this reference, it is therefore assumed that semi-critical patients can be treated in the SDU or in the ICU. The length of stay of semi-critical patients does then not depend on whether they are treated in the ICU or in the SDU. Besides that, in [41], it is assumed that subacute patients can be admitted to the ICU if no SDU bed is available (see also Section 4.3.2 for a further discussion of these papers).

Secondly, it is not always possible to assume that ICU patients can be overflowed to the SDU, since specialized medical equipment (e.g. ventilators) and qualified personnel (e.g. IC nurses) should then be present in the SDU. As a consequence, an SDU is often not equipped to provide care for ICU patients, which means that overflow of ICU patients to the SDU is not always a reasonable assumption. For example, in [41], it is assumed that acute patients can only be admitted to the ICU.

Nevertheless, in certain cases, it could be reasonable to assume that ICU patients can be overflowed to the SDU. For example, the simulation model that is considered in [2] relies on the assumption that critical patients can be "off-placed" in the SDU if the ICU is fully occupied. If an ICU bed becomes available, an off-placed critical patient may be transferred from the SDU to the ICU. However, it is mentioned that the SDU cannot provide the same level of monitoring and care for critical patients as they would receive in the ICU, which leads to a reduction of the quality of care ([2], p. 860). In this reference, the mean length of stay of critical patients is therefore multiplied by a factor x if the patient is off-placed in the SDU, where x is chosen equal to 1.5.

Furthermore, in [44] (pp. 1212-1213), it is mentioned that SDU beds could be located in a standalone unit, but also colocated within an ICU or general ward. In the latter case, the SDU beds can be separate beds that are reserved for patients who require intermediate care or "flexible" beds whose designation may change based on the needs of the patient. Therefore, if the SDU beds are designated, "flexible" beds that are colocated within an ICU, it may be reasonable to assume that ICU patients can be overflowed to the SDU.

Overflow systems in a hospital setting

Below, some papers that are concerned with overflow systems in a hospital setting are briefly discussed. An example of such a paper is [9]. In this reference, several policies are studied and compared, among which separate wards and earmarking. In the latter case, it is assumed that there is a number of "earmarked" beds for each patient type, which can only be occupied by patients of this specific type. Besides that, there is an overflow ward with fully flexible beds, which can be occupied by all patients irrespective of their type. It is then mentioned that “the earmarked beds should always be used as much as possible” ([9], p. 457). Hence, this means that patients can be transferred from the overflow ward to an earmarked bed if an earmarked bed becomes available.

Besides that, in [63], a short stay unit (SSU) and multiple inpatient wards are modelled as an overflow system, where the wards correspond to the primary stations and the SSU to the secondary station. More specifically, urgent patients who arrive at ward i are overflowed to the SSU if they find all beds in this ward occupied. It is then assumed that these patients may be repatriated from the SSU to ward i with a rate γ_i, where $0 \leq \gamma_i \leq \infty$. Moreover, it is noted that some other patient streams, such as direct arrivals at the SSU, are also incorporated.

Another paper that considers overflow in a hospital setting is [29]. In this reference, a system with multiple primary wards and multiple overflow wards is studied. More specifically, the primary wards consist of dedicated wards for each specialty. Besides that, the specialties are divided into clusters, and specialties that are in the same cluster then share one overflow ward. Patients who find all beds in the primary ward occupied go to the corresponding overflow ward. If the overflow ward is also fully occupied, they are rejected.

The references that are discussed up to now study systems with overflow wards that are located within the same hospital. Instead, it could also occur that a patient is overflowed to a ward at another hospital. For example, [39] considers a queueing network with multiple ICUs at different hospitals in the same region and an extra (virtual) ICU, which consists of reserved beds that are distributed over the ICUs. This extra (virtual) ICU is then specifically meant to care for (overflowed) regional emergency patients who are rejected at an original ICU. In [16], a similar overflow system with several ICUs is considered. Finally, in this context, it is also worthwhile to mention [3]. In this reference, a network of neonatal hospitals is modelled as a loss network with overflow.

4.4 Model and assumptions

In this section, the ICU-SDU system that is studied in this chapter is described. The notation follows the notation that is given in Section 4.2. The ICU-SDU system is then depicted in Figure 14. It is noted that the system is similar to the serial overflow system with call packing that is studied in Chapter 3 (see also Remark 16 at the end of this section).

Below, the assumptions of the model are discussed. Here, it is noted that each assumption is accompanied by some comments that explain why the assumption could be reasonable. The model then relies on the following assumptions:

(A1) The system consists of an ICU (station 1) and SDU (station 2). The ICU has N_1 operational beds, while N_2 operational beds are present in the SDU.

Comments: It is common to describe the capacity of a hospital ward in terms of operational or staffed beds. These are beds for which both the necessary medical equipment and personnel are available (see e.g. [9, 19]).

(A2) Patients arrive from outside the system at the ICU and SDU according to a Poisson process with rates λ_1 and λ_2, respectively. Patients who arrive from outside at the ICU are referred

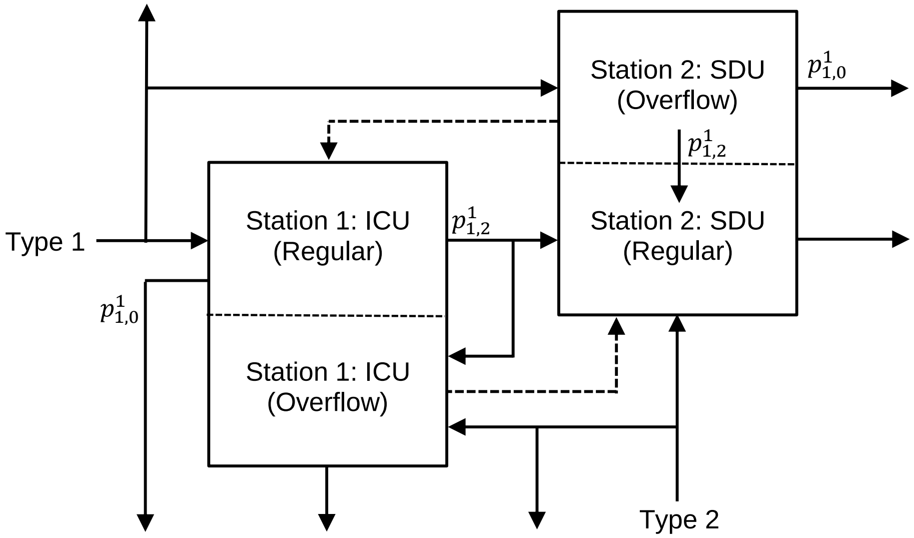


Figure 14: The ICU-SDU system that is studied in this chapter. Routing probabilities ($p^1_{1,k}$, $k = 0, 2$) are mentioned next to the applicable arrows

to as type 1 patients. Similarly, patients arriving from outside at the SDU are called type 2 patients.

Comments: In literature, it is frequently assumed that arrivals at the ICU and/or SDU (or a related unit) follow a Poisson process (see e.g. [2, 16, 19, 23, 25, 39, 41, 42, 59]). In particular, the arrival process of emergency patients is found to be well approximated by a Poisson process. Moreover, the Poisson process can be considered as a reasonable assumption for the arrival process of elective patients as well (see e.g. [19, 39] for a further discussion).

(A3) Patients stay in the ICU or SDU for an exponential length of stay (see also Assumption A7). After service completion, ICU patients become an SDU patient with probability $p^1_{1,2}$, while they leave the system with probability $p^1_{1,0}$, where $p^1_{1,0} = 1 - p^1_{1,2}$. SDU patients always leave the system after their service is completed.

Comments: It can be noted that transfers of ICU patients who become SDU patients (i.e. step-down patients) are incorporated. On the other hand, since transfers of SDU patients who become ICU patients can be expected to occur less often, these are not taken into account. It is noted that this assumption is in line with, for example, [2] (see Section 4.3.2 for a further discussion of this paper).

(A4) If arriving type 1 patients find all N_1 ICU beds occupied, they are overflowed to the SDU. If the SDU is also fully occupied, arriving type 1 patients are blocked and leave the system.

Comments: First of all, it is not always possible to assume that ICU patients can be overflowed to the SDU, since this would require the presence of specialized medical staff and equipment in the SDU. However, as more widely discussed in Section 4.3.3, overflow of ICU patients to the SDU may be a reasonable assumption in certain cases. An example of such a situation is when no (separate) SDU is present, and SDU care is provided in the ICU. The beds in the ICU could then be designated as either an ICU bed or an SDU bed, where the main difference between these beds is the nurse-to-patient ratio. Secondly, it is not uncommon to assume that arriving ICU patients are blocked if all beds are occupied (see e.g. [19, 42, 59]). In practice, this could mean, for example, that the patient is transferred to another hospital.

(A5) As mentioned above, arriving SDU patients are either finished ICU patients (type 1) or patients who arrive from outside the system (type 2). First of all, overflowed ICU patients who were already present in the SDU can always be admitted, since they already occupy an SDU bed. On the other hand, type 1 patients from the ICU and type 2 patients from outside may find the SDU fully occupied. In this case, type 1 patients stay in the ICU (note that this is always possible, since they already occupy an ICU bed), and type 2 patients are overflowed to the ICU (note that this may not be possible). If all ICU beds are also occupied, arriving type 2 patients are blocked and leave the system.

Comments: First of all, as more extensively discussed in Section 4.3.3, it can generally be assumed that SDU patients can also be admitted to the ICU if the SDU is fully occupied. Secondly, it is also not unusual to assume that patients who arrive at a hospital ward are blocked if all beds are occupied (see e.g. [9, 19, 29, 42, 59]).

(A6) An overflowed ICU patient in the SDU, if present, immediately goes to the ICU and preemptively resumes service once an operational bed in the ICU becomes available. Similarly, an overflowed SDU patient in the ICU (of type 1 or type 2), if present, goes to the SDU and preemptively resumes service as soon as an operational bed in the SDU becomes available.

Comments: This assumption basically says that overflow capacity is only used if own capacity is not available, which implies that patients are treated in their primary unit as much as possible. In this sense, this assumption can seen as similar to the assumption in [9] (p. 457) for the earmarking policy, which states that "earmarked beds should always be used as much as possible" (see Section 4.3.3 for a further discussion of this paper). Besides that, in [2], it is assumed that critical patients who are off-placed in the SDU may be transferred to the ICU if an ICU bed becomes available. In this reference, however, semi-critical patients who are treated in the ICU do not go to the SDU if an SDU bed becomes available (although they may be bumped to the SDU by an arriving critical patient). Finally, in [63], overflowed patients who are present in the short stay unit may be repatriated to their primary ward if a bed in this ward becomes available (see again Section 4.3.3 for a further discussion of this paper). In this reference, however, it is assumed that transfers do not occur immediately, but after a certain delay ([63], p. 22). Hence, although not exactly the same, these papers make a similar assumption. Therefore, also considering analytical tractability, this assumption may be considered as a reasonable (simplifying) assumption.

(A7) The lengths of stay in the ICU and SDU are assumed to be exponentially distributed with rates μ_{11}^1 for non-overflowed ICU patients in the ICU, μ_{12}^1 for overflowed ICU patients in the SDU, μ_2^1 for SDU patients of type 1 and μ_2^2 for SDU patients of type 2. It can thus be noted that non-overflowed and overflowed ICU patients may have a different mean length of stay (i.e. $\mu_{11}^1 \neq \mu_{12}^1$). On the other hand, it is assumed that SDU patients who are overflowed to the ICU have the same service parameter as non-overflowed SDU patients of the same type. However, the mean length of stay of SDU patients may depend on whether the patient is a step-down patient (type 1) or comes from outside the system (type 2), that is, it is allowed that $\mu_2^1 \neq \mu_2^2$.

Comments: First of all, for analytical tractability, it is not uncommon that the length of stay in the ICU or SDU is assumed to be exponentially distributed (see e.g. [2, 16, 23, 39, 59]). It is noted, though, that in several references it is found that the length of stay in the ICU is better described by another distribution, such as the lognormal distribution (see e.g. [39]) or hyperexponential distribution (see e.g. [25]). Secondly, the assumption that it is allowed that $\mu_{11}^1 \neq \mu_{12}^1$, while non-overflowed and overflowed SDU patients of the same type are assumed to have the same mean length of stay is in line with the discussion in Section 4.3.3.

It can thus be noted that several simplifying assumptions are made. Some of these are already discussed above (e.g. exponential lengths of stay), while some are not explicitly mentioned (e.g. no distinction is made between elective and emergency arrivals, and readmissions are regarded as new arrivals from outside). It is expected that (most of) the assumptions that are not explicitly mentioned are in line with literature. For example, in among others [19, 39, 41, 42, 59], a similar assumption regarding readmissions seems to be made.

Subsequently, the state of the system is given by $\mathbf{n} = (n_{11}^1, n_{12}^1, n_2^1, n_2^2)$, where:

- n_{11}^1: The number of (non-overflowed) ICU patients who are present in the ICU.
- n_{12}^1: The number of (overflowed) ICU patients who are present in the SDU.
- n_2^1: The number of SDU patients of type 1 who are present (i.e. the sum of the number of non-overflowed type 1 patients in the SDU and overflowed type 1 patients in the ICU).
- n_2^2: The number of SDU patients of type 2 who are present (i.e. the sum of the number of non-overflowed type 2 patients in the SDU and overflowed type 2 patients in the ICU).

It can thus be noted that three patient groups (or classes) are distinguished: ICU patients of type 1, SDU patients of type 1 and SDU patients of type 2. Moreover, since it is allowed that $\mu_{11}^1 \neq \mu_{12}^1$ (see Assumption A7), a distinction is made between non-overflowed ICU patients in the ICU and overflowed ICU patients in the SDU. On the other hand, such a distinction is not made for SDU patients of type 1 and SDU patients of type 2, since it is assumed that non-overflowed and overflowed SDU patients of the same type have the same mean length of stay (see Assumption A7).

Finally, the section ends with a remark that describes how the ICU-SDU system in Figure 14 relates to the overflow system that is studied in Chapter 3.

Remark 16 (ICU-SDU system and overflow system in Chapter 3). It can be noted that the ICU-SDU system as described above is similar to the serial overflow system with call packing that is studied in Chapter 3. However, it differs from the system in Chapter 3 in two points. First of all, for simplicity, the coordinate convex structure at station 2 (SDU) is now left out of account. Secondly, it is assumed that type 1 and type 2 patients can also be overflowed to the ICU if the SDU is fully occupied. This is different from the serial overflow system with call packing in Chapter 3, in which it is assumed that arrivals of type 1 jobs that come from station 1 and type 2 jobs from outside the system are blocked if no server at station 2 is available.

4.5 Steady-state distribution

For the ICU-SDU system that is described in Section 4.4, a product-form solution for the joint steady-state distribution of the number of patients in the system can be obtained. To this end, it is first noted that, because of Assumption A6, there can only be overflowed ICU patients present in the SDU if the ICU is fully occupied, and vice versa. Moreover, a situation with overflowed SDU patients in the ICU and overflowed ICU patients in the SDU at the same time cannot occur. Therefore, for every admissible state $\mathbf{n}$, one of the following conditions must apply:

- No overflowed patients ($n_{11}^1 \leq N_1$, $n_{12}^1 = 0$ and $n_2^1 + n_2^2 \leq N_2$).
- Overflowed ICU patients in the SDU ($n_{11}^1 = N_1$, $n_{12}^1 > 0$ and $n_2^1 + n_2^2 < N_2$).
- Overflowed SDU patients in the ICU ($n_{11}^1 < N_1$, $n_{12}^1 = 0$ and $n_2^1 + n_2^2 > N_2$).

As a consequence, the state space, denoted by $\boldsymbol{S}$, is as follows:

$$\begin{aligned}\boldsymbol{S} = \{\mathbf{n} \mid n_{12}^1 = 0,\ 0 \leq n_{11}^1 \leq N_1,\ 0 \leq n_2^1 + n_2^2 \leq N_2 \ \text{ or}\\ n_{12}^1 > 0,\ n_{11}^1 = N_1,\ 0 \leq n_{12}^1 + n_2^1 + n_2^2 \leq N_2 \ \text{ or}\\ n_{12}^1 = 0,\ 0 \leq n_{11}^1 + [n_2^1 + n_2^2 - N_2] \leq N_1,\ n_2^1 + n_2^2 > N_2\}\end{aligned} \tag{75}$$

Now, a product-form solution for the steady-state distribution can be obtained.

Theorem 4 (ICU-SDU system: Product form)**.** *The ICU-SDU system as described in Section 4.4 has the following steady-state distribution $\pi = (\pi(\mathbf{n}),\ \mathbf{n} \in \boldsymbol{S})$:*

$$\pi(\mathbf{n}) = cF(n_{12}^1)\frac{1}{n_{11}^1!}\left(\frac{\lambda_1}{\mu_{11}^1}\right)^{n_{11}^1}\frac{1}{n_2^1!}\left(\frac{p_{1,2}^1\lambda_1}{\mu_2^1}\right)^{n_2^1}\frac{1}{n_2^2!}\left(\frac{\lambda_2}{\mu_2^2}\right)^{n_2^2}, \quad \mathbf{n} = (n_{11}^1, n_{12}^1, n_2^1, n_2^2) \in \boldsymbol{S} \tag{76}$$

Here, c is a normalizing constant, $\boldsymbol{S}$ the state space as defined in (75) and F a function, which is as follows:

$$F(n) = \begin{cases} (\lambda_1)^n / \prod_{k=1}^n (N_1\mu_{11}^1 + k\mu_{12}^1) & n > 0 \\ 1 & n = 0 \end{cases} \tag{77}$$

Proof. The result follows by showing that the global balance equations (78) are satisfied for each $\mathbf{n} \in \boldsymbol{S}$ by substitution of the product form (76). Here, beforehand, it is noted that, similar as in the proof of Theorem 2, the terms in the left-hand side and right-hand side of the global balance equations are ordered in such a way that each part has an interpretation of either a flow out of $\mathbf{n}$ (left-hand side) or flow into state $\mathbf{n}$ (right-hand side). Moreover, this makes it possible to show that the global balance equations are satisfied by verifying specific class balances.

Table 18: Verification of the global balance equations in (78)

Patient class	Class balance
Non-overflowed ICU patients of type 1 in ICU	(78.1) = (78.9)
Overflowed ICU patients of type 1 in SDU	(78.2) = (78.10)
SDU patients of type 1[(a)]	(78.3) = (78.11) + (78.12) + (78.13)
SDU patients of type 2[(a)]	(78.4) = (78.14)
Type 1 patients at the outside	(78.5) + (78.6) = (78.15) + (78.16) + (78.17) + (78.18)
Type 2 patients at the outside	(78.7) + (78.8) = (78.19) + (78.20)

[(a)] Consisting of both non-overflowed SDU patients in the SDU and overflowed SDU patients in the ICU.

The global balance equations are then as follows (for $\mathbf{n} \in \boldsymbol{S}$):

$$\begin{array}{lr}
\left\{\begin{array}{l}
\pi(\mathbf{n}) n_{11}^1 \mu_{11}^1 1_{\{n_{11}^1>0\}} 1_{\{n_{12}^1=0\}} + \\
\pi(\mathbf{n}) (N_1\mu_{11}^1 + n_{12}^1\mu_{12}^1) 1_{\{n_{11}^1=N_1\}} 1_{\{n_{12}^1>0\}} + \\
\pi(\mathbf{n}) n_2^1 \mu_2^1 1_{\{n_2^1>0\}} + \\
\pi(\mathbf{n}) n_2^2 \mu_2^2 1_{\{n_2^2>0\}} + \\
\pi(\mathbf{n}) \lambda_1 1_{\{n_{11}^1<N_1\}} + \\
\pi(\mathbf{n}) \lambda_1 1_{\{n_{11}^1=N_1\}} 1_{\{n_{12}^1+n_2^1+n_2^2<N_2\}} + \\
\pi(\mathbf{n}) \lambda_2 1_{\{n_{12}^1+n_2^1+n_2^2<N_2\}} + \\
\pi(\mathbf{n}) \lambda_2 1_{\{n_{11}^1+[n_2^1+n_2^2-N_2]<N_1\}} 1_{\{n_2^1+n_2^2\geq N_2\}}
\end{array}\right\}
&
\begin{array}{r}
(78.1)\\ (78.2)\\ (78.3)\\ (78.4)\\ (78.5)\\ (78.6)\\ (78.7)\\ (78.8)
\end{array}
\\
= & (78)
\\
\left\{\begin{array}{l}
\pi(\mathbf{n} - e_{11}^1) \lambda_1 1_{\{n_{11}^1>0\}} 1_{\{n_{12}^1=0\}} + \\
\pi(\mathbf{n} - e_{12}^1) \lambda_1 1_{\{n_{11}^1=N_1\}} 1_{\{n_{12}^1>0\}} + \\
\pi(\mathbf{n} + e_{11}^1 - e_2^1) p_{1,2}^1 (n_{11}^1+1) \mu_{11}^1 1_{\{n_{11}^1<N_1\}} 1_{\{n_2^1>0\}} 1_{\{n_2^1+n_2^2\leq N_2\}} + \\
\pi(\mathbf{n} + e_{11}^1 - e_2^1) p_{1,2}^1 (n_{11}^1+1) \mu_{11}^1 1_{\{n_2^1>0\}} 1_{\{n_2^1+n_2^2>N_2\}} + \\
\pi(\mathbf{n} + e_{12}^1 - e_2^1) p_{1,2}^1 (N_1\mu_{11}^1 + (n_{12}^1+1)\mu_{12}^1) 1_{\{n_{11}^1=N_1\}} 1_{\{n_2^1>0\}} 1_{\{n_2^1+n_2^2\leq N_2\}} + \\
\pi(\mathbf{n} - e_2^2) \lambda_2 1_{\{n_2^2>0\}} + \\
\pi(\mathbf{n} + e_{11}^1) p_{1,0}^1 (n_{11}^1+1) \mu_{11}^1 1_{\{n_{11}^1<N_1\}} + \\
\pi(\mathbf{n} + e_{12}^1) p_{1,0}^1 (N_1\mu_{11}^1 + (n_{12}^1+1)\mu_{12}^1) 1_{\{n_{11}^1=N_1\}} 1_{\{n_{12}^1+n_2^1+n_2^2<N_2\}} + \\
\pi(\mathbf{n} + e_2^1) (n_2^1+1) \mu_2^1 1_{\{n_{12}^1+n_2^1+n_2^2<N_2\}} + \\
\pi(\mathbf{n} + e_2^1) (n_2^1+1) \mu_2^1 1_{\{n_{11}^1+[n_2^1+n_2^2-N_2]<N_1\}} 1_{\{n_2^1+n_2^2\geq N_2\}} + \\
\pi(\mathbf{n} + e_2^2) (n_2^2+1) \mu_2^2 1_{\{n_{12}^1+n_2^1+n_2^2<N_2\}} + \\
\pi(\mathbf{n} + e_2^2) (n_2^2+1) \mu_2^2 1_{\{n_{11}^1+[n_2^1+n_2^2-N_2]<N_1\}} 1_{\{n_2^1+n_2^2\geq N_2\}}
\end{array}\right\}
&
\begin{array}{r}
(78.9)\\ (78.10)\\ (78.11)\\ (78.12)\\ (78.13)\\ (78.14)\\ (78.15)\\ (78.16)\\ (78.17)\\ (78.18)\\ (78.19)\\ (78.20)
\end{array}
\end{array}$$

Now, the product form (76) can be substituted in the global balance equations (78), after which it can be verified that these are satisfied. As in the proof of Theorem 2, this can be done by verifying specific class balance equations. These class balance equations are given in Table 18 and can be verified in a similar manner as in the proof of Theorem 2. Then, since the class balances in Table 18 are satisfied, it immediately follows that the global balance equations (41) are also satisfied for all $\mathbf{n} \in \boldsymbol{S}$. Hence, this completes the proof. □

Remark 17 (Alternative proof of Theorem 4). The proof of Theorem 4 that is given above is based on the global balance equations. It is noted that the product form (76) could also be concluded from the product-form result in [26]. Moreover, it can then be shown that the ICU-SDU system is insensitive if $\mu_{11}^1 = \mu_{12}^1$, that is, the product form (76) is also valid for non-exponential service times if $\mu_{11}^1 = \mu_{12}^1$. In Appendix A.4, further details are provided.

Remark 18 (Insensitivity). As mentioned in Remark 17 and shown in Appendix A.4, the ICU-SDU system that is studied in this chapter is insensitive to the service time distributions if it is assumed that $\mu_{11}^1 = \mu_{12}^1$. It can be expected that this result could also be proven along similar lines as in the proof in Section 3.6.2, but this is not fully worked out. If $\mu_{11}^1 \neq \mu_{12}^1$, in contrast, the ICU-SDU system of interest is conjectured to be (slightly) sensitive based on a short simulation study. This is thus similar to the results for the serial overflow system that is subject of Chapter 3 (see Section 3.6).

4.6 Blocking probabilities and other performance measures

From the product-form solution for the steady-state distribution that is obtained in Section 4.5, several performance measures can be obtained. For example, it could be of interest to determine the fraction of time that overflowed ICU patients are present in the SDU or the probability that at least K overflowed ICU patients are present in the SDU (for $K \in \{1, ..., N_2\}$). Besides that, blocking probabilities can also be computed. These blocking probabilities can be obtained in a similar manner as for the overflow system that is studied in Chapter 3 (see Section 3.5.2).

The main difference is that it is now also possible that overflowed patients are present in the ICU (station 1). This means that the blocking probabilities that can be of interest are slightly different than those for the overflow system in Chapter 3 (see Section 3.5.1). For example, these now include the probability that an arriving type 2 patient at the SDU finds the SDU congested and is overflowed to the ICU and the probability that a step-down patient from the ICU finds the SDU congested and stays in the ICU. Furthermore, because of the possible presence of overflowed SDU patients in the ICU, the computation of the blocking probabilities is also slightly different. Below, the computation is illustrated for two blocking probabilities, which are b_1 and B_1.

First of all, the probability that an arriving type 1 patient at the ICU finds the ICU fully occupied and is blocked and lost or overflowed to the SDU, denoted by b_1, can be computed as follows:

$$b_1 = \sum_{\mathbf{n} \in \boldsymbol{S}_{\boldsymbol{b_1}}} \pi(\mathbf{n}), \text{ where } \boldsymbol{S}_{\boldsymbol{b_1}} = \{\mathbf{n} \in \boldsymbol{S} \mid n_{11}^1 + [n_2^1 + n_2^2 - N_2]^+ = N_1\} \tag{79}$$

Secondly, the probability that an arriving type 1 patient finds both the ICU and SDU fully occupied and is blocked and lost, denoted by B_1, can be determined as follows:

$$B_1 = \sum_{\mathbf{n} \in \boldsymbol{S}_{\boldsymbol{B_1}}} \pi(\mathbf{n}), \text{ where } \boldsymbol{S}_{\boldsymbol{B_1}} = \{\mathbf{n} \in \boldsymbol{S} \mid n_{11}^1 + [n_2^1 + n_2^2 - N_2]^+ = N_1, \; n_{12}^1 + n_2^1 + n_2^2 \geq N_2\} \tag{80}$$

4.7 Discussion and conclusions

In this chapter, an ICU-SDU system is modelled as an overflow system with serial structure. This overflow system can be seen as similar to the serial overflow system with call packing that is studied in Chapter 3 (see also Remark 16 in Section 4.4). It is then shown that a product-form solution for the joint steady-state distribution of the number of patients in the system can be found. From this product form, several performance measures, such as blocking probabilities, can then be computed. This could be useful to get a (quick) indication of the (expected) performance of the ICU-SDU system.

Another way to obtain (quick) insight into the performance of ICUs and SDUs is to model the units in isolation. For this purpose, several single-station queueing systems, such as the Erlang loss system, have been proposed in literature (see also Section 4.3.2). Compared with these single-station queueing systems, the ICU-SDU system as studied in this chapter has the advantage that it takes into account the interaction between the ICU and SDU. This is done in two ways. First of all, it is assumed that SDU patients can be admitted to the ICU if the SDU is fully occupied, and vice versa. Besides that, transfers of step-down patients who leave the ICU and go to the SDU are incorporated. Therefore, the overflow system that is studied in Chapter 3, in adapted form, could be useful to model the interaction between an ICU and SDU. However, some critical comments should also be made.

First of all, it is important to note that (most of) the model assumptions are based on scientific literature regarding ICUs and SDUs and that the model is not tested in practice for a (real) ICU-SDU system. It would therefore be useful to study whether a specific, real ICU-SDU system can be accurately described by the model as presented in Section 4.4. In particular, it would be interesting to test whether the observed performance measures can be well approximated by the model.

Secondly, it is assumed that ICU patients can be admitted to the SDU if the ICU is fully occupied. This means that specialized medical staff and equipment should be present in the SDU. As a consequence, it is not expected that the ICU-SDU system as studied in this chapter provides an adequate representation of all ICU-SDU systems. Nevertheless, in certain cases, this assumption may be reasonable. An example is the situation in which no (separate) SDU is present, and SDU care is provided in the ICU. In this case, the beds in the ICU could be designated as either an ICU bed or an SDU bed, where the main difference between these beds is the nurse-to-patient ratio (see Section 4.3.3 for a more extensive discussion).

Thirdly, in order to obtain an analytical (product-form) result, some simplifying assumptions are made. For example, it is assumed that the lengths of stay are exponentially distributed, and that overflowed patients are always immediately transferred to their primary unit once a bed at this unit becomes available. It could therefore be interesting to study the effect of these simplifying assumptions. For this purpose, discrete-event simulation could be a viable approach, since it allows for more complicated modelling assumptions.

Chapter 5

Conclusion

In this chapter, the objectives are briefly restated, the main findings are summarized, and some suggestions for further research are given.

Overflow system of interest and objectives

This report is concerned with the study of queueing systems with overflow. In particular, the focus is on the two-station overflow loss system with two job types that is formally described in Section 3.3. This system has the following distinguishing characteristics. First of all, it has a serial structure in the sense that it is possible that jobs that finish service at (primary) station 1 also go to (secondary) station 2. Secondly, the service rates may depend on the job type, the station at which the job is served and whether or not the job is overflowed (i.e. $\mu^1_{11} \neq \mu^1_{12} \neq \mu^1_{22} \neq \mu^2_{22}$). Thirdly, overflow as well as jump-over blocking are included. Finally, it is assumed that the number of jobs that can be present station 2 is restricted to a coordinate convex set.

The overflow system is then studied under the assumption of (immediate) repacking or call packing. This means that overflowed jobs at (secondary) station 2 immediately switch to (primary) station 1 once a server at this station becomes available. Besides that, the system without the assumption of call packing is also considered. In this case, overflowed jobs always finish service at (secondary) station 2, even if a server at (primary) station 1 becomes available.

As more widely discussed in Section 1.2, the objectives are to determine the joint steady-state distribution of the number of jobs in the overflow system, to compute and compare blocking probabilities, and to examine the feature of insensitivity. Moreover, it is aimed to illustrate a possible application of (an adapted version of) the overflow system to ICU-SDU modelling. Below, the main findings are then summarized chapter by chapter.

Summary of main findings

First of all, before studying the overflow system of interest, some theoretical background is provided in Chapter 2. In this chapter, several concepts, models and methods from queueing theory that are useful for the research project are identified and described. In particular, these include stochastic

Table 19: Summary of results that are obtained in Chapter 3

	Overflow system with call packing	Overflow system without call packing
Steady-state distribution	Determined by obtaining a product-form solution.	Determined by using Gauss-Seidel method and GTH algorithm.
Blocking probabilities	Determined by using PASTA property or computing a Palm probability. Illustrated by providing numerical results.	Determined by using PASTA property or computing a Palm probability. Illustrated by providing numerical results.
(In)sensitivity	Simulation indicates that the system is not insensitive for all $\mu_{11}^1 \neq \mu_{12}^1$. It is then shown that the system is insensitive if $\mu_{11}^1 = \mu_{12}^1$ and resume are assumed.	Simulation indicates that the system is sensitive.

processes, Markov chains, queueing networks, product forms, the Gauss-Seidel method, the GTH algorithm, the PASTA property, Palm probabilities, discrete-event simulation and insensitivity. Besides that, an overflow system with parallel structure is studied as an illustration, and some other related literature regarding overflow systems is briefly discussed.

Secondly, in Chapter 3, the overflow system of interest is studied as to steady-state distribution, blocking probabilities and insensitivity. This is done for the system with call packing as well as the system without call packing. The results of this analysis are summarized in Table 19. Besides that, Section 3.7 contains a more extensive discussion of the results.

Thirdly, Chapter 4 aims to illustrate a possible practical use of the overflow system of interest by studying an application to ICU-SDU modelling. Literature that is concerned with ICUs and SDUs is discussed, which provides insight into which assumptions are reasonable to make. It is then found that an adapted version of the overflow system could be useful to obtain a (quick) indication of several performance measures, such as blocking probabilities, if it can be assumed that ICU patients may be overflowed to the SDU in case of a congested ICU. This assumption is not applicable for all ICU-SDU systems, but may be reasonable in certain cases. In Section 4.7, a more extensive discussion of the results is included.

Finally, the alternative proofs that are provided in Appendices A.3 and A.4 are also worthwhile to mention. These proofs illustrate how the product-form results that are given in this report are related to other product-form results in literature. More specifically, in Appendix A.3, it is shown that the product form for a parallel overflow system with call packing can also be concluded from the product-form result in [13], which focuses on applications to stochastic Petri nets. Besides that, the alternative proof in Appendix A.4 illustrates that the product form for the ICU-SDU system that is studied in Chapter 4 can also be concluded from the product-form result in [26], which considers (alternative routing) queueing networks from a telecommunications perspective.

Further research

From both a theoretical and practical point of view, several points for further research remain of interest. For example, these include the following:

- Further study how the blocking probabilities for the overflow system with call packing compare to those for the overflow system without call packing and under which conditions on the parameters an ordering can be expected.
- Examine whether and how already existing methods to approximate blocking probabilities, which are often designed for overflow systems with a parallel structure, can also be used for overflow systems with a serial structure.
- Further examine the ICU-SDU system that is described in Chapter 4. For example, it can be interesting to study how the resulting performance measures compare to those of single-station queueing systems, such as the Erlang loss system, and to those of a real ICU-SDU system. Furthermore, it can also be useful to study the effect of the (simpifying) assumptions by using discrete-event simulation, which allows for more complicated modelling.

Appendix A

Proofs: Additional information

A.1 Proof details of Theorem 2

In Section 3.4.1, it is shown that the joint steady-state distribution of the number of jobs in the system has a product-form solution, which is given by (39) (see Theorem 2). This appendix contains further details of the proof of Theorem 2. More specifically, it is shown how the class balances in Table 5 can be verified.

Proof (cont.). Below, it is described how the class balance equations in Table 5 can be verified for an arbitrary $\mathbf{n} \in \boldsymbol{S}$.

(41.1) = (41.8): First of all, it is noted that the indicators on the left-hand side and right-hand side are the same. This means that the non-zero case ($n_{11}^1 > 0$ and $n_{12}^1 = 0$) is only left to be verified. To this end, the product form (39) is substituted and both sides are divided by $\pi(\mathbf{n})$. This yields the following for the right-hand side *rhs* and left-hand side *lhs*:

$$rhs = \frac{\pi(\mathbf{n} - e_{11}^1)}{\pi(\mathbf{n})}\lambda_1 = \frac{n_{11}^1 \mu_{11}^1}{\lambda_1}\lambda_1 = n_{11}^1 \mu_{11}^1 = lhs \tag{81}$$

Hence, since $lhs = rhs$, it follows that (41.1) = (41.8).

(41.2) = (41.9): First of all, it is noted that the same indicators are included on the left-hand side and right-hand side. This means that we only need to verify the non-zero case ($n_{11}^1 = N_1$ and $n_{12}^1 > 0$). To this end, the product form (39) is substituted and both sides are divided by $\pi(\mathbf{n})$. This leads to the following for the right-hand side *rhs* and left-hand side *lhs*:

$$rhs = \frac{\pi(\mathbf{n} - e_{12}^1)}{\pi(\mathbf{n})}\lambda_1 = \frac{N_1 \mu_{11}^1 + n_{12}^1 \mu_{12}^1}{\lambda_1}\lambda_1 = N_1 \mu_{11}^1 + n_{12}^1 \mu_{12}^1 = lhs \tag{82}$$

Hence, it follows that $lhs = rhs$, which means that (41.2) = (41.9).

(41.3) = (41.10) + (41.11): A distinction is made between two cases: $n_{11}^1 < N_1$ (Case I) and $n_{11}^1 = N_1$ (Case II).

Case I: In this case, the expression in (41.11) is equal to zero. Moreover, the same indicator remains in (41.3) and (41.10), which is $1_{\{n_{22}^1 > 0\}}$. Therefore, it must be verified that (41.3) = (41.10) if $n_{11}^1 < N_1$ and $n_{22}^1 > 0$. To this end, the product form (39) is substituted and both sides are divided by $\pi(\mathbf{n})$. This yields the following for the right-hand side rhs and left-hand side lhs:

$$\begin{aligned} rhs &= \frac{\pi(\mathbf{n}+e_{11}^1-e_{22}^1)}{\pi(\mathbf{n})} p_{1,2}^1 (n_{11}^1+1)\mu_{11}^1 \\ &= \frac{\lambda_1}{(n_{11}^1+1)\mu_{11}^1} \frac{n_{22}^1\mu_{22}^1}{p_{1,2}^1\lambda_1} p_{1,2}^1 (n_{11}^1+1)\mu_{11}^1 \\ &= n_{22}^1\mu_{22}^1 \\ &= lhs \end{aligned} \tag{83}$$

Hence, since $lhs = rhs$, we have that (41.3) = (41.10) if $n_{11}^1 < N_1$.

Case II: In this case, the expression in (41.10) is equal to zero. Moreover, the same indicator remains in (41.3) and (41.11), which is $1_{\{n_{22}^1>0\}}$. Therefore, it is only left to be verified that (41.3) = (41.11) if $n_{11}^1 = N_1$ and $n_{22}^1 > 0$. To this end, the product form (39) is substituted and both sides are divided by $\pi(\mathbf{n})$. This leads to the following for the right-hand side rhs and left-hand side lhs:

$$\begin{aligned} rhs &= \frac{\pi(\mathbf{n}+e_{12}^1-e_{22}^1)}{\pi(\mathbf{n})} p_{1,2}^1 (N_1\mu_{11}^1 + (n_{12}^1+1)\mu_{12}^1) \\ &= \frac{\lambda_1}{N_1\mu_{11}^1 + (n_{12}^1+1)\mu_{12}^1} \frac{n_{22}^1\mu_{22}^1}{p_{1,2}^1\lambda_1} p_{1,2}^1 (N_1\mu_{11}^1 + (n_{12}^1+1)\mu_{12}^1) \\ &= n_{22}^1\mu_{22}^1 \\ &= lhs \end{aligned} \tag{84}$$

Hence, since $lhs = rhs$, it follows that (41.3) = (41.11) if $n_{11}^1 = N_1$.

From this, it can thus be concluded that (41.3) = (41.10) + (41.11).

(41.4) = (41.12): First of all, it is noted that the left-hand side and right-hand side contain the same indicator. Therefore, we only need to verify the non-zero case ($n_{22}^2 > 0$). To this end, the product form (39) is substituted and both sides are divided by $\pi(\mathbf{n})$. This leads to the following for the right-hand side rhs and left-hand side lhs:

$$rhs = \frac{\pi(\mathbf{n}-e_{22}^2)}{\pi(\mathbf{n})}\lambda_2 = \frac{n_{22}^2\mu_{22}^2}{\lambda_2}\lambda_2 = n_{22}^2\mu_{22}^2 = lhs \tag{85}$$

Therefore, we have that $lhs = rhs$, which means that (41.4) = (41.12).

(41.5) + (41.6) = (41.13) + (41.14) + (41.15) + (41.16): At first, the indicators on both sides are disregarded. Then, the product form (39) is substituted and both sides are divided by $\pi(\mathbf{n})$. This yields the following for the left-hand side, consisting of lhs_5 and lhs_6:

$$lhs_5 = lhs_6 = \lambda_1 \tag{86}$$

Moreover, the following expressions are obtained for the right-hand side, consisting of rhs_{13}, rhs_{14}, rhs_{15} and rhs_{16}:

$$rhs_{13} = \frac{\pi(\mathbf{n}+e_{11}^1)}{\pi(\mathbf{n})} p_{1,0}^1 (n_{11}^1+1)\mu_{11}^1 = \frac{\lambda_1}{(n_{11}^1+1)\mu_{11}^1} p_{1,0}^1 (n_{11}^1+1)\mu_{11}^1 = p_{1,0}^1\lambda_1 \tag{87}$$

$$rhs_{14} = \frac{\pi(\mathbf{n}+e^1_{12})}{\pi(\mathbf{n})} p^1_{1,0}(N_1\mu^1_{11} + (n^1_{12}+1)\mu^1_{12}) = \frac{\lambda_1 p^1_{1,0}(N_1\mu^1_{11} + (n^1_{12}+1)\mu^1_{12})}{N_1\mu^1_{11} + (n^1_{12}+1)\mu^1_{12}} = p^1_{1,0}\lambda_1 \tag{88}$$

$$rhs_{15} = \frac{\pi(\mathbf{n}+e^1_{11})}{\pi(\mathbf{n})} p^1_{1,2}(n^1_{11}+1)\mu^1_{11} = \frac{\lambda_1}{(n^1_{11}+1)\mu^1_{11}} p^1_{1,2}(n^1_{11}+1)\mu^1_{11} = p^1_{1,2}\lambda_1 \tag{89}$$

$$rhs_{16} = \frac{\pi(\mathbf{n}+e^1_{22})}{\pi(\mathbf{n})} (n^1_{22}+1)\mu^1_{22} = \frac{p^1_{1,2}\lambda_1}{(n^1_{22}+1)\mu^1_{22}} (n^1_{22}+1)\mu^1_{22} = p^1_{1,2}\lambda_1 \tag{90}$$

Then, since $p^1_{1,0} + p^1_{1,2} = 1$, it follows that:

$$lhs_5 = rhs_{13} + rhs_{16} \tag{91}$$

$$lhs_5 = rhs_{13} + rhs_{15} \tag{92}$$

$$lhs_6 = rhs_{14} + rhs_{16} \tag{93}$$

Now, the indicators on the left-hand side and right-hand side are also taken into account. It is then verified that (41.5) + (41.6) = (41.13) + (41.14) + (41.15) + (41.16) by considering the following cases:

- Case I: $n^1_{11} < N_1$, $(n^1_{12}+n^1_{22}+1, n^2_{22}) \in \boldsymbol{C_2}$. In this case, the expressions in (41.6), (41.14) and (41.15) are equal to zero, while the indicators in the other equations simplify to one. Because of (91), the class balance is therefore verified for Case I.
- Case II: $n^1_{11} < N_1$, $(n^1_{12}+n^1_{22}+1, n^2_{22}) \notin \boldsymbol{C_2}$. In this case, the expressions in (41.6), (41.14) and (41.16) are equal to zero, while the indicators in the other equations simplify to one. From (92), the class balance is then verified for Case II.
- Case III: $n^1_{11} = N_1$, $(n^1_{12}+n^1_{22}+1, n^2_{22}) \in \boldsymbol{C_2}$. In this case, the expressions in (41.5), (41.13) and (41.15) are equal to zero, while the indicators in the other equations simplify to one. Because of (93), the class balance is therefore verified for Case III.
- Case IV: $n^1_{11} = N_1$, $(n^1_{12}+n^1_{22}+1, n^2_{22}) \notin \boldsymbol{C_2}$. In this case, the expressions in (41.5), (41.6), (41.13), (41.14), (41.15) and (41.16) are now all equal to zero. Therefore, the class balance is then immediately verified for Case IV as well.

Hence, it can be concluded that (41.5) + (41.6) = (41.13) + (41.14) + (41.15) + (41.16).

(41.7) = (41.17): First of all, it is noted that the same indicators are included on the left-hand side and right-hand side. Hence, only the non-zero case $((n^1_{12}+n^1_{22}, n^2_{22}+1) \in \boldsymbol{C_2})$ is left to be verified. To this end, the product form (39) is substituted and both sides are divided by $\pi(\mathbf{n})$. This yields the following for the right-hand side rhs and left-hand side lhs:

$$rhs = \frac{\pi(\mathbf{n}+e^2_{22})}{\pi(\mathbf{n})}(n^2_{22}+1)\mu^2_{22} = \frac{\lambda_2}{(n^2_{22}+1)\mu^2_{22}}(n^2_{22}+1)\mu^2_{22} = \lambda_2 = lhs \tag{94}$$

Hence, $lhs = rhs$, which means that (41.7) = (41.17). □

A.2 Proof details of Theorem 3

In Section 3.6.2, an insensitivity result is established. To this end, a detailed product form is first proven in Theorem 3. This is done by formulating and verifying the global balance equations. In this appendix, further details of the proof of Theorem 3 are given. More specifically, it is explained how it can be verified that (70.i) = (70.i)$'$, $i = 1, ..., 5$.

Proof (cont.). Consider an arbitrary state $\boldsymbol{R} = [\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}] \in \boldsymbol{S_d}$. Below, it is then verified that (70.i) = (70.i)$'$, $i = 1, ..., 5$.

(70.1) = (70.1)$'$: First of all, it is noted that the indicators on the left-hand side and right-hand side are the same. This means that we only need to verify the non-zero case ($n_1^1 > 0$). To this end, the detailed product form (69) is substituted and both sides are divided by $\pi_d(\boldsymbol{R})\nu_1^1$. This yields the following for the left-hand side lhs:

$$lhs = \sum_{l_1^1=1}^{n_1^1} \frac{\pi_d([\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}])}{\pi_d([\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}])\nu_1^1}\nu_1^1 = n_1^1 \tag{95}$$

Similarly, after cancelling out terms, it follows that the right-hand side rhs is equal to:

$$\begin{aligned} rhs &= \sum_{l_1^1=1}^{n_1^1} \left(\frac{\pi_d([\boldsymbol{X} - (x_{l_1^1})_{l_1^1}, \boldsymbol{Y}, \boldsymbol{Z}])}{\pi_d([\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}])\nu_1^1} \frac{1}{n_1^1} q_1^1(x_{l_1^1})\lambda_1 + \frac{\pi_d([\boldsymbol{X} - (x_{l_1^1})_{l_1^1} + (x_{l_1^1}+1)_{l_1^1}, \boldsymbol{Y}, \boldsymbol{Z}])}{\pi_d([\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}])\nu_1^1}\nu_1^1 \right) \\ &= \sum_{l_1^1=1}^{n_1^1} \left(n_1^1 \frac{\mu_1^1}{\lambda_1} \frac{1}{H_1^1(x_{l_1^1})} \frac{1}{\nu_1^1} \frac{1}{n_1^1} q_1^1(x_{l_1^1})\lambda_1 + \frac{H_1^1(x_{l_1^1}+1)}{H_1^1(x_{l_1^1})} \frac{\nu_1^1}{\nu_1^1} \right) \\ &= \sum_{l_1^1=1}^{n_1^1} \frac{q_1^1(x_{l_1^1}) \cdot (\mu_1^1/\nu_1^1) + H_1^1(x_{l_1^1}+1)}{H_1^1(x_{l_1^1})} \\ &= n_1^1 \end{aligned} \tag{96}$$

Here, the last step follows from the discrete renewal relation (60) and noting that $H_1^1(1) = \mu_1^1/\nu_1^1$. Hence, since $lhs = rhs$, it follows that (70.1) = (70.1)$'$.

(70.2) = (70.2)$'$: First of all, it is noted that the same indicator is included on the left-hand side and right-hand side. Therefore, it suffices to consider the non-zero case ($n_{22}^1 > 0$). To this end, the detailed product form (69) is substituted and both sides are divided by $\pi_d(\boldsymbol{R})\nu_{22}^1$. This yields the following for the left-hand side lhs:

$$\sum_{l_{22}^1=1}^{n_{22}^1} \frac{\pi_d([\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}])}{\pi_d([\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}])\nu_{22}^1}\nu_{22}^1 = n_{22}^1 \tag{97}$$

Similarly, after cancelling out and rearranging terms, the right-hand side rhs simplifies to:

$$\begin{aligned} rhs = \sum_{l_{22}^1=1}^{n_{22}^1} \Bigg(\sum_{l_1^1=1}^{n_1^1+1} & \frac{\pi_d([\boldsymbol{X} + (1)_{l_1^1}, \boldsymbol{Y} - (y_{l_{22}^1})_{l_{22}^1}, \boldsymbol{Z}])}{\pi_d([\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}])\nu_{22}^1} p_{1,2}^1 \frac{1}{n_{22}^1} q_{22}^1(y_{l_{22}^1})\nu_1^1 + \\ & \frac{\pi_d([\boldsymbol{X}, \boldsymbol{Y} - (y_{l_{22}^1})_{l_{22}^1} + (y_{l_{22}^1}+1)_{l_{22}^1}, \boldsymbol{Z}])}{\pi_d([\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}])\nu_{22}^1}\nu_{22}^1 \Bigg) \\ = \sum_{l_{22}^1=1}^{n_{22}^1} \Bigg(\sum_{l_1^1=1}^{n_1^1+1} & \frac{1}{n_1^1+1} \frac{\lambda_1}{\mu_1^1} H_1^1(1) n_{22}^1 \frac{\mu_{22}^1}{p_{1,2}^1\lambda_1} \frac{1}{H_{22}^1(y_{l_{22}^1})} \frac{1}{\nu_{22}^1} p_{1,2}^1 \frac{1}{n_{22}^1} q_{22}^1(y_{l_{22}^1})\nu_1^1 + \frac{H_{22}^1(y_{l_{22}^1}+1)}{H_{22}^1(y_{l_{22}^1})} \frac{\nu_{22}^1}{\nu_{22}^1} \Bigg) \end{aligned}$$

$$
\begin{aligned}
&= \sum_{l_{22}^1=1}^{n_{22}^1} \left(\frac{q_{22}^1(y_{l_{22}^1})}{H_{22}^1(y_{l_{22}^1})} \frac{\mu_{22}^1}{\nu_{22}^1} \frac{1}{n_1^1+1} \frac{\nu_1^1}{\mu_1^1} \sum_{l_1^1=1}^{n_1^1+1} H_1^1(1) + \frac{H_{22}^1(y_{l_{22}^1}+1)}{H_{22}^1(y_{l_{22}^1})} \right) \\
&= \sum_{l_{22}^1=1}^{n_{22}^1} \left(\frac{q_{22}^1(y_{l_{22}^1})}{H_{22}^1(y_{l_{22}^1})} \frac{\mu_{22}^1}{\nu_{22}^1} \frac{1}{n_1^1+1} \frac{\nu_1^1}{\mu_1^1} (n_1^1+1) \frac{\mu_1^1}{\nu_1^1} + \frac{H_{22}^1(y_{l_{22}^1}+1)}{H_{22}^1(y_{l_{22}^1})} \right) \\
&= \sum_{l_{22}^1=1}^{n_{22}^1} \frac{q_{22}^1(y_{l_{22}^1}) \cdot (\mu_{22}^1/\nu_{22}^1) + H_{22}^1(y_{l_{22}^1}+1)}{H_{22}^1(y_{l_{22}^1})} \\
&= n_{22}^1
\end{aligned} \tag{98}
$$

Here, the last step follows from the discrete renewal relation (63) and noting that $H_{22}^1(1) = \mu_{22}^1/\nu_{22}^1$. Therefore, $lhs = rhs$, from which it can be concluded that (70.2) = (70.2)′.

(70.3) = (70.3)′: First of all, it is noted that the indicators on both sides are equal. Therefore, the non-zero case ($n_{22}^2 > 0$) is only left to be verified. To this end, the detailed product form (69) is substituted and both sides are divided by $\pi_d(\boldsymbol{R})\nu_{22}^2$. The left-hand side lhs is then as follows:

$$
\sum_{l_{22}^2=1}^{n_{22}^2} \frac{\pi_d([\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z}])}{\pi_d([\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z}])\nu_{22}^2} \nu_{22}^2 = n_{22}^2 \tag{99}
$$

Similarly, after cancelling out terms, the right-hand side rhs is as follows:

$$
\begin{aligned}
rhs &= \sum_{l_{22}^2=1}^{n_{22}^2} \left(\frac{\pi_d([\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z}-(z_{l_{22}^2})_{l_{22}^2}])}{\pi_d([\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z}])\nu_{22}^2} \frac{q_{22}^2(z_{l_{22}^2})\lambda_2}{n_{22}^2} + \frac{\pi_d([\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z}-(z_{l_{22}^2})_{l_{22}^2}+(z_{l_{22}^2}+1)_{l_{22}^2}])}{\pi_d([\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z}])\nu_{22}^2} \nu_{22}^2 \right) \\
&= \sum_{l_{22}^2=1}^{n_{22}^2} \left(n_{22}^2 \frac{\mu_{22}^2}{\lambda_2} \frac{1}{H_{22}^2(z_{l_{22}^2})} \frac{1}{\nu_{22}^2} \frac{q_{22}^2(z_{l_{22}^2})\lambda_2}{n_{22}^2} + \frac{H_{22}^2(z_{l_{22}^2}+1)}{H_{22}^2(z_{l_{22}^2})} \frac{\nu_{22}^2}{\nu_{22}^2} \right) \\
&= \sum_{l_{22}^2=1}^{n_{22}^2} \frac{q_{22}^2(z_{l_{22}^2}) \cdot (\mu_{22}^2/\nu_{22}^2) + H_{22}^2(z_{l_{22}^2}+1)}{H_{22}^2(z_{l_{22}^2})} \\
&= n_{22}^2
\end{aligned} \tag{100}
$$

Here, the last step follows from the discrete renewal relation (63) and noting that $H_{22}^2(1) = \mu_{22}^2/\nu_{22}^2$. Hence, as $lhs = rhs$, it follows that (70.3) = (70.3)′.

(70.4) = (70.4)′: Since the right-hand side is divided into three parts, it must be verified that (70.4) = (70.4)′, where (70.4)′ = (70.4a)′ + (70.4b)′ + (70.4c)′. At first, the indicators on both sides are disregarded. Then, the detailed product form (69) is substituted and both sides are divided by $\pi_d(\boldsymbol{R})$. This leads to the following for the left-hand side lhs:

$$
lhs = \sum_{l_1^1=1}^{n_1^1+1} \frac{\pi_d([\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z}])}{\pi_d([\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z}])} \frac{1}{n_1^1+1} \lambda_1 = (n_1^1+1) \frac{1}{n_1^1+1} \lambda_1 = \lambda_1 \tag{101}
$$

Similarly, the right-hand side, which consists of rhs_a, rhs_b and rhs_c, is then given by:

$$
rhs_a = \sum_{l_1^1=1}^{n_1^1+1} \frac{\pi_d([\boldsymbol{X}+(1)_{l_1^1},\boldsymbol{Y},\boldsymbol{Z}])}{\pi_d([\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z}])} p_{1,0}^1 \nu_1^1 = \sum_{l_1^1=1}^{n_1^1+1} \frac{1}{n_1^1+1} \frac{\lambda_1}{\mu_1^1} H_1^1(1) p_{1,0}^1 \nu_1^1 = p_{1,0}^1 \lambda_1 \tag{102}
$$

$$rhs_b = \sum_{l_1^1=1}^{n_1^1+1} \frac{\pi_d([\boldsymbol{X}+(1)_{l_1^1}, \boldsymbol{Y}, \boldsymbol{Z}])}{\pi_d([\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}])} p_{1,2}^1 \nu_1^1 = \sum_{l_1^1=1}^{n_1^1+1} \frac{1}{n_1^1+1} \frac{\lambda_1}{\mu_1^1} H_1^1(1) p_{1,2}^1 \nu_1^1 = p_{1,2}^1 \lambda_1 \tag{103}$$

$$rhs_c = \sum_{l_{22}^1=1}^{n_{22}^1+1} \frac{\pi_d([\boldsymbol{X}, \boldsymbol{Y}+(1)_{l_{22}^1}, \boldsymbol{Z}])}{\pi_d([\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}])} \nu_{22}^1 = \sum_{l_{22}^1=1}^{n_{22}^1+1} \frac{1}{n_{22}^1+1} \frac{p_{1,2}^1 \lambda_1}{\mu_{22}^1} H_{22}^1(1) \nu_{22}^1 = p_{1,2}^1 \lambda_1 \tag{104}$$

Now, since $p_{1,0}^1 + p_{1,2}^1 = 1$, it follows that:

$$lhs = rhs_a + rhs_b \tag{105}$$

$$lhs = rhs_a + rhs_c \tag{106}$$

Next, the indicators on the left-hand side and right-hand side are also taken into account. It can then be verified that (70.4) = (70.4)′ by considering the following four cases.

- Case I: $n_1^1 < N_1$ and $(n_{22}^1+1, n_{22}^2) \in \boldsymbol{C_2}$. The expression in (70.4b)′ is equal to zero, while the indicators in the other equations simplify to one. From (106), it therefore follows that (70.4) = (70.4)′ for Case I.
- Case II: $n_1^1 < N_1$ and $(n_{22}^1+1, n_{22}^2) \notin \boldsymbol{C_2}$. The expression in (70.4c)′ is equal to zero, while the indicators in the other equations simplify to one. From (105), it therefore follows that (70.4) = (70.4)′ for Case II.
- Case III: $n_1^1 \geq N_1$ and $([n_1^1 - N_1] + n_{22}^1 + 1, n_{22}^2) \in \boldsymbol{C_2}$. The expression in (70.4b)′ is equal to zero, while the indicators in the other equations again simplify to one. From (106), it therefore follows that (70.4) = (70.4)′ for Case III.
- Case IV: $n_1^1 \geq N_1$ and $([n_1^1 - N_1] + n_{22}^1 + 1, n_{22}^2) \notin \boldsymbol{C_2}$. The expressions in (70.4), (70.4a)′, (70.4b)′ and (70.4c)′ are now all equal to zero. Hence, it can immediately be concluded that (70.4) = (70.4)′ for Case IV as well.

For all possible cases, it is thus shown that (70.4) = (70.4)′.

(70.5) = (70.5)′: First of all, it is noted that the left-hand side and right-hand side contain the same indicator. Therefore, the non-zero case $(([n_1^1 - N_1]^+ + n_{22}^1, n_{22}^2 + 1) \in \boldsymbol{C_2})$ is only left to be verified. To this end, the detailed product form (69) is substituted and both sides are divided by $\pi_d(\boldsymbol{R})$. This yields the following for left-hand side lhs:

$$lhs = \sum_{l_{22}^2=1}^{n_{22}^2+1} \frac{\pi_d([\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}])}{\pi_d([\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}])} \frac{1}{n_{22}^2+1} \lambda_2 = (n_{22}^2+1) \frac{1}{n_{22}^2+1} \lambda_2 = \lambda_2 \tag{107}$$

Similarly, after cancelling out terms, the right-hand side rhs is then as follows:

$$\begin{aligned} rhs &= \sum_{l_{22}^2=1}^{n_{22}^2+1} \frac{\pi_d([\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}+(1)_{l_{22}^2}])}{\pi_d([\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}])} \nu_{22}^2 \\ &= \sum_{l_{22}^2=1}^{n_{22}^2+1} \frac{1}{n_{22}^2+1} \frac{\lambda_2}{\mu_{22}^2} H_{22}^2(1) \nu_{22}^2 \end{aligned}$$

$$= (n_{22}^2+1)\frac{1}{n_{22}^2+1}\frac{\lambda_2}{\mu_{22}^2}\left(\frac{\mu_{22}^2}{\nu_{22}^2}\right)\nu_{22}^2$$

$$= \lambda_2 \tag{108}$$

Hence, $lhs = rhs$, from which it can be concluded that (70.5) = (70.5)′. □

A.3 Alternative proof of Theorem 1

Theorem 1 in Section 2.7.1 states that the parallel overflow system with call packing has a product-form solution for the joint steady-state distribution of the number of jobs in the system, which is given by (31). It is also mentioned that this result can be proven by modelling the parallel overflow system with call packing as competing Markov chains. The product form (31) can then be concluded from the product-form result in [13]. In [60] (pp. 29-31), this is illustrated for the case without type 2 jobs and coordinate convex structure at station 2. In this appendix, a proof is given for the case that type 2 jobs and coordinate convex structure at station 2 are included as well.

Proof. In order to describe the parallel overflow system with call packing as competing Markov chains, the following two continuous-time Markov chains are first defined (i.e. $K = 2$).

Markov chain 1: First of all, Markov chain 1 describes the transitions of type 1 jobs at station 1. This means that the state space, denoted by S_1, is given by:

$$S_1 = \{n_1 | 0 \leq n_1 \leq N_1\} \tag{109}$$

Moreover, the transition rates $q_1 = (q_1(n_1, n_1'),\ n_1, n_1' \in S_1)$ are as follows:

$$q_1(n_1, n_1') = \begin{cases} \lambda_1 & n_1' = n_1 + 1 \\ n_1\mu_1 & n_1' = n_1 - 1\ , \\ 0 & \text{else} \end{cases} \qquad n_1 \neq n_1' \tag{110}$$

$$q_1(n_1, n_1) = -\sum_{n_1' \in S_1 \setminus n_1} q(n_1, n_1') \tag{111}$$

Next, it is noted that Markov chain 1 describes a standard Erlang loss system ($M|M|N_1|N_1$ queue) with arrival rate λ_1 and mean service rate μ_1. Therefore, it immediately follows that the steady-state distribution π_1 is as follows:

$$\pi_1(n_1) = c_1 \frac{1}{n_1!}\left(\frac{\lambda_1}{\mu_1}\right)^{n_1}, \qquad n_1 \in S_1 \tag{112}$$

Here, c_1 is a normalizing constant.

Markov chain 2: Secondly, Markov chain 2 describes the transitions of both overflowed type 1 jobs at station 2 when station 1 is congested (i.e. $n_1 = N_1$) and type 2 jobs at station 2. This means that the set of admissible states, denoted by S_2, is as follows:

$$S_2 = \{(n_2, m) | (n_2, m) \in \boldsymbol{C}\} \tag{113}$$

Moreover, the transition rates $q_2 = (q_2((n_2, m), (n_2, m)'),\ (n_2, m), (n_2, m)' \in S_2)$, are given by:

$$q_2((n_2, m), (n_2, m)') = \begin{cases} \lambda_1 & (n_2, m)' = (n_2, m+1) \\ \lambda_2 & (n_2, m)' = (n_2+1, m) \\ N_1\mu_1 + m\gamma & (n_2, m)' = (n_2, m-1) \\ n_2\mu_2 & (n_2, m)' = (n_2-1, m) \\ 0 & \text{else} \end{cases}, \qquad (n_2, m) \neq (n_2, m)' \tag{114}$$

$$q_2((n_2, m), (n_2, m)) = -\sum_{(n_2,m)' \in S_2 \setminus (n_2,m)} q_2((n_2, m), (n_2, m)') \tag{115}$$

Here, it is noted that the term $N_1\mu_1$ is included, since an overflowed type 1 job at station 2 also leaves if a job at station 1 completes its service (because of call packing).

Then, with c_2 a normalizing constant, the following steady-state distribution π_2 results:

$$\pi_2(n_2, m) = \begin{cases} c_2 \frac{1}{n_2!} \left(\frac{\lambda_2}{\mu_2}\right)^{n_2} & \text{if } m = 0 \\ c_2 \frac{1}{n_2!} \left(\frac{\lambda_2}{\mu_2}\right)^{n_2} \frac{(\lambda_1)^m}{\prod_{k=1}^{m}(N_1\mu_1 + k\gamma)} & \text{if } m > 0 \end{cases}, \qquad (n_2, m) \in S_2 \tag{116}$$

This can be shown by verifying the following global balance equations by substitution of the expression for π_2 in (116) (note that $(117.i) = (117.i)'$, $i = 1, ..., 4$):

$$\begin{aligned} &\left\{\begin{array}{l} \pi_2(n_2, m)(N_1\mu_1 + m\gamma)1_{\{m>0\}} + \\ \pi_2(n_2, m)n_2\mu_2 1_{\{n_2>0\}} + \\ \pi_2(n_2, m)\lambda_1 1_{\{(n_2, m+1) \in \boldsymbol{C}\}} + \\ \pi_2(n_2, m)\lambda_2 1_{\{(n_2+1, m) \in \boldsymbol{C}\}} \end{array}\right\} && \begin{array}{l} (117.1) \\ (117.2) \\ (117.3) \\ (117.4) \end{array} \\ &= && (117) \\ &\left\{\begin{array}{l} \pi_2(n_2, m-1)\lambda_1 1_{\{m>0\}} + \\ \pi_2(n_2-1, m)\lambda_2 1_{\{n_2>0\}} + \\ \pi_2(n_2, m+1)(N_1\mu_1 + (m+1)\gamma)1_{\{(n_2, m+1) \in \boldsymbol{C}\}} + \\ \pi_2(n_2+1, m)(n_2+1)\mu_2 1_{\{(n_2+1, m) \in \boldsymbol{C}\}} \end{array}\right\} && \begin{array}{l} (117.1)' \\ (117.2)' \\ (117.3)' \\ (117.4)' \end{array} \end{aligned}$$

Next, as becomes apparent later on, it is necessary to distinguish between the transitions of overflowed type 1 jobs and type 2 jobs at station 2. Therefore, the transition rates for Markov chain 2 are separated into two parts (i.e. $R_2 = 2$). These are a part that describes the behaviour of overflowed type 1 jobs (denoted by $q_2^{(1)}$) and a part that describes the behaviour of type 2 jobs (denoted by $q_2^{(2)}$). Hence, for $(n_2, m), (n_2, m)' \in S_2$, the transition rates $q_2^{(1)}$ and $q_2^{(2)}$ are as follows:

$$q_2^{(1)}((n_2, m), (n_2, m)') = \begin{cases} \lambda_1 & (n_2, m)' = (n_2, m+1) \\ N_1\mu_1 + m\gamma & (n_2, m)' = (n_2, m-1) \\ 0 & \text{else} \end{cases}, \qquad (n_2, m) \neq (n_2, m)' \tag{118}$$

$$q_2^{(1)}((n_2, m), (n_2, m)) = -\sum_{(n_2,m)' \in S_2 \setminus (n_2,m)} q_2^{(1)}((n_2, m), (n_2, m)') \tag{119}$$

$$q_2^{(2)}((n_2,m),(n_2,m)') = \begin{cases} \lambda_2 & (n_2,m)' = (n_2+1,m) \\ n_2\mu_2 & (n_2,m)' = (n_2-1,m)\ , \qquad (n_2,m) \neq (n_2,m)' \\ 0 & \text{else} \end{cases} \tag{120}$$

$$q_2^{(2)}((n_2,m),(n_2,m)) = -\sum_{(n_2,m)' \in S_2 \setminus (n_2,m)} q_2^{(2)}((n_2,m),(n_2,m)') \tag{121}$$

Hence, $q_2^{(1)}$ and $q_2^{(2)}$ are defined such that, for all $(n_2,m),(n_2,m)' \in S_2$, the following holds:

$$q_2((n_2,m),(n_2,m)') = q_2^{(1)}((n_2,m),(n_2,m)') + q_2^{(2)}((n_2,m),(n_2,m)') \tag{122}$$

Moreover, it can be seen that Markov chain 2 is locally balanced with respect to the separation (122). Here, Markov chain 2 is said to be locally balanced with respect to this separation if the following is satisfied by the steady-state distribution π_2 for $r = 1,2$ (see [13], p. 539):

$$\sum_{(n_2,m)' \in S_2} \left\{ \pi((n_2,m)) q_2^{(r)}((n_2,m),(n_2,m)') - \pi((n_2,m)') q_2^{(r)}((n_2,m)',(n_2,m)) \right\} = 0 \tag{123}$$

It can then be seen from the global balance equations (117) that this condition holds for $r = 1,2$ by noting that (117.1) + (117.3) = (117.1)′ + (117.3)′ and (117.2) + (117.4) = (117.2)′ + (117.4)′.

The processes with transition rates $q_2^{(1)}$ and $q_2^{(2)}$ are then Markov chains on their own. These Markov chains are referred to as Markov chain (2,1) for overflowed type 1 jobs at station 2 and Markov chain (2,2) for type 2 jobs at station 2. It is noted that these Markov chains operate on the same state space, which is S_2.

In contrast, the transition rates of Markov chain 1 are not separated into multiple Markov chains. For notational purposes, however, this is seen as if the transition rates of Markov chain 1 are 'separated' into one part (i.e. $R_1 = 1$). The resulting chain is then referred to as Markov chain (1,1) and its transition rates are denoted by $q_1^{(1)}$, where:

$$q_1^{(1)}(n_1,n_1') = q_1(n_1,n_1'), \qquad n_1,n_1' \in S_1 \tag{124}$$

Hence, three Markov chains are obtained:

- Markov chain (1,1) describing the transitions of type 1 jobs at station 1.
- Markov chain (2,1) describing the transitions of overflowed type 1 jobs at station 2.
- Markov chain (2,2) describing the transitions of type 2 jobs at station 2.

However, it can be noted that the parallel overflow system with call packing is not accurately described by these Markov chains yet. More specifically, if $n_1 < N_1$ (and hence $m = 0$), there should be no arrivals of overflowed type 1 jobs at station 2, since arriving type 1 jobs would go to station 1. Moreover, if $m > 0$ (and hence $n_1 = N_1$), there should be no departures from station 1, since an overflowed type 1 job from station 2 would immediately take the place of a departing job at station 1 (because of call packing). Therefore, the competition mechanism is introduced.

To this end, define the index set $I = \{1,2\}$, and let $A_{(k,r),i}$ and $C_{(k,r),i}$, $k = 1,2$, $r = 1,...,R_k$ and $i \in I$, be as follows (see [13] for the precise interpretations):

$$A_{(1,1),1} = \{n_1 | n_1 = N_1\}; \qquad C_{(1,1),1} = \emptyset \tag{125}$$

$$A_{(1,1),2} = \{n_1 | 0 \leq n_1 < N_1\}; \qquad C_{(1,1),2} = \{(2,1)\} \tag{126}$$

$$A_{(2,1),1} = \{(n_2,m) | m = 0, (n_2,0) \in \boldsymbol{C}\}; \qquad C_{(2,1),1} = \emptyset \tag{127}$$

$$A_{(2,1),2} = \{(n_2,m) | m > 0, (n_2,m) \in \boldsymbol{C}\}; \qquad C_{(2,1),2} = \{(1,1)\} \tag{128}$$

$$A_{(2,2),1} = \{(n_2,m) | m = 0, (n_2,0) \in \boldsymbol{C}\}; \qquad C_{(2,2),1} = \emptyset \tag{129}$$

$$A_{(2,2),2} = \{(n_2,m) | m > 0, (n_2,m) \in \boldsymbol{C}\}; \qquad C_{(2,2),2} = \emptyset \tag{130}$$

Hence, Markov chains (1,1) and (2,1) compete over resource 2, while they do not compete over resource 1. Moreover, Markov chain (2,2) does not compete over resources at all. More specifically, all Markov chains can make a transition if $n_1 \in A_{(1,1),1}$ and $(n_2, m) \in A_{(2,1),1}$ (then, none of the Markov chains are frozen). On the other hand, only Markov chains (1,1) and (2,2) are allowed to make a transition if $n_1 \in A_{(1,1),2}$ and $(n_2, m) \in A_{(2,1),1}$ (then, Markov chain (2,1) is frozen). Similarly, only Markov chains (2,1) and (2,2) can make a transition if $n_1 \in A_{(1,1),1}$ and $(n_2, m) \in A_{(2,1),2}$ (then, Markov chain (1,1) is frozen). Finally, it is not possible to be in a state (n_1, n_2, m) with $n_1 \in A_{(1,1),2}$ and $(n_2, m) \in A_{(2,1),2}$.

Here, it is noted that Markov chain (2,2) does not compete over resources, which means that it is never frozen (i.e. it can always make a transition independent of the state). As a consequence, if $n_1 < N_1$, Markov chain (2,2) can still undergo transitions, while Markov chain (2,1) is frozen. Hence, type 2 jobs can still arrive at and leave from station 2, while arrivals of overflowed type 1 jobs cannot occur, which is as desired. Besides that, note that, as $C_{(2,2),1} = C_{(2,2),2} = \emptyset$, there are multiple possibilities to choose $A_{(2,2),1}$ and $A_{(2,2),2}$.

As mentioned before, because of the competition mechanism, a state (n_1, n_2, m) with both $n_1 \in A_{(1,1),2}$ and $(n_2, m) \in A_{(2,1),2}$ cannot occur. Therefore, the state space S is as follows:

$$\begin{aligned} S &= S_1 \times S_2 \backslash A_{(1,1),2} \times A_{(2,1),2} \\ &= \{(n_1, n_2, m) \mid 0 \leq n_1 < N_1,\ m = 0,\ (n_2, 0) \in \boldsymbol{C} \ \text{ or } \ n_1 = N_1,\ (n_2, m) \in \boldsymbol{C}\} \end{aligned} \tag{131}$$

Subsequently, the coefficients $c_1(n_1)$ and $c_2(n_2, m)$ are chosen equal to one for all $n_1 \in S_1$ and $(n_2, m) \in S_2$, respectively. The transition rates $q = (q(\bar{n}, \bar{n}'),\ \bar{n}, \bar{n}' \in S)$, where $\bar{n} = (n_1, (n_2, m))$ and $\bar{n}' = (n_1', (n_2, m)')$, are then given by:

$$\begin{aligned} q(\bar{n}, \bar{n}') = q_1^{(1)}(n_1, n_1') 1_{\{(n_2,m)=(n_2,m)' \in A_{(2,1),1}\}} + \\ q_2^{(1)}((n_2, m), (n_2, m)') 1_{\{n_1 = n_1' \in A_{(1,1),1}\}} + q_2^{(2)}((n_2, m), (n_2, m)') 1_{\{n_1 = n_1'\}} \end{aligned} \tag{132}$$

Here, $q_1^{(1)}$, $q_2^{(1)}$ and $q_2^{(2)}$ are as defined in (124), (118) and (120), respectively.

Summarizing, Markov chain 1 and Markov chain 2 are defined, after which the transition rates of Markov chain 2 are separated into two parts, as in (122). Moreover, Markov chain 2 is locally balanced with respect to this separation. Subsequently, competition between the Markov chains is introduced. This results in the Markov chain at state space S as given in (131) and with transition rates q as defined in (132). This Markov chain accurately describes the behaviour of the parallel overflow system with call packing.

Moreover, the conditions to apply Theorem 2 in [13] (p. 539) are satisfied. Therefore, it follows that the steady-state distribution π_{cp} is as follows:

$$\pi_{cp}(n_1, n_2, m) = B\pi_1(n_1)\pi_2(n_2, m), \qquad (n_1, n_2, m) \in S \tag{133}$$

Here, B is the normalizing constant, which is determined by the competition mechanism.

Next, (112) and (116) are substituted for π_1 and π_2, respectively. Then, with c_{cp} the normalizing constant, the following product form results:

$$\pi_{cp}(n_1, n_2, m) = \begin{cases} c_{cp}\frac{1}{n_1!}\left(\frac{\lambda_1}{\mu_1}\right)^{n_1}\frac{1}{n_2!}\left(\frac{\lambda_2}{\mu_2}\right)^{n_2} & \text{if } m = 0 \\ c_{cp}\frac{1}{n_1!}\left(\frac{\lambda_1}{\mu_1}\right)^{n_1}\frac{1}{n_2!}\left(\frac{\lambda_2}{\mu_2}\right)^{n_2}\frac{(\lambda_1)^m}{\prod_{k=1}^{m}(N_1\mu_1+k\gamma)} & \text{if } m > 0 \end{cases}, \qquad (n_1, n_2, m) \in S \tag{134}$$

It can be noted that this expression is equal to the product form (31). Hence, this completes the (alternative) proof of Theorem 1. □

A.4 Alternative proof of Theorem 4

In Section 4.5, the ICU-SDU system that is subject of Chapter 4 is shown to exhibit a product-form solution for the steady-state distribution (see (76) in Theorem 4). To this end, it is verified that the global balance equations are solved by the product form (76). As another approach, the product form (76) can also be concluded from the product-form result in [26]. In this appendix, this is further explained.

Proof. In order to prove the result, it is shown that the ICU-SDU system as described in Section 4.4 satisfies conditions (a) to (g) as stated in [26] and also given below. It is noted, though, that the formulation below is slightly adapted by changing “calls” to “patients” and “type” to “class” (as type is already used in a different context in this report).

First of all, the patients are categorized into three classes: type 1 patients who require ICU care (class 1a), type 1 patients who require SDU care (class 1b) and type 2 patients who require SDU care (class 2). Then, some notation is introduced.

Let $T = \{1\text{a}, 1\text{b}, 2\}$, and $\mathbf{n} = \{n(t), t \in T\} = \{n(1\text{a}), n(1\text{b}), n(2)\}$, where:

- $n(1\text{a})$: The number of present type 1 patients who require ICU care (i.e. $n(1\text{a}) = n^1_{11} + n^1_{12}$).
- $n(1\text{b})$: The number of present type 1 patients who require SDU care (i.e. $n(1\text{b}) = n^1_2$).
- $n(2)$: The number of present type 2 patients who require SDU care (i.e. $n(2) = n^2_2$).

Hence, the state vector $\mathbf{n}$ now only contains the total number of present type 1 patients who require ICU care, which is denoted by $n(1\text{a})$, instead of the number of non-overflowed type 1 patients at the ICU (i.e. n^1_{11}) and overflowed type 1 patients at the SDU (i.e. n^1_{12}) separately. However, n^1_{11} and n^1_{12} can be obtained from $n(1\text{a})$ by noting that $n^1_{11} = \min\{n(1\text{a}), N_1\}$ and $n^1_{12} = \max\{n(1\text{a}) - N_1, 0\}$.

Next, $\mathbf{e(t)}$ denotes the state that consists of exactly one patient of class t, $t \in T$. Moreover, the arrival rate of class t patients, which is denoted by $\lambda(t)$ $(t \in T)$, is as follows:

$$\lambda(1\text{a}) = \lambda_1; \quad \lambda(1\text{b}) = 0; \quad \lambda(2) = \lambda_2 \tag{135}$$

Subsequently, the set of feasible states $\mathcal{F}$ is given by $\boldsymbol{S}$, where $\boldsymbol{S}$ is as specified by (75). Using the notation that is introduced in this appendix, this means that $\mathcal{F}$ is as follows:

$$\begin{aligned}\mathcal{F} = \{\mathbf{n} \mid\ & 0 \leq n(1\text{a}) \leq N_1,\ 0 \leq n(1\text{b}) + n(2) \leq N_2 \ \text{ or} \\ & n(1\text{a}) > N_1,\ 0 \leq [n(1\text{a}) - N_1] + n(1\text{b}) + n(2) \leq N_2 \ \text{ or} \\ & 0 \leq n(1\text{a}) + [n(1\text{b}) + n(2) - N_2] \leq N_1,\ n(1\text{b}) + n(2) > N_2\}\end{aligned} \tag{136}$$

Afterwards, the following non-negative function $\phi(\mathbf{n})$, $\mathbf{n} \in \mathcal{F}$, is defined:

$$\phi(\mathbf{n}) = \begin{cases} \frac{1}{n(1\text{a})!}\left(\frac{1}{\mu_{11}^1}\right)^{n(1\text{a})} \frac{1}{n(1\text{b})!}\left(\frac{1}{\mu_2^1}\right)^{n(1\text{b})} \frac{1}{n(2)!}\left(\frac{1}{\mu_2^2}\right)^{n(2)} & \text{for } n(1\text{a}) \leq N_1 \\ \frac{1}{N_1!}\left(\frac{1}{\mu_{11}^1}\right)^{N_1} \frac{1}{\prod_{k=1}^{n(1\text{a})-N_1}(N_1\mu_{11}^1 + k\mu_{12}^1)} \frac{1}{n(1\text{b})!}\left(\frac{1}{\mu_2^1}\right)^{n(1\text{b})} \frac{1}{n(2)!}\left(\frac{1}{\mu_2^2}\right)^{n(2)} & \text{for } n(1\text{a}) > N_1 \end{cases} \tag{137}$$

Finally, in order to describe the ICU-SDU system, $r(t)$, $p(t,s)$, $\delta_t(l,\mathbf{n})$ and $\gamma_t(l,\mathbf{n})$ are defined as follows (see below for the interpretations of these functions):

$$r(1\text{a}) = p_{1,0}^1 \tag{138}$$

$$r(t) = 1 \qquad t \in \{1\text{b}, 2\} \tag{139}$$

$$p(1\text{a}, 1\text{b}) = p_{1,2}^1 \tag{140}$$

$$p(t,s) = 0 \qquad t \in \{1\text{a}, 1\text{b}, 2\},\ \ s \in \{1\text{a}, 1\text{b}, 2\}\ (\text{except } t = 1\text{a},\ \ s = 1\text{b}) \tag{141}$$

$$\delta_{1\text{a}}(l,\mathbf{n}) = 1/n(1\text{a}) \qquad \text{for } l = 1, ..., n(1\text{a}) \qquad \mathbf{n} \in \mathcal{F} \text{ with } n(1\text{a}) \leq N_1 \tag{142}$$

$$\delta_{1\text{a}}(l,\mathbf{n}) = \begin{cases} 0 & \text{for } l = 1, ..., , n(1\text{a}) - 1 \\ 1 & \text{for } l = n(1\text{a}) \end{cases} \qquad \mathbf{n} \in \mathcal{F} \text{ with } n(1\text{a}) > N_1 \tag{143}$$

$$\delta_{1\text{b}}(l,\mathbf{n}) = 1/n(1\text{b}) \qquad \text{for } l = 1, ..., n(1\text{b}) \qquad \mathbf{n} \in \mathcal{F} \tag{144}$$

$$\delta_2(l,\mathbf{n}) = 1/n(2) \qquad \text{for } l = 1, ..., n(2) \qquad \mathbf{n} \in \mathcal{F} \tag{145}$$

$$\gamma_{1\text{a}}(l,\mathbf{n}) = 1/n(1\text{a}) \qquad \text{for } l = 1, ..., n(1\text{a}) \qquad \mathbf{n} \in \mathcal{F} \text{ with } n(1\text{a}) \leq N_1 \tag{146}$$

$$\gamma_{1\text{a}}(l,\mathbf{n}) = \begin{cases} \frac{\mu_{11}^1}{N_1\mu_{11}^1 + (n(1\text{a}) - N_1)\mu_{12}^1} & \text{for } l = 1, ..., N_1 \\ \frac{\mu_{12}^1}{N_1\mu_{11}^1 + (n(1\text{a}) - N_1)\mu_{12}^1} & \text{for } l = N_1 + 1, ..., n(1\text{a}) \end{cases} \qquad \mathbf{n} \in \mathcal{F} \text{ with } n(1\text{a}) > N_1 \tag{147}$$

$$\gamma_{1\text{b}}(l,\mathbf{n}) = 1/n(1\text{b}) \qquad \text{for } l = 1, ..., n(1\text{b}) \qquad \mathbf{n} \in \mathcal{F} \tag{148}$$

$$\gamma_2(l,\mathbf{n}) = 1/n(2) \qquad \text{for } l = 1, ..., n(2) \qquad \mathbf{n} \in \mathcal{F} \tag{149}$$

Now, conditions (a) to (g) as given in [26] (pp. 5.1B.2.1-5.1B.2.2) can be verified. This can be done as follows.

(a) Patients of class t arrive in a Poisson stream with parameter $\lambda(t)$.

By (135) and Assumption A2, it immediately follows that condition (a) is satisfied.

(b) For all $\mathbf{n}$ *with* $n(t) > 0$, *the service facility serves patients of class t at a total rate given by:*

$$c(t,\mathbf{n}) = \frac{\phi(\mathbf{n} - \mathbf{e(t)})}{\phi(\mathbf{n})} \tag{150}$$

with ϕ any arbitrary non-negative function.

Let the function ϕ be as in (137). Then, $c(t, \mathbf{n})$ is as follows for all $\mathbf{n}$ with $n(t) > 0$ and $t \in T$:

$$c(1\text{a}, \mathbf{n}) = \begin{cases} n(1\text{a})\mu_{11}^1 & \text{if } n(1\text{a}) \leq N_1 \\ N_1\mu_{11}^1 + (n(1\text{a}) - N_1)\mu_{12}^1 & \text{if } n(1\text{a}) > N_1 \end{cases} \tag{151}$$

$$c(1\text{b}, \mathbf{n}) = n(1\text{b})\mu_2^1 \tag{152}$$

$$c(2, \mathbf{n}) = n(2)\mu_2^2 \tag{153}$$

Hence, it can be concluded that condition (b) is satisfied.

(c) *A patient of class t who finishes service departs the network with probability $r(t)$ or changes into a patient of class s with probability $p(t, s)$ where $r(t) + \sum_{s \in T} p(t, s) = 1$ for all $t \in T$.*

Let $r(t)$, $t \in T$, and $p(t, s)$, $t, s \in T$, be as defined in (138), (139), (140) and (141). Then, by Assumption A3, it immediately follows that condition (c) is satisfied.

(d) *A patient of class t who arrives to find the network in state $\mathbf{n} - \mathbf{e(t)}$ is allocated label l amongst the class t patients with probability $\delta_t(l, \mathbf{n})$. When this happens, patients with labels $l, l + 1, ..., n(t) - 1$ are relabelled $l + 1, l + 2, ..., n(t)$.*

First of all, let $\delta_{1\text{a}}(l, \mathbf{n})$ be as defined in (142) and (143). Then, by (142), an arriving class 1a patient is randomly allocated a label l amongst the class 1a patients if $n(1\text{a}) \leq N_1$ (i.e. there are no overflowed patients at the SDU). Moreover, by (143), if $n(1\text{a}) > N_1$ (i.e. there are overflowed patients at the SDU), an arriving class 1a patient is allocated label $n(1\text{a})$ amongst the class 1a patients (see also condition (e) below). It can thus be noted that a distinction is made between non-overflowed class 1a patients, who have a label l that is smaller than or equal to N_1, and overflowed class 1a patients, who have a label l that is greater than N_1. This is necessary, since it is allowed that these patients have a different service rate (i.e. $\mu_{11}^1 \neq \mu_{12}^1$). Next, let $\delta_{1\text{b}}(l, \mathbf{n})$ be as defined in (144). This means that arriving class 1b patients are randomly allocated a label l amongst the class 1b patients. This is possible, since the service rates of non-overflowed and overflowed class 1b patients are assumed to be equal. As a consequence, it is not necessary to distinguish between these patients (see also Section 4.5). Finally, let $\delta_2(l, \mathbf{n})$ be as in (145). This means that arriving class 2 patients are randomly allocated a label l amongst the class 2 patients. Similar as for class 1b patients, this can be done, because non-overflowed and overflowed class 2 patients are assumed to have an equal service rate. Hence, it follows that condition (d) is satisfied.

(e) *The proportion of service effort that is given to the lth patient of class t when the state is $\mathbf{n}$ is $\gamma_t(l, \mathbf{n})$. When a patient with label l departs, patients with labels $l + 1, l + 2, ..., n(t)$ are relabelled $l, l + 1, ..., n(t) - 1$.*

First of all, let $\gamma_{1\text{a}}(l, \mathbf{n})$ be as defined in (146) and (147). In that case, the service rate of a non-overflowed class 1a patient, who, by design, has a label l that is smaller than or equal to N_1 (see also condition (d) above), is μ_{11}^1. Moreover, the service rate of an overflowed class 1a patient, who has a label that is greater than N_1 (see again condition (d) above), is equal to μ_{12}^1. Subsequently, let $\gamma_{1\text{b}}(l, \mathbf{n})$ and $\gamma_2(l, \mathbf{n})$ be as defined in (148) and (149), respectively.

Then, class 1b patients are served with rate μ_2^1, while the service rate of class 2 patients is equal to μ_2^2. The first part of condition (e) is therefore satisfied. In order to verify the second part of condition (e), it is important to pay attention to the situation that a class 1a patient with label $N_1 + 1$ is relabelled N_1. More specifically, if $n(1\text{a}) > N_1$ and a non-overflowed class 1a patient with a label l that is smaller than or equal to N_1 departs, the class 1a patient with label $N_1 + 1$ is relabelled N_1. This can be interpreted as that this overflowed class 1a patient goes from the SDU to the ICU. Moreover, because of (143), this patient is present for a longer time than the other overflowed class 1a patients (i.e. FIFO). Hence, assuming FIFO, the second part of condition (e) is also satisfied. Moreover, it is noted that, instead of FIFO, several other assumptions (e.g. LIFO or random) could also be made by defining $\delta_{1\text{a}}(l, \mathbf{n})$ in (143) in a different way.

(f) If class t patients have a non-exponential service time distribution, then $\delta_t(l, \mathbf{n}) = \gamma_t(l, \mathbf{n})$.
Since the service times are assumed to be exponentially distributed (see Assumption A7), condition (f) is satisfied. On the other hand, because $\delta_{1\text{a}}(l, \mathbf{n}) \neq \gamma_{1\text{a}}(l, \mathbf{n})$, condition (f) would not be satisfied if the service time distribution is assumed to be non-exponential. It is noted, though, that this is not the case if the service rates of non-overflowed and overflowed class 1a patients are assumed to be equal (i.e. $\mu_{11}^1 = \mu_{12}^1$). More specifically, similar as for class 1b and class 2 patients, $\delta_{1\text{a}}(l, \mathbf{n})$ can then be chosen equal to $1/n(1\text{a})$ for all $l = 1, ..., n(1\text{a})$, even if $n(1\text{a}) > N_1$. Moreover, if $\mu_{11}^1 = \mu_{12}^1$, $\gamma_{1\text{a}}(l, \mathbf{n})$ simplifies to $1/n(1\text{a})$ for all $l = 1, ..., n(1\text{a})$. Therefore, it follows that $\delta_{1\text{a}}(l, \mathbf{n}) = \gamma_{1\text{a}}(l, \mathbf{n})$, which means that condition (f) is also satisfied for non-exponential service times if $\mu_{11}^1 = \mu_{12}^1$.

(g) For all $t \in T$ and $\mathbf{n} + \mathbf{e}(\mathbf{t}) \in \mathcal{F}$, if $\lambda(t) > 0$ or $r(t) > 0$, then $\mathbf{n} \in \mathcal{F}$, and if $p(s, t) > 0$ or $p(t, s) > 0$, then $\mathbf{n} + \mathbf{e}(\mathbf{s}) \in \mathcal{F}$.
Since $r(t) > 0$ for $t \in \{1\text{b}, 2\}$, $r(1\text{a}) = p_{1,0}^1$ and $p(1\text{a}, 1\text{b}) = p_{1,2}^1$, where $p_{1,0}^1$ and $p_{1,2}^1$ may be larger than zero, it follows that the following conditions must hold:

$$\mathbf{n} + \mathbf{e}(\mathbf{1a}) \in \mathcal{F} \Rightarrow \mathbf{n} \in \mathcal{F} \tag{154}$$

$$\mathbf{n} + \mathbf{e}(\mathbf{1b}) \in \mathcal{F} \Rightarrow \mathbf{n} \in \mathcal{F} \tag{155}$$

$$\mathbf{n} + \mathbf{e}(\mathbf{2}) \in \mathcal{F} \Rightarrow \mathbf{n} \in \mathcal{F} \tag{156}$$

$$\mathbf{n} + \mathbf{e}(\mathbf{1a}) \in \mathcal{F} \Rightarrow \mathbf{n} + \mathbf{e}(\mathbf{1b}) \in \mathcal{F} \tag{157}$$

$$\mathbf{n} + \mathbf{e}(\mathbf{1b}) \in \mathcal{F} \Rightarrow \mathbf{n} + \mathbf{e}(\mathbf{1a}) \in \mathcal{F} \tag{158}$$

Now, from (136), it can be readily seen that these conditions are fulfilled. Hence, it can be concluded that condition (g) is satisfied.

It is thus verified that conditions (a) to (g) are satisfied by the ICU-SDU system as introduced in Section 4.4. Hence, it can be concluded from Theorem 2 in [26] (p. 5.1B.2.2) that the steady-state distribution $\pi(\mathbf{n})$, $\mathbf{n} \in \mathcal{F}$, has the following solution:

$$\pi(\mathbf{n}) = c\phi(\mathbf{n}) \prod_{t \in T} [y(t)]^{n(t)} \tag{159}$$

Here, c is the normalizing constant and the $y(t)$ satisfy:

$$y(t) = \lambda(t) + \sum_{s \in T} y(s) p(s,t) \qquad \text{for } t \in T \tag{160}$$

It can therefore be noted that the $y(t)$, $t \in T$, are as follows:

$$y(1\text{a}) = \lambda(1\text{a}) + 0 = \lambda_1; \quad y(1\text{b}) = 0 + y(1\text{a})p(1\text{a}, 1\text{b}) = p^1_{1,2}\lambda_1; \quad y(2) = \lambda(2) + 0 = \lambda_2 \tag{161}$$

Consequently, the following expression for $\pi(\mathbf{n})$, $\mathbf{n} \in \mathcal{F}$, is obtained:

$$\pi(\mathbf{n}) = \begin{cases} \frac{1}{n(1\text{a})!} \left(\frac{\lambda_1}{\mu^1_{11}}\right)^{n(1\text{a})} \frac{1}{n(1\text{b})!} \left(\frac{p^1_{1,2}\lambda_1}{\mu^1_2}\right)^{n(1\text{b})} \frac{1}{n(2)!} \left(\frac{\lambda_2}{\mu^2_2}\right)^{n(2)} & n(1\text{a}) \leq N_1 \\ \frac{1}{N_1!} \left(\frac{\lambda_1}{\mu^1_{11}}\right)^{N_1} \frac{(\lambda_1)^{n(1)-N_1}}{\prod_{k=1}^{n(1\text{a})-N_1} (N_1\mu^1_{11} + k\mu^1_{12})} \frac{1}{n(1\text{b})!} \left(\frac{p^1_{1,2}\lambda_1}{\mu^1_2}\right)^{n(1\text{b})} \frac{1}{n(2)!} \left(\frac{\lambda_2}{\mu^2_2}\right)^{n(2)} & n(1\text{a}) > N_1 \end{cases} \tag{162}$$

Next, since $n^1_{11} = \min\{n(1\text{a}), N_1\}$, $n^1_{12} = \max\{n(1\text{a}) - N_1, 0\}$, $n^1_2 = n(1\text{b})$ and $n^2_2 = n(2)$, the following expression for $\pi(n^1_{11}, n^1_{12}, n^1_2, n^2_2)$, $(n^1_{11}, n^1_{12}, n^1_2, n^2_2) \in \boldsymbol{S}$, can be obtained:

$$\pi(n^1_{11}, n^1_{12}, n^1_2, n^2_2) = \begin{cases} \frac{1}{n^1_{11}!} \left(\frac{\lambda_1}{\mu^1_{11}}\right)^{n^1_{11}} \frac{1}{n^1_2!} \left(\frac{p^1_{1,2}\lambda_1}{\mu^1_2}\right)^{n^1_2} \frac{1}{n^2_2!} \left(\frac{\lambda_2}{\mu^2_2}\right)^{n^2_2} & \text{if } n^1_{12} = 0 \\ \frac{1}{N_1!} \left(\frac{\lambda_1}{\mu^1_{11}}\right)^{N_1} \frac{(\lambda_1)^{n^1_{12}}}{\prod_{k=1}^{n^1_{12}} (N_1\mu^1_{11} + k\mu^1_{12})} \frac{1}{n^1_2!} \left(\frac{p^1_{1,2}\lambda_1}{\mu^1_2}\right)^{n^1_2} \frac{1}{n^2_2!} \left(\frac{\lambda_2}{\mu^2_2}\right)^{n^2_2} & \text{if } n^1_{12} > 0 \end{cases} \tag{163}$$

This expression is equivalent to the product-form solution for the steady-state distribution in (76). Hence, this completes the (alternative) proof of Theorem 4. □

Remark 19 (Overflow system in Chapter 3). It is noted that the overflow system with call packing that is studied in Chapter 3 does not satisfy conditions (a) to (g), since condition (g) is not fulfilled. This can be illustrated by the following example. Let $N_1 = N_2 = M^1_2 = M^2_2 = 5$ and consider the admissible state $(3, 0, 1, 4)$, that is, there are three type 1 jobs at station 1, no overflowed type 1 jobs at station 2, one non-overflowed type 1 job at station 2 and four type 2 jobs at station 2 present. Then, if $p^1_{1,2} > 0$, condition (g) would require that $(2, 0, 2, 4)$ (i.e. the same state with one less type 1 job at station 1 and one more non-overflowed type 1 job at station 2) is also an admissible state. However, as $(2, 4) \notin \boldsymbol{C_2}$, this is not the case, which means that condition (g) is not satisfied. This means that the product form (39) cannot be concluded from Theorem 2 in [26]. It is noted, though, that the product form may be concluded from Theorem 1 in [26] if it is possible to model the required restrictions on the state space via the functions that are mentioned in conditions (a) to (f) instead of using condition (g). However, this is not fully worked out.

Appendix B

Blocking probabilities: Additional information

In Chapter 3, an overflow system with serial structure is studied, where Section 3.5 focuses on the blocking probabilities for this system. This appendix provides some additional information regarding these blocking probabilities. First of all, in Appendix B.1, details on the calculation of the blocking probabilities are given. Secondly, Appendix B.2 contains some numerical results.

B.1 Calculation of the blocking probabilities

In Section 3.5.2, it is briefly described how the blocking probabilities for the overflow system that is studied in Chapter 3 can be determined from the steady-state distributions. This appendix provides additional details on these calculations. First of all, in Appendix B.1.1, it is described how the blocking probabilities for the overflow system with call packing can be calculated. Next, the computations of the blocking probabilities for the overflow system without call packing are discussed in Appendix B.1.2.

B.1.1 Serial overflow system with call packing

Let π be the steady-state distribution of the number of jobs in the serial overflow system with call packing as given in (39). Because of the PASTA property of Poisson arrivals, the blocking probabilities b_1, B_1, O_1 and B_2 can then be calculated by summing $\pi(\mathbf{n})$ over the appropriate states $\mathbf{n} = (n^1_{11}, n^1_{12}, n^1_{22}, n^2_{22})$. This yields:

$$b_1 = \sum_{n^1_{12}=0}^{M^1_2} \sum_{n^1_{22}=0}^{M^1_2 - n^1_{12}} \sum_{n^2_{22}=0}^{\min\{M^2_2, N_2 - n^1_{12} - n^1_{22}\}} \pi(N_1, n^1_{12}, n^1_{22}, n^2_{22}) = B_1 + O_1 \tag{164}$$

$$B_1 = \sum_{n^1_{12}=0}^{M^1_2} \sum_{n^2_{22}=0}^{\min\{M^2_2, N_2 - n^1_{12}\}} \pi(N_1, n^1_{12}, \min\{M^1_2 - n^1_{12}, N_2 - n^1_{12} - n^2_{22}\}, n^2_{22}) = b_1 - O_1 \tag{165}$$

$$O_1 = \sum_{n^1_{12}=0}^{M^1_2-1} \sum_{n^1_{22}=0}^{M^1_2-n^1_{12}-1} \sum_{n^2_{22}=0}^{\min\{M^2_2,N_2-n^1_{12}-n^1_{22}-1\}} \pi(N_1, n^1_{12}, n^1_{22}, n^2_{22}) = b_1 - B_1 \tag{166}$$

$$\begin{aligned} B_2 = &\sum_{n^1_{11}=0}^{N_1-1} \sum_{n^1_{22}=0}^{M^1_2} \pi(n^1_{11}, 0, n^1_{22}, \min\{M^2_2, N_2 - n^1_{22}\}) + \\ &\sum_{n^1_{12}=0}^{M^1_2} \sum_{n^1_{22}=0}^{M^1_2-n^1_{12}} \pi(N_1, n^1_{12}, n^1_{22}, \min\{M^2_2, N_2 - n^1_{12} - n^1_{22}\}) \end{aligned} \tag{167}$$

However, the PASTA property can no longer be used when the arrivals come from one of the stations instead of from outside the system. Instead, the blocking probability can then be determined by computation of a Palm probability. This means that $B^1_{11,2}$ can be computed as follows:

$$B^1_{11,2} = \frac{U}{V+W}, \quad \text{where} \tag{168}$$

$$U = \frac{1}{c} \sum_{n^1_{11}=1}^{N_1} \sum_{n^2_{22}=0}^{M^2_2} \pi(n^1_{11}, 0, \min\{M^1_2, N_2 - n^2_{22}\}, n^2_{22}) \cdot (p^1_{1,2} n^1_{11} \mu^1_{11}) \tag{169}$$

$$V = \frac{1}{c} \sum_{n^1_{11}=1}^{N_1-1} \sum_{n^1_{22}=0}^{M^1_2} \sum_{n^2_{22}=0}^{\min\{M^2_2,N_2-n^1_{22}\}} \pi(n^1_{11}, 0, n^1_{22}, n^2_{22}) \cdot (p^1_{1,2} n^1_{11} \mu^1_{11}) \tag{170}$$

$$W = \frac{1}{c} \sum_{n^1_{12}=0}^{M^1_2} \sum_{n^1_{22}=0}^{M^1_2-n^1_{12}} \sum_{n^2_{22}=0}^{\min\{M^2_2,N_2-n^1_{12}-n^1_{22}\}} \pi(N_1, n^1_{12}, n^1_{22}, n^2_{22}) \cdot (p^1_{1,2} n^1_{11} \mu^1_{11}) \tag{171}$$

Similarly, $B^1_{1,2}$ can be determined as follows:

$$B^1_{1,2} = \frac{X}{Y+Z}, \quad \text{where} \tag{172}$$

$$X = \frac{1}{c} \sum_{n^1_{11}=1}^{N_1} \sum_{n^2_{22}=0}^{M^2_2} \pi(n^1_{11}, 0, \min\{M^1_2, N_2 - n^2_{22}\}, n^2_{22}) \cdot (p^1_{1,2} n^1_{11} \mu^1_{11}) \tag{173}$$

$$Y = \frac{1}{c} \sum_{n^1_{11}=1}^{N_1-1} \sum_{n^1_{22}=0}^{M^1_2} \sum_{n^2_{22}=0}^{\min\{M^2_2,N_2-n^1_{22}\}} \pi(n^1_{11}, 0, n^1_{22}, n^2_{22}) \cdot (p^1_{1,2} n^1_{11} \mu^1_{11}) \tag{174}$$

$$Z = \frac{1}{c} \sum_{n^1_{12}=0}^{M^1_2} \sum_{n^1_{22}=0}^{M^1_2-n^1_{12}} \sum_{n^2_{22}=0}^{\min\{M^2_2,N_2-n^1_{12}-n^1_{22}\}} \pi(N_1, n^1_{12}, n^1_{22}, n^2_{22}) \cdot (p^1_{1,2} n^1_{11} \mu^1_{11} + p^1_{1,2} n^1_{12} \mu^1_{12}) \tag{175}$$

It can be noted that in (169) and (173) we only sum over states with zero overflowed type 1 jobs at station 2 ($n^1_{12} = 0$). The reason for this is that, because of the call packing assumption, a job that finishes service at station 1 can only be blocked at station 2 if there are no overflowed type 1 jobs present. It is also noted that there is only one difference between (168) and (172), which is that the arrival stream of finished overflowed type 1 jobs is only included in (172). This is reflected in the addition of the term $p^1_{1,2} n^1_{12} \mu^1_{12}$ in (175). Finally, it can be seen that the factor $1/c$ is included in the expressions for U, V, W, X, Y and Z. The reason for this is that it is not necessary to compute the normalizing constant c in the product form (39), since these cancel out in (168) and (172).

B.1.2 Serial overflow system without call packing

Let π be the steady-state distribution of the number of jobs in the serial overflow system without call packing. For example, π can be determined by using a numerical algorithm (e.g. GTH algorithm), as discussed in Section 3.4.2. Because of the PASTA property of Poisson arrivals, the blocking probabilities b_1, B_1, O_1 and B_2 can then be computed by summing $\pi(\mathbf{n})$ over the appropriate states $\mathbf{n} = (n_{11}^1, n_{12}^1, n_{22}^1, n_{22}^2)$. This yields:

$$b_1 = \sum_{n_{12}^1=0}^{M_2^1} \sum_{n_{22}^1=0}^{M_2^1-n_{12}^1} \sum_{n_{22}^2=0}^{\min\{M_2^2, N_2-n_{12}^1-n_{22}^1\}} \pi(N_1, n_{12}^1, n_{22}^1, n_{22}^2) = B_1 + O_1 \tag{176}$$

$$B_1 = \sum_{n_{12}^1=0}^{M_2^1} \sum_{n_{22}^2=0}^{\min\{M_2^2, N_2-n_{12}^1\}} \pi(N_1, n_{12}^1, \min\{M_2^1 - n_{12}^1, N_2 - n_{12}^1 - n_{22}^2\}, n_{22}^2) = b_1 - O_1 \tag{177}$$

$$O_1 = \sum_{n_{12}^1=0}^{M_2^1-1} \sum_{n_{22}^1=0}^{M_2^1-n_{12}^1-1} \sum_{n_{22}^2=0}^{\min\{M_2^2, N_2-n_{12}^1-n_{22}^1-1\}} \pi(N_1, n_{12}^1, n_{22}^1, n_{22}^2) = b_1 - B_1 \tag{178}$$

$$B_2 = \sum_{n_{11}^1=0}^{N_1} \sum_{n_{12}^1=0}^{M_2^1} \sum_{n_{22}^1=0}^{M_2^1-n_{12}^1} \pi(N_1, n_{12}^1, n_{22}^1, \min\{M_2^2, N_2 - n_{12}^1 - n_{22}^1\}) \tag{179}$$

Here, it is noted that the situation at station 2 does not have any effect on the number of jobs at station 1 when overflowed type 1 jobs at station 2 do no switch to station 1. As a consequence, b_1 can also be calculated using the Erlang loss formula (see also (23) in Example 7 in Section 2.5.1), that is, b_1 can also be determined as follows:

$$b_1 = B_{Er}(\lambda_1, \mu_1, N_1) = \left(\sum_{n=0}^{N_1} \frac{1}{n!} \left(\frac{\lambda_1}{\mu_1}\right)^n \right)^{-1} \frac{1}{N_1!} \left(\frac{\lambda_1}{\mu_1}\right)^{N_1} \tag{180}$$

Next, the PASTA property can no longer be used when the arrivals come from one of the stations instead of from outside the system. Instead, the blocking probability can be determined by computation of a Palm probability. This means that $B_{11,2}^1$ can be computed as follows:

$$B_{11,2}^1 = \frac{D}{F}, \quad \text{where} \tag{181}$$

$$D = \sum_{n_{11}^1=0}^{N_1} \sum_{n_{12}^1=0}^{M_2^1} \sum_{n_{22}^2=0}^{\min\{M_2^2, N_2-n_{12}^1\}} \pi(n_{11}^1, n_{12}^1, \min\{M_2^1 - n_{12}^1, N_2 - n_{12}^1 - n_{22}^2\}, n_{22}^2) \cdot (p_{1,2}^1 n_{11}^1 \mu_{11}^1) \tag{182}$$

$$F = \sum_{n_{11}^1=0}^{N_1} \sum_{n_{12}^1=0}^{M_2^1} \sum_{n_{22}^1=0}^{M_2^1-n_{12}^1} \sum_{n_{22}^2=0}^{\min\{M_2^2, N_2-n_{12}^1-n_{22}^1\}} \pi(n_{11}^1, n_{12}^1, n_{22}^1, n_{22}^2) \cdot (p_{1,2}^1 n_{11}^1 \mu_{11}^1) \tag{183}$$

Similarly, $B_{1,2}^1$ can be determined as follows:

$$B_{1,2}^1 = \frac{G}{H}, \quad \text{where} \tag{184}$$

$$G = \sum_{n_{11}^1=0}^{N_1} \sum_{n_{12}^1=0}^{M_2^1} \sum_{n_{22}^2=0}^{\min\{M_2^2, N_2-n_{12}^1\}} \pi(n_{11}^1, n_{12}^1, \min\{M_2^1 - n_{12}^1, N_2 - n_{12}^1 - n_{22}^2\}, n_{22}^2) \cdot (p_{1,2}^1 n_{11}^1 \mu_{11}^1) \tag{185}$$

$$H = \sum_{n^1_{11}=0}^{N_1} \sum_{n^1_{12}=0}^{M^1_2} \sum_{n^1_{22}=0}^{M^1_2-n^1_{12}} \sum_{n^2_{22}=0}^{\min\{M^2_2, N_2-n^1_{12}-n^1_{22}\}} \pi(n^1_{11}, n^1_{12}, n^1_{22}, n^2_{22}) \cdot (p^1_{1,2} n^1_{11}\mu^1_{11} + p^1_{1,2} n^1_{12}\mu^1_{12}) \quad (186)$$

It is noted that the only difference between (181) and (184) is the arrival stream of overflowed type 1 jobs that complete service, which is included in (184), but not in (181). As a consequence, the term $p^1_{1,2} n^1_{12}\mu^1_{12}$ is only added in (186).

B.2 Results of numerical experiments

Section 3.5.3 contains some numerical results of the blocking probabilities for the overflow system that is subject of Chapter 3. In this appendix, some additional numerical results are provided. Table 20 contains the parameter values, and the resulting blocking probabilities are given in Table 21. Here, it is noted that the parameter values for the experiments correspond to the parameter values for the experiments in Section 3.5.3.

Table 20: Numerical experiments: Parameter values

Experiment #	λ_1	λ_2	μ^1_{11}	μ^1_{12}	μ^1_{22}	μ^2_{22}	$p^1_{1,2}$	N_1	N_2	M^1_2	M^2_2
Experiment 1	50	15	10	-	20	16	1	5	5	5	1
Experiment 2	15	30	1	2	3	4	-	15	10	10	10
Experiment 3	14	16	3	2	4	6	1	7	10	-	10

Table 21: Numerical experiments: Resulting blocking probabilities

Experiment #	μ^1_{12}, $p^1_{1,2}$ or M^1_2	CP?[(a)]	b_1	B_1	O_1	B_2	$B^1_{11,2}$	$B^1_{1,2}$
Experiment 1	$\mu^1_{12} = 1$	Yes	0.5562	0.1617	0.3945	0.5360	0.0577	0.0567
	$\mu^1_{12} = 1$	No	0.2849	0.2109	0.0739	0.7679	0.7170	0.6498
	$\mu^1_{12} = 15$	Yes	0.4633	0.0974	0.3658	0.5166	0.0731	0.0637
	$\mu^1_{12} = 15$	No	0.2849	0.0652	0.2197	0.5314	0.2097	0.1605
Experiment 2	$p^1_{1,2} = 0.1$	Yes	0.3764	0.0741	0.3023	0.1500	0.0879	0.0819
	$p^1_{1,2} = 0.1$	No	0.1803	0.0477	0.1327	0.1815	0.1883	0.1621
	$p^1_{1,2} = 1$	Yes	0.3116	0.1161	0.1955	0.3377	0.2612	0.2507
	$p^1_{1,2} = 1$	No	0.1803	0.0716	0.1088	0.3694	0.3728	0.3291
Experiment 3	$M^1_2 = 1$	Yes	0.1120	0.0901	0.0219	0.0011	0.7606	0.7590
	$M^1_2 = 1$	No	0.1000	0.0823	0.0177	0.0011	0.8047	0.7892
	$M^1_2 = 10$	Yes	0.1882	0.0167	0.1715	0.0560	0.0410	0.0400
	$M^1_2 = 10$	No	0.1000	0.0096	0.0905	0.0795	0.0832	0.0756

[(a)] Blocking probabilities for system with call packing (Yes) or system without call packing (No).

Appendix C

Simulation: Additional information

In Section 3.6.1, the serial overflow system that is subject of Chapter 3 is analyzed by discrete-event simulation. This appendix provides some information on how the simulations are performed. First of all, the simulation model, and in particular the required input and provided output of the simulation model, are discussed in Appendix C.1. Next, Appendix C.2 describes how the length of the warm-up period, run length and number of replications are determined. Finally, verification and validation are discussed in Appendix C.3.

C.1 The simulation model

In Section 3.6.1, discrete-event simulation is used to analyze whether or not the overflow system that is studied in Chapter 3 is insensitive. The simulation model of the overflow system is implemented in Rockwell's Arena software. The Arena file can be obtained from the author upon request. Below, it is discussed which input is required by the simulation model and which output it produces.

First of all, the simulation model requires the following input (see also Table 1 in Section 3.2):

- The number of servers at station 1, $N_1 \in \mathbb{N}$.
- The number of servers at station 2, $N_2 \in \mathbb{N}$.
- The maximum number of type 1 jobs allowed at station 2, $M_2^1 \in \{0, ..., N_2\}$.
- The maximum number of type 2 jobs allowed at station 2, $M_2^2 \in \{0, ..., N_2\}$.
- The arrival rate of type 1 jobs at station 1, $\lambda_1 > 0$.
- The arrival rate of type 2 jobs at station 2, $\lambda_2 > 0$.
- The service rate of type 1 jobs at station 1, $\mu_{11}^1 > 0$.
- The service rate of overflowed type 1 jobs at station 2, $\mu_{12}^1 > 0$.
- The service rate of non-overflowed type 1 jobs at station 2, $\mu_{22}^1 > 0$.
- The service rate of type 2 jobs at station 2, $\mu_{22}^2 > 0$.
- Probability that a type 1 job that completes service at station 1 goes to station 2, $p_{1,2}^1 \in [0, 1]$.
- The service time distribution, which is assumed to be either exponential or lognormal. Moreover, if the service times are lognormally distributed, the coefficient of variation (denoted by cv) is required, $cv > 0$.

- It must be specified whether the system with call packing or the system without call packing is considered. If the system with call packing is considered, it must also be specified whether the service of an overflowed type 1 job that switches from station 2 to station 1 is preemptively resumed (resume) or completely started over (resample). Besides that, by default, the overflowed job that is present for the longest time switches from station 2 to station 1 (i.e. FIFO). This could be changed to either LIFO or random.

As discussed in Section 3.6.1, we are interested in determining the steady-state probability B_1, which is equal to the proportion of time that the system spends in a state $\mathbf{n}$ with $n_{11}^1 = N_1$ and $(n_{12}^1 + n_{22}^1 + 1, n_{22}^2) \notin \boldsymbol{C_2}$.[1] To this end, we formulate a Boolean expression that is equal to one if the system is in such a state and equal to zero otherwise. Subsequently, a time-persistent statistic on this expression is created (see Appendix C.2.2 and, for example, [48] for a further discussion on time-persistent statistics). This makes it possible to keep track of the time that the system spends in a state $\mathbf{n}$ with $n_{11}^1 = N_1$ and $(n_{12}^1 + n_{22}^1 + 1, n_{22}^2) \notin \boldsymbol{C_2}$.

Moreover, next to the time-persistent statistic, the following counter statistics are also defined:

- The number of type 1 jobs that arrive at station 1.
- The number of type 1 jobs that are blocked at station 1.
- The number of type 1 jobs that are overflowed to station 2.
- The number of type 2 jobs that arrive at station 2.
- The number of type 2 jobs that are blocked at station 2.
- The number of type 1 jobs that complete service at station 1 and are then routed to station 2.
- The number of overflowed type 1 jobs that stay at station 2 after service completion.
- The number of non-overflowed type 1 jobs that are blocked at station 2.

From these counter statistics, the blocking probabilities that are described in Section 3.5.1 can be determined. This can be useful, for example, when verifying the simulation model. It is noted, though, that Section 3.6.1 only presents results of the time-persistent statistic.

C.2 Warm-up period, run length and number of replications

C.2.1 Non-terminating simulation and method of replication-deletion

In Section 3.6.1, discrete-event simulation is used to determine the steady-state probability B_1 when the service times are lognormally distributed. The simulation model can therefore be referred to as a non-terminating (or infinite-horizon or steady-state) simulation model (see e.g. [6, 35, 37, 40, 48] and references therein). This means that a "natural" event that specifies the length of a run does not exist ([37], p. 79). A method that is often used for non-terminating simulations and also used in this report is the method of replication-deletion (see e.g. [37, 48]). This means that multiple replications are executed, and for each replication the observations from the beginning of a run

[1] Because of the PASTA property of Poisson arrivals, this probability can also be interpreted as the blocking probability of type 1 jobs that arrive at station 1. Therefore, it is denoted by B_1 (see also Section 3.6.1).

(i.e. from the warm-up period) are deleted. In order to use the method of replication-deletion, the following three things should thus be specified:

- Length of the warm-up period.
- Run length.
- Number of replications.

In Appendix C.2.2, it is discussed how the length of the warm-up period is determined. Besides that, the determination of the run length and number of replications is described in Appendix C.2.3.

C.2.2 Determining the length of the warm-up period

As discussed in Appendices C.1 and C.2.1, the steady-state (or long-run) behaviour of the overflow system that is studied by simulation, and in particular the steady-state probability B_1, are of interest. In order to study the steady-state behaviour accurately, it is important that the effects of the initial conditions do not influence the simulation results (see e.g. [6]). Therefore, it is useful to "warm up" the simulation model (see e.g. [6, 35, 37, 38, 40, 48]). This means that a so-called warm-up period is specified, for which the observations are excluded. Then, only the observations that are obtained after the warm-up period has ended are used to estimate the performance measure(s) of interest.

The question that then arises is which length of the warm-up period should be chosen. For this purpose, many methods have been developed (see e.g. [40, 48] and references therein). One of the most frequently used techniques for determining the length of the warm-up period is Welch's method, which is also applied in this report. The steps of Welch's method are given in Algorithm 3. It can be seen that the number of replications n, run length m and window w should be specified when Welch's method is applied. It is then stated in [48] that the number of replications n is typically recommended to be at least five, while the run length m should be longer than the expected length of the warm-up period. Besides that, in [38, 40], it is suggested to choose the window w as the smallest value of w for which the graph appears smooth.

In order to perform Welch's method, the value of Y_{ji} should then be determined for all $j = 1, ..., n$ and $i = 1, ..., m$. In the current context, Y_{ji} denotes the fraction of time that the system spends in a state $\mathbf{n}$ with $n_{11}^1 = N_1$ and $(n_{12}^1 + n_{22}^1 + 1, n_{22}^2) \notin \boldsymbol{C_2}$ during the ith interval of the jth replication. As discussed in Appendix C.1, a time-persistent statistic is defined in order to keep track of the time that the system spends in such a state. The value of Y_{ji} $(j = 1, ..., n,\ i = 1, ..., m)$ can therefore be determined from these time-persistent observations.

To this end, it is noted that the output that results from the simulation consists of a vector with times, denoted by $\vec{t}$, and a vector with corresponding values, denoted by $\vec{v}$. Here, the vector $\vec{t}$ contains the times at which the state of the system changes from a state $\mathbf{n}$ for which $n_{11}^1 = N_1$ and $(n_{12}^1 + n_{22}^1 + 1, n_{22}^2) \notin \boldsymbol{C_2}$ do not hold to a state for which these conditions hold or vice versa (i.e. the times that the Boolean expression changes from one to zero or from zero to one).[2] The vector $\vec{v}$

[2] The value of the Boolean expression is also recorded at times 0, m and L, where m and L are the run length and length of the warm-up period, respectively.

Algorithm 3 Welch's method for determining the length of the warm-up period ([38], cited in [40], p. 664)

```
Require: Simulation output of $n$ replications each of length $m$, $n, m \in \mathbb{N}$. Let $Y_{ji}$ denote the $i$th observation of the $j$th replication for $i = 1, ..., m$ and $j = 1, ..., n$. Moreover, the window $w$, which is a positive integer (i.e. $w \in \mathbb{N}$) that is smaller than or equal to $m/4$ (i.e. $w \leq m/4$) should be specified.
Calculate the average $\overline{Y}_i$ for $i = 1, ..., m$, where:
    $\overline{Y}_i = \frac{1}{n} \sum_{j=1}^{n} Y_{ji}, \qquad i = 1, ..., m \qquad (187)$
Define a moving average $\overline{Y}_i(w)$ to smooth out the high-frequency oscillations in $\overline{Y}_1$,...,$\overline{Y}_m$:
    $\overline{Y}_i(w) = \begin{cases} \frac{1}{2i-1} \sum_{s=-(i-1)}^{i-1} \overline{Y}_{i+s} & \text{for } i = 1, 2, ..., w \\ \frac{1}{2w+1} \sum_{s=-w}^{w} \overline{Y}_{i+s} & \text{for } i = w, w+1, ..., m-w \end{cases} \qquad (188)$
Plot $\overline{Y}_i(w)$ for $i = 1, ..., m - w$, and choose the warm-up length to be the value of $i$ beyond which $\overline{Y}_i(w)$ appears to be converged.
return Length of the warm-up period.
```

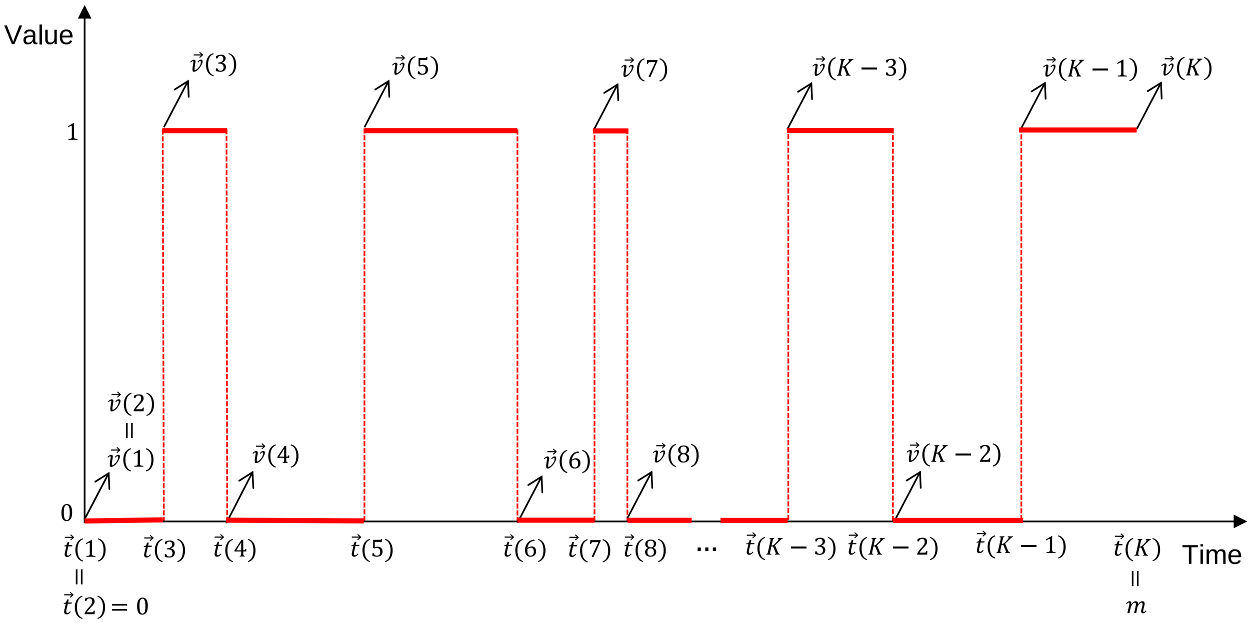


Figure 15: Example of the values of the time-persistent statistic when the run length is m and the length of the warm-up period is 0

Algorithm 4 Determining Y_{ji} for replication j and all $i = 1, ..., m$

```
Require: Simulation output for replication $j$ consisting of a $K \times 1$ vector with times, denoted by $\vec{t}$, and a $K \times 1$ vector with corresponding values, denoted by $\vec{v}$. Let $\vec{t}(k)$ and $\vec{v}(k)$ denote the $k$th element of vector $\vec{t}$ and $\vec{v}$, respectively.
Obtain a $K \times 1$ vector, denoted by $\vec{a}$, that has the following entries, denoted by $\vec{a}(k)$:
    $\vec{a}(1) = 0; \qquad \vec{a}(k) = \vec{a}(k-1) + \{\vec{t}(k) - \vec{t}(k-1)\} \cdot \vec{v}(k-1), \quad k = 2, ..., K \qquad (189)$
  Note that $\vec{a}(k)$ represents the amount of time that the system is in a state $\mathbf{n}$ with $n_{11}^1 = N_1$ and $(n_{12}^1 + n_{22}^1 + 1, n_{22}^2) \notin \boldsymbol{C_2}$ up to time $\vec{t}(k)$.
for $i = 1, ..., m-1$ do
    Find the largest element of vector $\vec{t}$ that is still smaller than or equal to $i$. Let $l$ be the corresponding index, that is, $\vec{t}(l) \leq i$ and $\vec{t}(l+1) > i$.
    Determine $Y_{ji}$. This can be done as follows:
        $Y_{ji} = \begin{cases} \vec{a}(l+1) - \{\vec{t}(l+1) - i\} \cdot \vec{v}(l) & i = 1 \\ \vec{a}(l+1) - \{\vec{t}(l+1) - i\} \cdot \vec{v}(l) - \sum_{s=1}^{i-1} Y_{js} & i > 1 \end{cases} \qquad (190)$
end for
Determine $Y_{jm}$. Since the last element of $\vec{t}$ is equal to $m$, this can be done by subtracting $\sum_{s=1}^{m-1} Y_{js}$ from the last element of $\vec{a}$, denoted by $\vec{a}(K)$ (i.e. $Y_{jm} = \vec{a}(K) - \sum_{s=1}^{m-1} Y_{js}$).
return An $m \times 1$ vector that contains the value of $Y_{ji}$ at entry $i$.
```

then contains the corresponding values (i.e. zero or one). This means that the value of the Boolean expression between times $\vec{t}(k-1)$ and $\vec{t}(k)$ is equal to $\vec{v}(k-1)$, where $\vec{t}(k)$ and $\vec{v}(k)$ denote the kth element of $\vec{t}$ and $\vec{v}$, respectively. This is visually illustrated by Figure 15.

Hence, before performing Welch's method, the Y_{ji} ($i = 1, ..., m$, $j = 1, ..., n$) must first be determined from the simulation output. This can be done as described in Algorithm 4. Then, the length of the warm-up period can be determined by Welch's method (see Algorithm 3). This is illustrated for two examples below.

Example 14 (Welch's method, 1/2)**.** Consider the serial overflow system with call packing (resume, FIFO) and exponential service times. Moreover, let the parameter values be as given in Table 11 (scenario 1) in Section 3.6.1. The required length of the warm-up period is then determined using Welch's method (see Algorithm 3) as follows.

First of all, ten replications (i.e. $n = 10$) of length one year (i.e. $m = 24 \cdot 365 = 8760$ hours) are performed. From the obtained simulation output, the Y_{ji} ($i = 1, ..., m$, $j = 1, ..., n$), which are required as input for Welch's method, are then computed using Algorithm 4. Next, a window of 1000 is chosen. This leads to the plot of $\overline{Y}_i(w)$ that is shown in Figure 16. From

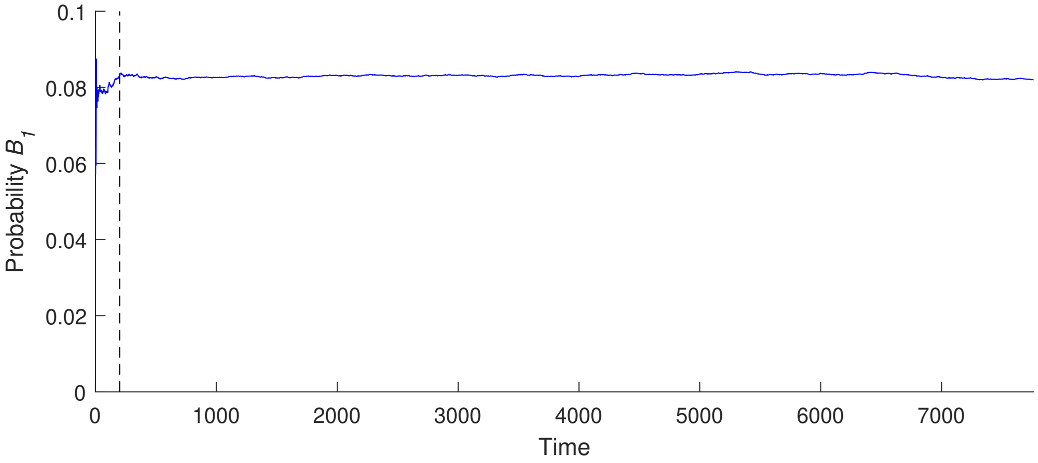


Figure 16: Welch's method: Plot of moving averages corresponding to Example 14

this plot, it appears that $\overline{Y}_i(w)$ is converged beyond $m = 200$. Therefore, the required length of the warm-up period is found to be 200 hours.

Example 15 (Welch's method, 2/2)**.** Consider the serial overflow system with call packing (resume, FIFO). Moreover, let the service times be lognormally distributed with a coefficient of variation of five and the parameter values be as given in Table 11 (scenario 1) in Section 3.6.1. The required length of the warm-up period is then determined using Welch's method (see Algorithm 3) as follows.

First of all, ten replications (i.e. $n = 10$) of length one year (i.e. $m = 24 \cdot 365 = 8760$ hours) are performed. From the obtained simulation output, the Y_{ji} ($i = 1, ..., m$, $j = 1, ..., n$), which are required as input, are then determined using Algorithm 4. Subsequently, the window is chosen to be 1000. This results in the plot of $\overline{Y}_i(w)$ that is shown in Figure 17. From this plot, it appears that $\overline{Y}_i(w)$ is converged beyond $m = 400$. Consequently, it is concluded that the required length of the warm-up period is 400 hours.

For the other cases, the required length of the warm-up period is determined using Welch's method in a similar manner as in Examples 14 and 15. This leads to the required lengths of the warm-up periods that are mentioned in Table 22. It can thus be seen that a slightly longer warm-up period is required when the service times are assumed to be lognormally distributed with a coefficient of variation of five. Moreover, it can be noted that the required lengths of the warm-up periods for scenario 1 and scenario 2 are more or less similar.

Ultimately, it is chosen to use a warm-up period of 1000 hours for all simulations. This is done for the following reasons. First of all, it is more convenient to choose the same warm-up period for all simulations. Secondly, it takes into account that Welch's method is a subjective method (see e.g. [40, 48]), which means that others may conclude that longer warm-up periods than those that

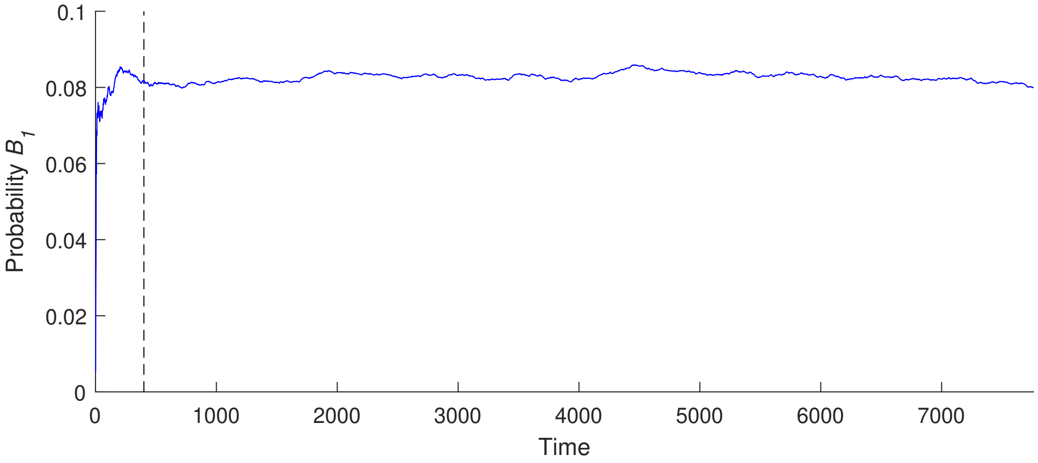


Figure 17: Welch's method: Plot of moving averages corresponding to Example 15

Table 22: Insensitivity experiment: Required lengths of the warm-up periods based on Welch's method

Assumptions	Distribution	CV	Scenario 1	Scenario 2
Call packing Resample FIFO	Exponential	1	200	200
	Lognormal	0.2	250	250
	Lognormal	5	350	350
Call packing Resume FIFO	Exponential	1	200 (see Example 14)	200
	Lognormal	0.2	250	250
	Lognormal	5	400 (see Example 15)	450
Without call packing	Exponential	1	250	250
	Lognormal	0.2	300	300
	Lognormal	5	400	400

Note: times are assumed to be in hours.

are mentioned in Table 22 are necessary.

C.2.3 Determining the run length and number of replications

After the length of the warm-up period is specified, the run length and number of replications should also be determined. In this section, it is discussed how this can be done. The information that is provided is mainly based on [6, 35, 37, 48].

First of all, the run length should be much larger than the length of the warm-up period ([37], p. 81). There are then several ways to get an indication of an appropriate run length. For example, in [7], it is mentioned that, as a rule of thumb, the run length should be at least ten times the length of the warm-up period. Besides that, more sophisticated methods to determine the run length, such as the convergence method that is mentioned in [46] (see also e.g. [50]), could also be thought of.

Secondly, the number of replications is often chosen such that the $100(1-\alpha)\%$ confidence intervals of the performance measures of interest (in this case, the steady-state probability B_1) are not too wide. Therefore, it is first described how the confidence interval of a performance measure can be determined. To this end, let $\overline{X}_n$ be the average value of the performance measure across n replications. The $100(1-\alpha)\%$ confidence interval, denoted by CI, can then be found as follows:

$$CI = [\overline{X}_n - h, \overline{X}_n + h] \tag{191}$$

Here, h is the half-width, which is as follows:

$$h = t_{n-1,1-\alpha/2} \frac{S_n}{\sqrt{n}}, \tag{192}$$

Here, n is the number of replications, S_n the standard deviation across the n replications and $t_{n-1,1-\alpha/2}$ the upper $1-\alpha/2$ critical value for a t-distribution with $n-1$ degrees of freedom. Besides that, it is noted that the observed values of the performance measure across the replications should be independent and identically distributed and normally distributed, which can be reasonably assumed (see e.g. [35], pp. 237-239, for a further discussion).

As mentioned above, it is thus desired to choose the number of replications such that the $(1-\alpha)\%$ confidence interval in (191) is not too wide. This means that the half-width h should be smaller than or equal to a desired half-width, denoted by h_d. Multiple methods to achieve this desired half-width h_d can then be thought of. For example, it can be chosen to sequentially increase the number of replications and compute the half-width after each replication. The simulation can then be stopped when the desired half-width h_d is met. Another option is to perform a pilot run of n_0 replications. If this leads to a half-width h_0 that is larger than the desired half-width h_d (i.e. $h_0 > h_d$), some more replications can be made. The required number of replications, denoted by n_r, can then be approximated in several ways. For example, if it is assumed that $t_{n_r-1,1-\alpha/2} \approx t_{n_0-1,1-\alpha/2}$ and $S_{n_r} \approx S_{n_0}$, the following expression for n_r can be derived from (192):

$$n_r \approx n_0 \left(\frac{h_0}{h_d}\right)^2 \tag{193}$$

Hence, if it appears from the simulation results that the desired half-width h_d is not met yet, the number of replications can be increased to n_r. Besides that, instead of increasing the number of replications, the precision can also be improved by increasing the run length.

The following two examples then illustrate how the run length and number of replications are determined.

Example 16 (Run length and number of replications, 1/2)**.** Consider the overflow system with call packing (resume, FIFO) and exponential service times. Moreover, let the parameter values be as given in Table 11 (scenario 1) in Section 3.6.1. For this system, the steady-state probability B_1 is determined using discrete-event simulation. To this end, the length of the warm-up period is first determined, which leads to a warm-up period of 1000 hours (see

Appendix C.2.2). Besides that, the run length is set equal to five years (i.e. $24 \cdot 365 \cdot 5 = 43800$ hours), which is considerably longer than the warm-up period. Finally, a level of significance α of 0.05 and a desired half-width h_d of 0.0005 are chosen.

Now, a pilot run of ten replications is performed (i.e. $n_0 = 10$). From the simulation output, the average value of B_1 across the ten replications, denoted by $\overline{X}_{10}$, and standard deviation, denoted by S_{10}, can then be obtained. Subsequently, the half-width, denoted by h_0, is as follows:

$$h_0 = t_{9,0.975} \frac{S_{10}}{\sqrt{10}} \approx 0.0004 < h_d \tag{194}$$

It can thus be noted that the resulting half-width h_0 is already smaller than the desired half-width h_d. The steady-state probability B_1 and corresponding 95% confidence interval CI are therefore found to be as follows:

$$B_1 = \overline{X}_{10} \approx 0.0828; \quad CI = [\overline{X}_{10} - h_0, \overline{X}_{10} + h_0] \approx [0.0825, 0.0832] \tag{195}$$

Example 17 (Run length and number of replications, 2/2)**.** Consider the overflow system with call packing (resume, FIFO). Moreover, let the service times be lognormally distributed with a coefficient of variation of five, and let the parameter values be as given in Table 11 (scenario 1) in Section 3.6.1. Discrete-event simulation is then used to determine the steady-state probability B_1. To this end, the length of the warm-up period is first determined, which leads to a warm-up period of 1000 hours (see Appendix C.2.2). Besides that, a run length of five years (i.e. $24 \cdot 365 \cdot 5 = 43800$ hours) is (initially) chosen. Finally, a level of significance α of 0.05 and a desired half-width h_d of 0.0005 are chosen.

Then, a pilot run of ten replications is performed (i.e. $n_0 = 10$). The average value of B_1 across the ten replications, denoted by $\overline{X}_{10}$, and standard deviation, denoted by S_{10}, can then be obtained from the simulation output. The following half-width, denoted by h_0, is then obtained:

$$h_0 = t_{9,0.975} \frac{S_{10}}{\sqrt{10}} \approx 0.0013 > h_d \tag{196}$$

Hence, it appears that the resulting half-width h_0 is larger than the desired half-width h_d. In order to meet the desired half-width, it can therefore be an option to increase the number of replications. As mentioned above, the number of replications that is required to achieve the desired half-width h_d, denoted by n_r, can then be approximated in several ways. For example, according to the approximation in (193), n_r is as follows:

$$n_r \approx 10 \cdot \left(\frac{h_0}{h_d}\right)^2 \approx 65 \tag{197}$$

The number of replications could therefore be set to 65, after which the simulation can be performed with the same warm-up period of 1000 hours and run length of five years. However, instead of increasing the number of replications, the half-width could also be reduced by increasing the run length. It is therefore chosen to set the run length to ten years (i.e. $24 \cdot 365 \cdot 10 = 87600$ hours) and the number of replications to 30. This leads to the following half-width, denoted by h:

$$h = t_{29,0.975} \frac{S_{30}}{\sqrt{30}} \approx 0.0003 < h_d \tag{198}$$

It can thus be seen that the resulting half-width is now smaller than the desired half-width h_d. The following steady-state probability B_1 and corresponding 95% confidence interval CI are therefore obtained:

$$B_1 = \overline{X}_{30} \approx 0.0827; \quad CI = [\overline{X}_{30} - h, \overline{X}_{30} + h] \approx [0.0824, 0.0830] \tag{199}$$

For the other cases, the run length and number of replications are determined in a similar manner as in Examples 16 and 17. This leads to a run length of five years (i.e. $24 \cdot 365 \cdot 5 = 43800$ hours) and ten replications for all instances that are considered in Section 3.6.1, except those that consider lognormally distributed service times with a coefficient of variation of five. For these cases, the run length is set equal to ten years (i.e. $24 \cdot 365 \cdot 10 = 87600$ hours), and the number of replications is chosen equal to 30. This results in the steady-state probabilities and corresponding 95% confidence intervals that are given in Table 13 in Section 3.6.1.

C.3 Verification and validation

An important aspect when a simulation analysis is performed is to verify and validate the simulation model. Therefore, this appendix briefly discusses verification and validation. The information that is provided is mainly based on [6] (pp. 22-25), [35] (pp. 512-514) and [48] (pp. 14-15).

First of all, it is important that it is ensured that the simulation model behaves as intended. This task can be referred to as verification. Moreover, when the simulation model represents a real system (instead of a conceptual system), it should also be validated. This means that it is determined whether the simulation model provides an adequate representation of the real system.

In this report, discrete-event simulation is used to study the overflow system that is described in Section 3.3 under the assumption of lognormally distributed service times. In order to verify the simulation model, several steps are then taken, among which the following:

- The simulation model is considered with deterministic interarrival and service times and only a limited number of jobs. It is then watched how these jobs flow through the system in order to check whether the model behaves as expected.
- As discussed in Section 3.6.1, the steady-state probability B_1 is computed as the proportion of time that the system spends in a state with $n_{11}^1 = N_1$ and $(n_{12}^1 + n_{22}^1 + 1, n_{22}^2) \notin \boldsymbol{C_2}$. Because of the PASTA property of Poisson arrivals, this steady-state probability should be the same as the probability that type 1 jobs that arrive at station 1 are blocked. Therefore, the resulting

Table 23: Insensitivity experiment: Verification of simulation model by comparison of B_1

Assumptions	Scenario	Numerical/analytical$^{(a)}$	Simulation
Call packing (resample, FIFO)	Scenario 1	0.0829	0.0829 (0.0825-0.0833)
	Scenario 2	0.0898	0.0897 (0.0893-0.0901)
Call packing (resume, FIFO)	Scenario 1	0.0829	0.0828 (0.0825-0.0832)
	Scenario 2	0.0898	0.0895 (0.0892-0.0898)
Without call packing	Scenario 1	0.0592	0.0592 (0.0589-0.0594)
	Scenario 2	0.0636	0.0636 (0.0633-0.0639)

$^{(a)}$For the system with call packing (both for resume and resample), the steady-state probability B_1 is computed from the product form (39). For the system without call packing, the steady-state probability B_1 is determined from the steady-state distribution that is obtained by applying the Gauss-Seidel method.

steady-state probability is compared with the resulting blocking probability, which indeed leads to similar results.

- The simulation model is considered with service times that are exponentially distributed instead of lognormally distributed. In this case, the steady-state probability B_1 can also be analytically or numerically determined (see Section 3.6.1). If the simulation model behaves as intended, the resulting steady-state probabilities should therefore be close to the corresponding analytical and numerical steady-state probabilities. As illustrated by the results in Table 23, this indeed appears to be the case.

Appendix D

Matlab code

This appendix contains and discusses the Matlab code that is used in this report. Because of space considerations, the appendix is included in a separate file. It can be obtained from the author upon request.

References

[1] Adan, I., & Resing, J. (2015). *Queueing Systems.* Eindhoven University of Technology. Retrieved from: https://www.win.tue.nl/~iadan/queueing.pdf (March 10th, 2021).

[2] Armony, M., Chan, C.W., & Zhu, B. (2018). Critical care capacity management: Understanding the role of a Step Down Unit. *Production and Operations Management, 27*(5), 859-883.

[3] Asaduzzaman, M., & Chaussalet, T.J. (2014). Capacity planning of a perinatal network with generalised loss network model with overflow. *European Journal of Operational Research, 232*(1), 178-185.

[4] Baccelli, F., & Brémaud, P. (1994). *Elements of Queueing Theory: Palm-martingale Calculus and Stochastic Recurrences.* Berlin: Springer.

[5] Balsamo, S., Harrison, P.G., & Marin, A. (2010). A unifying approach to product-forms in networks with finite capacity constraints. *ACM SIGMETRICS Performance Evaluation Review, 38*(1), 25-36.

[6] Banks, J. (1998). Principles of simulation. In J. Banks (ed.), *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice* (pp. 3-31). New York: Wiley.

[7] Banks, J., Carson, J.S., Nelson, B.L., & Nicol, D.M. (2005). *Discrete-Event System Simulation* (Ed. 4). Upper Saddle River, NJ: Prentice-Hall.

[8] Baskett, F., Chandy, K.M., Muntz, R.R., & Palacios, F.G. (1975). Open, closed, and mixed networks of queues with different classes of customers. *Journal of the Association for Computing Machinery, 22*(2), 248-260.

[9] Bekker, R., Koole, G., & Roubos, D. (2017). Flexible bed allocations for hospital wards. *Health Care Management Science, 20*(4), 453-466.

[10] Berry, L.T.M., & Henderson, W. (1989). Some exact results in performance analysis of alternative routing communications networks. *A.T.R., 23*(1), 35-42.

[11] Borst, S., Boucherie, R.J., & Boxma, O.J. (1999). ERMR: A generalised equivalent random method for overflow systems with repacking. In *Proceedings of the 16th International Teletraffic Conference* (pp. 313-323). Edinburgh.

[12] Boucherie, R.J. (1992). *Product-form in queueing networks* (Doctoral dissertation). Vrije Universiteit, Amsterdam.

[13] Boucherie, R.J. (1994). A characterization of independence for competing Markov chains with applications to stochastic Petri nets. *IEEE Transactions on Software Engineering, 20*(7), 536-544.

[14] Boucherie, R.J., & Van Dijk, N.M. (1997). On the arrival theorem for product form queueing networks with blocking. *Performance Evaluation, 29*(3), 155-176.

[15] Brockmeyer, E., Halstrøm, H.L., & Jensen, A. (1948). The life and works of A.K. Erlang. *Transactions of the Danish Academy of Technical Sciences, 2*.

[16] Chan, Y.-C., Wong, E.W.M., Joynt, G., Lai, P., & Zukerman, M. (2018). Overflow models for the admission of intensive care patients. *Health Care Management Science, 21*(4), 554-572.

[17] Chandy, K.M., Howard, J.H., & Towsley, D.F. (1977). Product form and local balance in queueing networks . *Journal of the Association for Computing Machinery, 24*(2), 250-263.

[18] Chandy, K.M., & Martin, A.J. (1983). A characterization of product-form queuing networks. *Journal of the Association for Computing Machinery, 30*(2), 286-299.

[19] De Bruin, A.M., Bekker, R., Van Zanten, L., & Koole, G.M. (2010). Dimensioning hospital wards using the Erlang loss model. *Annals of Operations Research, 178*(1), 23-43.

[20] Dobson, G., Lee, H., & Pinker, E. (2010). A model of ICU bumping. *Operations Research, 58*(6), 1564-1576.

[21] El-Taha, M., & Heath, J.R. (2000). Traffic overflow in loss systems with selective trunk reservation. *Performance Evaluation, 41*(4), 295-306.

[22] Gordon, W.J., & Newell, G.F. (1967). Closed queuing systems with exponential servers. *Operations Research, 15*(2), 254-265.

[23] Green, L.V. (2002). How many hospital beds? *INQUIRY, 39*(4), 400-412.

[24] Green, L.V. (2006). Queueing analysis in healthcare. In R.W. Hall (ed.), Patient Flow: Reducing Delay in Healthcare Delivery (pp. 281-307). New York: Springer.

[25] Griffiths, J.D., Price-Lloyd, N., Smithies, M., & Williams, J. (2006). A queueing model of activities in an intensive care unit. *IMA Journal of Management Mathematics, 17*(3), 277-288.

[26] Henderson, W., & Taylor, P.G. (1988). Alternative routing networks and interruptions. In *Proceedings of the 12th International Teletraffic Conference* (pp. 5.1B.2.1-5.1B.2.7). Torino.

[27] Hordijk, A., & Ridder, A. (1987). Stochastic inequalities for an overflow model. *Journal of Applied Probability, 24*(3), 696-708.

[28] Hordijk, A., & Ridder, A. (1988). Insensitive bounds for the stationary distribution of non-reversible Markov chains. *Journal of Applied Probability, 25*(1), 9-20.

[29] Izady, N., & Mohamed, I. (2021). A clustered overflow configuration of inpatient beds in hospitals *Manufacturing and Service Operations Management, 23*(1), 139-154.

[30] Jackson, J.R. (1957). Networks of waiting lines. *Operations Research, 5*(4), 518-521.

[31] Jackson, R.R.P. (1954). Queueing systems with phase type service. *Operational Research Society, 5*(4), 109-120.

[32] Jensen, P.A., & Bard, J.F. (2003). *Operations Research: Models and Methods*. New York: Wiley.

[33] Karlin, S., & Taylor, H.M. (1975). *A first course in stochastic processes* (Ed. 2). New York: Academic Press.

[34] Kelly, F.P. (1979). *Reversibility and Stochastic Networks*. New York: Wiley.

[35] Kelton, W., Sadowski, R., & Sadowski, D. (2002). *Simulation with Arena* (Ed. 2). New York:

McGraw-Hill.

[36] Kingman, J.F.C. (1969). Markov population processes. *Journal of Applied Probability, 6*(1), 1-18.

[37] Law, A.M. (2007). Statistical analysis of simulation output data: The practical state of the art. In *Proceedings of the 2007 Winter Simulation Conference* (pp. 77-83). Washington, DC.

[38] Law, A.M., & Kelton, W.D. (2000). *Simulation Modeling and Analysis* (Ed. 3). New York: McGraw-Hill.

[39] Litvak, N., Rijsbergen, M., Boucherie, R.J., & Houdenhoven, M. (2008). Managing the overflow of intensive care patients. *European Journal of Operational Research, 185*(3), 998-1010.

[40] Mahajan, P.S., & Ingalls, R.G. (2004). Evaluation of methods used to detect warm-up period in steady state simulation. In *Proceedings of the 2004 Winter Simulation Conference* (pp. 663-671). Washington, DC.

[41] Mathews, K.S., & Long, E.F. (2015). A conceptual framework for improving critical care patient flow and bed use. *Annals of the American Thoracic Society, 12*(6), 886-894.

[42] McManus, M.L., Long, M.C., Copper, A., & Litvak, E. (2004). Queuing theory accurately models the need for critical care. *Anesthesiology, 100*(5), 1271-1276.

[43] Pittel, B. (1979). Closed exponential networks of queues with saturation: The Jackson-type stationary distribution and its asymptotic analysis. *Mathematics of Operations Research, 4*(4), 357-378.

[44] Prin, M., & Wunsch, H. (2014). The role of stepdown beds in hospital care. *American Journal of Respiratory and Critical Care Medicine, 190*(11), 1210-1216.

[45] Quarteroni, A., Sacco, R., & Saleri, F. (2000). *Numerical Mathematics* (Ed. 1). New York: Springer.

[46] Robinson, S. (2004). *Simulation: The Practice of Model Development and Use* (Ed. 1). Chichester, U.K.: Wiley.

[47] Ross, K.W. (1995). *Multiservice Loss Models for Broadband Telecommunication Networks* (Ed. 1). London: Springer.

[48] Rossetti, M.D. (2016). *Simulation modeling and Arena* (Ed. 2). Hoboken: Wiley.

[49] Schehrer, R.G. (1997). A two moments method for overflow systems with different mean holding times. In *Proceedings of the 15th International Teletraffic Conference* (pp. 1303-1314). Washington D.C.

[50] Schneider, A.J., Besselink, P.L., Zonderland, M.E., Boucherie, R.J., Van den Hout, W.B., Kievit, J., Bilars, P., Fogteloo, A.J., & Rabelink, T.J. (2018). Allocating emergency beds improves the emergency admission flow. *Interfaces*, 48(4), 384-394.

[51] Serfozo, R. (2009). *Basics of Applied Stochastic Processes* (Ed. 1). Berlin: Springer.

[52] Shortle, J.F. (2004). An Equivalent Random Method with hyper-exponential service. *Performance Evaluation, 57*(3), 409-422.

[53] Stewart, W.J. (2009). *Probability, Markov Chains, Queues, and Simulation: The Mathematical Basis of Performance Modeling.* Princeton, NJ: Princeton University Press.

[54] Taylor, P.G. (2011). Insensitivity in stochastic models. In R.J. Boucherie and N.M. van Dijk

(eds.), *Queueing Networks: A Fundamental Approach* (pp. 121-140). New York: Springer.

[55] Tijms, H.C. (2003). *A First Course in Stochastic Models*. Chichester, U.K.: Wiley.

[56] Van Dijk, N.M. (1988). On Jackson's product form with 'jump-over' blocking. *Operations Research Letters, 7*(5), 233-235.

[57] Van Dijk, N.M. (1993). *Queueing Networks and Product Forms: A Systems Approach*. Chichester, U.K.: Wiley.

[58] Van Dijk, N.M. (2011). On practical product form characterizations. In R.J. Boucherie and N.M. van Dijk (eds.), *Queueing Networks: A Fundamental Approach* (pp. 1-83). New York: Springer.

[59] Van Dijk, N.M., & Kortbeek, N. (2009). Erlang loss bounds for OT-ICU systems. *Queueing Systems, 63*(1), 253-289.

[60] Van Dijk, N.M., & Schilstra, B. (2021). On two product form modifications for finite overflow systems. *To appear in Annals of Operations Research.*

[61] Van Dijk, N.M., & Van der Sluis, E. (2009). Call packing bound for overflow loss systems. *Performance Evaluation, 66*(1), 1-20.

[62] Wilkinson, R.I. (1956). Theories for toll traffic engineering in the USA. *Bell System Technical Journal, 35*(2), 421-514.

[63] Zonderland, M.E., Boucherie, R.J., Carter, M.W., & Stanford, D.A. (2015). Modeling the effect of short stay units on patient admissions. *Operations Research for Health Care, 5*, 21-27.

[64] Zorginstituut Nederland (2016). *Kwaliteitsstandaard Organisatie van Intensive Care*. Produced by the Adviescommissie Kwaliteit. Retrieved from: https://nvic.nl/richtlijnen/organisatie-van-intensive-care-kwaliteitsstandaard-2016 (April 26th, 2021).