Performance modeling : structured Markov chains, optical grids and switches

Universiteit Antwerpen
Faculteit Wetenschappen
Department Wiskunde en Informatica

Performance modeling: structured Markov chains, optical grids and switches

Prestatiemodellering: gestructureerde Markov-ketens, optische grids en schakelaars

Dissertation for the degree of doctor in science: informatics
at the University of Antwerp to be defended by

Juan Fernando PÉREZ BERNAL

Supervisor: Prof. dr. Benny Van Houdt

Antwerp, 2010

To Julián, Elsy and Marco.

Acknowledgments

It is not a simple task to thank everyone who has somehow contributed to this process, as such contributions occur in the most varied ways. I will try to do my best and I offer my apologies in advance to those that are unintentionally left out. First and foremost I want to thank my supervisor prof. dr. Benny Van Houdt, who gave me the chance to work with him. During these years he was always been able to challenge me with interesting questions, as well as to offer his never-trivial insights. Apart from that, working with Benny has been very entertaining, and I have gotten used to him always having a funny remark at hand. Finally, I want to thank him for the very real support he offered me all the time, from picking me up at the airport the day of my arrival to carefully reading all the material in this thesis. I want to express my gratitude to prof. dr. Chris Blondia, who gave me the opportunity to join the PATS research group. I also thank Chris Develder for the smooth communication we had while working on one of the grid models that are now part of this thesis.

The first half of my stay in the university was greatly eased by the helpful Jeroen V. and Joris, who were always ready to answer one of my many practical, and not so practical, questions. They also provided a very nice atmosphere in the office, including those now-long-gone musical fridays. The rhythm was extensive to the hallway, were Joke used to pass lightning fast to the printer, many times stopping by to say hello or to share a though about some FDL-related issue. During the second part of my stay I was very much amused by the chats with Jeroen A. and Gino. Our discussion topics were very broad, with enough room to talk about the perfect sense that makes living in an eleven-dimension world or the basic steps of a mazurka. Apart from these activities, there was time for serious stuff, which very much relied on the laborious Wim, aka Gulielmus, who designed the famous Tour de Fun and the award-winning Kolonisten van Kantoor. And how could I forget Dessie, whose strong opinions made perfect sense under the effect of some traditional rakia. I am very indebted to all these guys for many good moments, which, in the end, made my life at the UA very enjoyable. With some of them, there was enough enthusiasm to out-of-the-office activities, which I hope they enjoyed as much as I did. And, since Murray has never been much of a talker, I really want to thank Daniel for bringing the office back to life.

There is still another group of friends with whom, thanks to the almighty Internet, I was able not only to stay in touch, but to talk about almost everything during our multi-hours-long conversations. First in my list, as expected, is Adriana, whose support was crucial for my survival, especially during my first year. Her critical opinions still

help me to find my many mistakes, being these in writing or behaving. My two closest virtual friends, as weird as that might sound, have been Andrea and Camila, who made me feel like home by sharing so much of their lives, and listening to my never-ending list of complaints. I also want to thank the CGA (Colombian Gang in Antwerp), for the many amusing evenings, that included arepas, salsa and rum. Thanks Pablo, César, Juan Carlos and Ana Maria.

Mention apart is deserved by my brother and parents. I especially thank my parents, Elsy and Marco, for never giving up on me, and for teaching me so many things about life, each from her/his very own point of view. And to Julián, my brother, I thank him for so many sunday-morning chats, for suffering my parents during my absence, and for letting our friendship grow despite the distance.

A todos, muchas muchas gracias.

Contents

List of Tables

List of Figures

List of Acronyms

CDF	Cumulative Distribution Function
CPH	Continuous Phase-Type
CTMC	Continuous-time Markov chain
DPH	Discrete Phase-Type
DTMC	Discrete-time Markov chain
FDL	Discrete-time Markov chain
IAT	Inter-arrival time
MAP	Markovian Arrival Process
MC	Markov chain
OBS	Optical Burst Switching
OPS	Optical Packet Switching
PH	Phase-Type
QBD	Quasi-Birth-and-Death
SMC	Structured Markov chain
WC	Wavelength Converter
WDM	Wavelength Division Multiplexing

Introduction

It is hard to imagine a technological advance that has impacted our daily lives in the last twenty years more than computer communications. It has been a very short period in which we have gone from letters on paper and expensive long-distance phone calls to communicate instantly by email or video-conference at almost no cost. Now we can instantly share information with friends living on the most distant places, or buy the latest hit from our favorite musician while sitting on our couch, or even check the last updates on an event happening in another country, receiving feeds not only from newspapers or magazines, but also from the attendees. Moreover, computer communication has enabled many research efforts to use remote computing resources that make possible the analysis of huge amounts of data. And the revolution is still going on, with new applications popping up every week, and new devices being developed to take you closer to the information network.

A central role in this revolution has been played by the networks and computing resources deployed around the world. A computer network is typically made of many components, the deployment of which is very expensive. Therefore, the design of these components, and of the network as a whole, is a process on which much attention must be paid. During this process many non-trivial questions arise, such as how much transmission capacity to install between a pair of nodes in the network, or how many servers must be allocated in each computing site in order to provide a certain quality of service. Moreover, these questions must be considered in light of the significant fluctuations in the traffic that these networks must face. As a result, modeling tools that consider the probabilistic nature of the traffic, such as queueing theory, are of special relevance in the design and analysis of computer networks. This is exactly the subject of this thesis, where we consider various systems arising in computer communication for which we propose models to assess their performance. In addition, we pay special attention to the generality of the models, so that general traffic conditions can be analyzed. Also, we are particularly concerned with the efficiency with which these models are able to compute the performance measures for a specific system. This is of particular relevance since, in many cases, it is necessary to evaluate a system under very different conditions, which means that a large number of scenarios must be considered.

As can be deduced from the title, out of the immensely large field of computer communications, in this thesis we focus on three topics: structured Markov chains, optical grids and switches. Structured Markov chains are a powerful modeling tool that allows the analysis of very general systems subject to a stochastic environment. This type of

Markov chains is characterized by a block transition matrix with a repeating structure that can be exploited to efficiently compute its stationary probability vector. In particular, we have considered the case where the transition matrix's blocks possess some inner structure that allows an even faster computation of the stationary vector. This structure arises, or can be induced, in the analysis of various queueing models of communication systems. The second main topic in this work is the analysis of optical grids, which are networks connecting users to computing sites that are themselves interconnected. The distinctive feature of a grid is that the users do not care about which site ends up processing its request. Therefore, a site that does not have available capacity to process an incoming request, can rely on other sites in the network to process the request and send the results back to the user. This type of network originally arose in research efforts in fields such as astrophysics, particle physics, chemistry or biomedicine, where huge amounts of information must be analyzed. The last topic of this work is the modeling and analysis of optical switching technologies. An optical switch allows an incoming signal to be processed mostly in the optical domain, avoiding the opto-electronic conversions required when an optical signal enters an electronic switch. Therefore, optical switching offers a solution for the backbone network, where the switches must keep up with the ever-increasing capacity of optical fibers. Our main interest lies on the analysis of the contention resolution strategies in an optical switch. In this type of switch, contention arises when two or more packets attempt transmission through the same output port using the same wavelength. There are two main alternatives for contention resolution in the optical domain: optical buffering and wavelength conversion. We consider three different switch architectures that implement some form of these contention resolution alternatives, and analyze the effect of the design parameters and the traffic conditions on the switch performance.

In agreement with the three main topics mentioned, this thesis is divided in three parts. Each of these parts comprises two or three chapters, a brief description of which is given in the next section. In addition, the appendices give a brief account of some topics that are relevant in some or all of the chapters in the thesis. The chapters, as well as the appendices, are mostly self-contained and can therefore be approached in any order. A small introduction has been included at the beginning of each part, which provides more details on each of the topics and highlights related works in the area as well as our contributions.

Summary and Overview

The first part of this thesis deals with structured Markov chains. As already mentioned, these Markov chains are characterized by a transition matrix with a block structure that can be exploited to speed up the computation of the stationary probability vector. However, these chains may suffer from the curse of dimensionality, which in this case is reflected in an exponential increase of the block size. Therefore, there is an interest in exploiting the inner structure of the transition matrix's blocks. A particular class of structured Markov chains is known as Quasi-Birth-and-Death (QBD) Markov chains, the transition matrix of which has a block structure that allows the stationary probability

vector to be compactly expressed in terms of a boundary vector and a *rate* matrix. For this class, a specific structure of the blocks has been recently pointed out in the literature, which we refer to as restricted-downward/upward transitions. In Chapter 1, we analyze this structure by applying a censoring argument on the transitions of the chain, and splitting the computation of the rate matrix in two steps. As a result, the total time to compute the rate matrix is significantly reduced, in many cases by an order of magnitude, compared to the general approaches. We extend this approach in Chapter 2 to analyze M/G/1-type Markov chains, which is a more general class of structured Markov chains. In this case, one must also determine a matrix, called G, but the computation of the stationary probability vector from this matrix is more involved than for QBD chains. Therefore, we not only exploit the restricted-transition structure to obtain the G matrix, but also to compute the stationary vector once G is found. Furthermore, additional gains can be made by considering an even more specific structure of the blocks, which arises in the analysis of batch queues. In fact, the restricted-transitions structure is not only computationally appealing, but it arises, or can be induced, in modeling various queueing systems such as priority queues, overflow queues or general BMAP/PH/1 queues, among others.

The second part of this thesis concerns the analysis of optical grids, which we already described as networks where the computing sites can rely on each other to carry out the requests submitted by the users. Therefore, optical grids differ from more traditional networks as a request does not have a predefined path to follow along the network. To analyze this type of network, we have considered two cases: the first contemplates a grid arranged in a ring topology, while the second focuses on a network with a large number of stations. The case of the ring topology is treated in Chapter 3, where we develop two different methods to approximate the complex arrival process at each station in the network. Using these methods, each site can be analyzed separately to evaluate the network performance. Both methods provide very good approximations for the fraction of jobs processed locally and the inter-node traffic, when the total load of the network is at most 90%. On the other hand, Chapter 4 considers a grid made of a very large number of sites, for which we propose a mean field model that is exact when the number of sites tends to infinity. In this case the job routing does not depend on the topology, but on the state of the different stations. This allows a request that cannot be processed locally to be sent, for instance, to the station with the largest number of idle servers. The model has shown to be very accurate to approximate the fraction of traffic processed locally and the traffic among the stations. In designing an optical grid, the traffic among the sites is considered the main performance measure.

The last part focuses on the modeling and analysis of contention resolution strategies in optical switches. As stated above, we consider two technologies for contention resolution: optical buffering and wavelength conversion. Given the lack of optical random access memory, optical buffering is implemented by means of Fiber Delay Lines (FDLs), which only provide a small discrete set of delay values. On the other hand, and thanks to wavelength division multiplexing (WDM), optical fibers carry many signals simultaneously using different wavelengths. Therefore, if a packet requires transmission through a busy wavelength, it can be translated to an idle wavelength using a wavelength converter

(WC). These converters may provide either full- or limited-range conversion capabilities, depending on whether they are able to translate a signal to any or to a specific set of wavelengths, respectively. As can be seen, the alternatives for contention resolution in optical switches differ significantly from those available in traditional switches. To analyze the effect of these contention resolution alternatives on the performance of the switch, we have considered three different switch architectures. In Chapter 5, we analyze a bufferless switch endowed with a *centralized* pool of WCs, i.e., the WCs are shared among all the ports. This architecture has the potential of reducing the total number of WCs to achieve a certain performance, but requires a more complex switching matrix. Here we look at the case where the number of wavelengths is very large, which is relevant given the advances in WDM. Therefore, we propose a mean field model that is exact when the number of wavelengths tends to infinity, and can be used to approximate the performance of a switch with a large number of these. Chapter 6 considers a second architecture, where the switch has a pool of FDLs and full-range WCs *per port*. Here we also consider the case where the number of wavelengths per port is large, and therefore a mean field model is proposed to analyze the effect of various traffic parameters on the switch performance. In the models in these two chapters, we pay special attention to the minimum number of WCs required to attain zero losses in the infinite-wavelength case. The relevance of this measure arises from the need to economically dimension the switch's conversion capabilities. Chapter 7 ends this part by studying a switch with a pool of FDLs and *limited-range* WCs *per output port*. The limited-range case imposes great modeling difficulties due to the complex interaction among adjacent wavelengths. For this switch architecture we propose an approximation based on a Markovian model, which performs well when the conversion range is small. The model also highlights the important performance gains that can be obtained by combining FDLs and WCs, even when the conversion range of the latter is very limited.

Part I

Structured Markov Chains with Restricted Transitions

Structured Markov Chains with Restricted Transitions

Structured Markov chains (SMCs) have been central to the development of computational probability during the last forty years. The distinctive characteristic of a Markov chain (MC) within this class is that its transition matrix possess a structure that can be exploited to efficiently compute its stationary probability vector. The complete body of SMCs can be classified in subclasses, each one defining a particular structure for the transition matrix. Among the first structures to be identified is the one that characterizes the set of Quasi-Birth-and-Death (QBD) MCs, introduced by Wallace [119]. These chains are a generalization of simple birth-and-death processes where the addition of a second dimension, called the *phase*, allows the representation of more general systems. The first dimension, referred to as the *level*, takes values on the non-negative integers and is allowed to increase or decrease its value at most by one in a single transition epoch, as in the well-known birth-and-death process. The second dimension typically (phase) describes a random environment or the state of the arrival and service processes in the case of queueing systems. The main feature of a QBD process is that its stationary probability vector, if it exists, has a matrix-geometric form, such that it can be expressed as a function of a boundary probability vector and a *rate* matrix R. A thorough analysis of these chains was carried out by Neuts in [91], where the MCs of the GI/M/1 type were also introduced. A GI/M/1-type MC can be seen as a QBD where the *level* is allowed to decrease its value without restriction in a single transition, as long as it remains in the non-negative integers. The stationary probability vector of this class of MCs also has the matrix-geometric property. A similar generalization of QBDs can be obtained by relaxing the restriction on the upward transitions, allowing the level to increase its value by more than one at each transition epoch. This relaxation results in the class known as M/G/1-type MCs, which were introduced and studied in detail by Neuts in [92]. This class however lacks the matrix-geometric property, but the computation of its stationary probability vector can be achieved by means of Ramaswami's formula [97]. There are many other classes of SMCs, such as tree-structured MCs or non-skip-free MCs, but our focus will be on QBD, GI/M/1- and M/G/1-type MCs. For more on these and other structures the reader is referred to [18, 76, 91, 92].

Due to the matrix-geometric property, a crucial step in finding the stationary probability vector of a QBD or GI/M/1-type MC is the computation of the rate matrix R.

This is done by solving a nonlinear (quadratic in the case of QBD MCs) matrix equation. In the case of M/G/1-type MCs, the so-called matrix G underlies the computation of the stationary probability vector, and it is also found as the solution of a nonlinear matrix equation. Many iterative algorithms have been proposed to solve these nonlinear equations, including functional iterations [91,92], logarithmic reduction [75], cyclic reduction [19], the invariant subspace method [5], and others. The ability of these algorithms to solve the matrix equations depends on many aspects, one of the most relevant being the size m of the phase space (the set of values that the phase variable may take). The R and G matrices are of size m, as are the sub-matrices of the transition matrix involved in their calculation (also called *blocks*). As any MC, SMCs suffer from the curse of dimensionality, which implies that m may easily become very large. In this case even the most efficient algorithms, such as Cyclic Reduction [19] or Logarithmic Reduction [75], require long computation times to solve the matrix equations.

One way to deal with the dimensionality problem is to consider the specific structure of the blocks of the SMC, and exploit it to reduce the computation times. For instance, when these blocks are triangular it is possible to solve the matrix equations significantly faster than in the general case, as has been shown in [117, 118]. Also, if the blocks are themselves block-circulant, the solution of the matrix equation can also be accelerated [37]. In this thesis we will consider the structure that arises if an increase (resp. decrease) in the value of the *level*, i.e., an upward (resp. downward) transition, is restricted to occur in (resp. lead to) a specific subset of the phase space. In a QBD MC, if this subset is of size one, then either the matrix G or R can be explicitly expressed without resorting to the iterative algorithms mentioned before [76]. For an M/G/1-type (resp. GI/M/1-type) MC, the G (resp. R) matrix can also be found explicitly if the downward (resp. upward) transitions can only lead to (resp. occur in) an unitary subset of the phase space. In this thesis we consider the more general case where this subset is small compared to the block size m, but not unitary. This structure will be referred to as restricted (downward or upward) transitions.

In the next two chapters we will show how the restricted-transitions structure can be exploited to reduce the time required to compute the stationary probability vector of an SMC. We start in Chapter 1 with the case of QBD MCs, and we illustrate how to speed up the computation of the matrix R (resp. G) when the chain has restricted upward (resp. downward) transitions. Once either R or G has been found, it is straightforward to compute the stationary vector of this MC thanks to the matrix geometric property. To briefly describe the methodology, consider the case of restricted downward transitions and let $\mathcal{S}^+$ be the subset of the phase space toward which these transitions lead. To determine G we start by defining a new process by observing the QBD process when the phase variable is in $\mathcal{S}^+$. The new process is of the M/G/1 type, but the size of its blocks is equal to the cardinality of $\mathcal{S}^+$. Therefore, we can use Cyclic Reduction [19] to find its associated matrix G_+, which will be shown to be a sub-matrix of the matrix G of the QBD. After finding G_+ we obtain the remaining entries of G by solving a Sylvester matrix equation. If the QBD has restricted upward transitions, the steps to find R are similar, but in this case the censored process is of the GI/M/1 type.

Our methodology can be extended to the cases of GI/M/1- and M/G/1-type MCs.

In Chapter 2 we consider the extension to M/G/1-type MCs with restricted downward transitions, which can be easily translated to GI/M/1-type MCs with restricted upward transitions. For this latter case it is enough to compute R efficiently as this MC also possess the matrix-geometric property. However, in Chapter 2 we not only extend the methodology to compute the matrix G, but we also make use of the restricted-transitions structure to expedite the computation of the stationary probability vector of an M/G/1-type MC, for which the matrix-geometric property does not hold.

Recently, some attention has been given to the structure that we have referred to as restricted transitions. For the case of QBD MCs, Grassmann and Tavakoli [50] have exploited this structure to accelerate the time per iteration of the linearly-convergent U-based method [74] by means of an UL decomposition. Also, in [25] the authors introduce a model to analyze a wireless relay node, which turns out to be a QBD MC with restricted downward transitions. In both cases, the method of solution relies on a specific functional iteration algorithm, which has the drawback of being linearly convergent [18]. As mentioned before, our approach does not rely on a particular algorithm for the solution of the nonlinear matrix equation, but it defines a new process of smaller size that can be solved using any algorithm, including the quadratically-convergent Cyclic Reduction [19]. On the other hand, and to the best of our knowledge, M/G/1-type MCs with restricted transitions have only been treated in [30]. There the authors exploit the referred structure to speed up the computation of G by using two specific iterative algorithms: a functional iteration and a sub-Newton method. In contrast, our approach is able to make use of the most efficient algorithms as the definition of the censored process is independent of the solution method. Moreover, in [30] an additional restriction is imposed on the chain (see Remark 2.2 on page 34), which reduces the applicability of their method. For instance, the MC to model the BMAP[2]/PH[2]/1 preemptive priority queue is of the M/G/1-type and has the restricted-downward-transitions property. However, this queue cannot be analyzed with the methods introduced in [30] because of these additional restrictions.

An SMC with restricted transitions is not only computationally appealing, but there are many applications where this property arises naturally or can be induced by adequate modeling. In fact, in the next chapters we will examine various examples where this structure arises, such as the priority queue with two customer classes, with and without batch arrivals, or an overflow queueing system [83] consisting of two queues, where the second queue receives arrivals only when the buffer of the first queue is full. Also, we will show that by adequately modeling the service time distribution, it is possible to induce the restricted-downward-transitions structure in the MCs that model the MAP/PH/1 and BMAP/PH/1 queues (for details on MAPs and PH distributions see Appendix A.1). These examples are used to illustrate the relevance of our methodology, as well as the computational gains obtained by exploiting the referred structure, compared with solving the original system. As stated before, this structure has also been considered in [50] and [25], where the authors propose algorithms to speed up the solution of the matrix equation. We compare these methods with our approach and show that ours outperforms the others in many cases. We will also show that, for the priority queue with batch arrivals, our approach is not only faster than solving the original M/G/1-type MC, but it is also faster than other methods previously proposed to find the queue-length distribution

of this particular queue [129]. In conclusion, the methods to be introduced in the next chapters are able to significantly reduce the computation times required to determine the stationary probability vector of QBD, GI/M/1- and M/G/1-type MCs, when the blocks of these SMCs possess the restricted-transitions structure. These methods have been implemented as part of a software tool that will be made available online.

Chapter 1

Quasi-Birth-and-Death processes with restricted transitions

A discrete-time QBD MC is a two-dimensional process $\{(N_n, X_n), n \geq 0\}$, where N_n is called the level variable and takes values on $\mathbb{N}$. The phase variable X_n takes values on the set $\{1, 2, \ldots, m_0\}$ or $\{1, 2, \ldots, m\}$ depending on whether the level is equal to or greater than 0. The level variable can only increase or decrease its value by one at each time epoch and the transition probabilities are level-independent. Therefore the QBD MC has a transition matrix P of the form

$$P = \begin{bmatrix} B_1 & B_2 & 0 & 0 & \ldots \\ B_0 & A_1 & A_2 & 0 & \ldots \\ 0 & A_0 & A_1 & A_2 & \ldots \\ \vdots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}, \tag{1.1}$$

where B_1 and A_1 are square matrices of size m_0 and m, respectively. The matrices B_1 and B_2 hold the transition probabilities from level 0 to levels 0 and 1, respectively, while the matrix B_0 contains the transition probabilities from level 1 to level 0. Similarly, the matrices A_0, A_1 and A_2 carry the transition probabilities from level i to levels $i-1$, i and $i+1$, respectively, for $i > 0$. The key when computing the stationary probability vector $\pi = [\pi_0, \pi_1, \pi_2, \ldots]$ of P, if it exists, is to find the minimal nonnegative solution R of the matrix equation

$$R = A_2 + RA_1 + R^2 A_0. \tag{1.2}$$

The vectors π_i can then be computed as $\pi_i = \pi_1 R^i$, for $i > 1$, where $[\pi_0, \pi_1]$ is the solution of the boundary equation

$$[\pi_0, \pi_1] \begin{bmatrix} B_1 & B_2 \\ B_0 & A_1 + RA_0 \end{bmatrix} = [\pi_0, \pi_1].$$

Another way to find the matrix R is from $R = A_1(I - A_0 - A_1G)^{-1}$, where G is the minimal nonnegative solution of the matrix equation

$$G = A_0 + A_1G + A_0G^2. \tag{1.3}$$

Many iterative algorithms have been developed to solve equations (1.2) and (1.3), including quadratically-convergent algorithms such as Cyclic Reduction (CR) [19] and Logarithmic Reduction [75]. The method introduced in this chapter aims at computing either the matrix R or G, from which the stationary probability vector can be obtained. Moreover, our approach is actually independent of the behavior of the QBD near the boundary at level 0. Therefore a more general boundary behavior can be assumed as long as the QBD shows a repeating structure (matrices A_0, A_1 and A_2) from a given level onward.

As discussed in the introduction to this Part, we will show how to speed up the computation of the matrix R or G, when the QBD MC has restricted upward or downward transitions, respectively. These types of transitions cause the blocks of the QBD to have a special structure, which will be illustrated in Section 1.1. We next provide a detailed explanation of how this structure can be exploited to reduce the times required to compute R or G. The case of restricted downward transitions is treated in Section 1.2, while restricted upward transitions are the topic of Section 1.3. Section 1.2 also includes some special cases where additional structure can be exploited to further reduce the computation times.

To illustrate our approach, we consider four different examples in Section 1.4, including a priority queue with two customer classes and an overflow queueing system. In addition, we show how the restricted-downward-transitions structure can be induced in the QBD MC that describes a general MAP/PH/1 queue. This is achieved by defining a (slightly larger) representation of the service-time distribution, forcing the state of its underlying process to re-start in a specific phase, which occurs whenever there is a service completion (downward transition). Our last example is based on the model introduced in [25] to evaluate the packet delay in a wireless relay node, which falls within the set of QBD MCs with restricted downward transitions. These examples are also used to illustrate the computational gains obtained by using the approach introduced here, compared to the traditional methods and to the methods proposed in [50] and [25].

1.1 QBDs with restricted transitions

In this chapter we consider two special cases where the structure of the matrices A_0 and A_2 can be exploited to speed up the computation of the matrix G or R. We consider a partition of the set $\{1, \ldots, m\}$ into two sets: $\mathcal{S}^+$ containing the first r phases, and $\mathcal{S}^-$ containing the remaining $m - r$ phases. Using this partition the matrices A_i, for $i = \{0, 1, 2\}$, can be written as

$$A_i = \begin{bmatrix} A_i^{++} & A_i^{+-} \\ A_i^{-+} & A_i^{--} \end{bmatrix}, \tag{1.4}$$

where A_i^{++} and A_i^{--} are square matrices of size r and $m - r$, respectively. We assume that the MC is irreducible, and therefore the matrices A_i^{++} and A_i^{--} are sub-stochastic

and the inverses $(I - A_i^{++})^{-1}$ and $(I - A_i^{--})^{-1}$ exist. In Section 1.2 we consider the case where downward transitions can only lead to a state with phase in $\mathcal{S}^+$, hence the matrix A_0 has only $r \ll m$ nonzero columns such that it can be written as

$$A_0 = \begin{bmatrix} A_0^{++} & 0 \\ A_0^{-+} & 0 \end{bmatrix}. \tag{1.5}$$

When the set $\mathcal{S}^+$ contains only one phase the matrix G can be computed explicitly without the need of resorting to iterative algorithms [76]. This particular case has also been exploited to compute performance measures in an efficient manner without computing all the terms of the vector π [32]. In this chapter we consider the more general case where the cardinality of $\mathcal{S}^+$ is greater than one, meaning that the matrix G is not known explicitly from the parameters of the QBD.

The analogous case where upward transitions only occur in a state with phase in $\mathcal{S}^+$ is treated in Section 1.3. In this case the matrix A_2 has only $r \ll m$ nonzero rows, i.e.,

$$A_2 = \begin{bmatrix} A_2^{++} & A_2^{+-} \\ 0 & 0 \end{bmatrix}. \tag{1.6}$$

This structure was analyzed by Grassmann and Tavakoli in [50], where it was exploited to reduce the time per iteration in the so-called U-algorithm [74], which computes a matrix U such that $R = A_2(I - U)^{-1}$. The algorithm starts with $U_0 = A_1$ and iteratively computes $U_{k+1} = A_1 + A_2(I - U_k)^{-1}A_0$, such that the iterates converge to the actual value of the matrix U. Even though the approach proposed in [50] provides an important computational gain per iteration, the number of iterations required may be large since this is a linearly-convergent algorithm [18]. In Section 1.4 we consider an example with the structure described by (1.6) and compare the performance of our approach with the one proposed in [50]. This method can also be adapted to the case where the matrix A_0 has the form in (1.5). In the next section we briefly review the definition of M/G/1- and GI/M/1-type MCs, as these are central for the methodology to be introduced in sections 1.2 and 1.3.

1.1.1 Markov chains of the M/G/1 and GI/M/1 type

An M/G/1-type MC [92] can be seen as a generalization of a QBD MC, where the level is allowed to increase its value by more than one in a single transition. Therefore, the transition matrix $\bar{P}$ of an M/G/1-type MC is of the form

$$\bar{P} = \begin{bmatrix} \bar{B}_0 & \bar{B}_1 & \bar{B}_2 & \bar{B}_3 & \cdots \\ \bar{A}_0 & \bar{A}_1 & \bar{A}_2 & \bar{A}_3 & \cdots \\ & \bar{A}_0 & \bar{A}_1 & \bar{A}_2 & \cdots \\ & & \bar{A}_0 & \bar{A}_1 & \cdots \\ 0 & & & \ddots & \ddots \end{bmatrix},$$

where $(\bar{A}_i)_{i\geq 0}$ and $(\bar{B}_i)_{i\geq 0}$ are nonnegative matrices in $\mathbb{R}^{b\times b}$ such that $\sum_{i=0}^{+\infty} \bar{A}_i$ and $\sum_{i=0}^{+\infty} \bar{B}_i$ are stochastic. A numerically stable method to find the stationary probability

vector of this MC is Ramaswami's formula [97], which depends on the matrix $\bar{G}$, that is the minimal non-negative solution of

$$\bar{G} = \sum_{i=0}^{\infty} \bar{A}_i \bar{G}^i. \tag{1.7}$$

The quadratically-convergent CR algorithm can also be applied to solve this equation.

On the other hand, a GI/M/1-type MC [91] can be seen as a QBD where the chain is allowed to decrease several levels in a single transition. The transition matrix of this MC is thus given by

$$\hat{P} = \begin{bmatrix} \hat{B}_0 & \hat{A}_0 & & & 0 \\ \hat{B}_1 & \hat{A}_1 & \hat{A}_0 & & \\ \hat{B}_2 & \hat{A}_2 & \hat{A}_1 & \hat{A}_0 & \\ \hat{B}_3 & \hat{A}_3 & \hat{A}_2 & \hat{A}_1 & \dots \\ \vdots & \vdots & \ddots & \ddots & \ddots \end{bmatrix},$$

where $(\hat{A}_i)_{i\geq 0}$ and $(\hat{B}_i)_{i\geq 0}$ are nonnegative matrices in $\mathbb{R}^{b\times b}$ such that $\sum_{i=0}^{n} \hat{A}_i + \hat{B}_n$ is stochastic for all $n \geq 0$. In this case the stationary probability vector can be computed as $\pi_i = \pi_0 \hat{R}^i$, where $\hat{R}$ is the minimal non-negative solution to

$$\hat{R} = \sum_{i=0}^{\infty} \hat{R}^i \hat{A}_i. \tag{1.8}$$

To solve this equation we first compute the blocks of the *dual* process, which is of the M/G/1 type, as this allows us to use the quadratically-convergent CR algorithm. There are two different duals that can be used for this purpose. A brief description of both is included in Appendix A.5. Here we have assumed that the boundary level has the same size as all the other levels in both the M/G/1- and GI/M/1-type MCs. A more general boundary can be assumed since our results are related to the behavior of the MCs away from the boundary, which is described by the $(\bar{A}_i)_{i\geq 0}$ or the $(\hat{A}_i)_{i\geq 0}$ matrices.

1.2 QBDs with restricted downward transitions

In this section we describe how the special structure of the matrix A_0 can be exploited to compute the matrix G. Consider the case where the matrix A_0 has only $r \ll m$ nonzero columns as shown in Equation (1.5). The (i, j)-th entry of the matrix G holds the probability that the first visit to level $k-1$ occurs by visiting state $(k-1, j)$, starting from state (k, i), for $k > 1$ [76]. In a QBD MC a path that takes the chain from level k to level $k-1$ must end with a downward transition from level k to level $k-1$. Since the downward transitions can only trigger the phase to one of the first r states of any level, the G matrix has the structure

$$G = \begin{bmatrix} G_+ & 0 \\ G_0 & 0 \end{bmatrix},$$

where G_+ (resp. G_0) is an $r \times r$ (resp. $(m-r) \times r$) matrix. The computation of G_+ and G_0 will be split in two steps such that for $r \ll m$ the total computation time can be significantly reduced.

1.2.1 Computing G_+

To compute G_+ we define a new process by observing the QBD MC only when the phase variable is in the set $\mathcal{S}^+$. In the original process any transition to a lower level triggers the phase to a state in $\mathcal{S}^+$, therefore the new process can only move one level down at each transition. On the other hand, the original process can move several levels upward while the phase is in $\mathcal{S}^-$, i.e., between two visits to $\mathcal{S}^+$. Therefore the new process can move several levels up in one transition, but only one level down. Hence, the new process is of the M/G/1 type and its behavior away from the boundary is characterized by the set of $r \times r$ matrices $(\bar{A}_i)_{i\geq 0}$. The minimal nonnegative solution $\bar{G}$ of Equation (1.7) is actually equal to the matrix G_+. This follows from the definition of the (i,j)-th entry of $\bar{G}$ as the first passage probability to the state $(k-1,j)$ starting from state (k,i) in the new process, and the fact that in the original process the downward transitions can only lead to $\mathcal{S}^+$. Hence, to compute the matrix G_+ we first need to determine the $r \times r$ blocks $(\bar{A}_i)_{i\geq 0}$ and then solve Equation (1.7).

To specify the blocks $(\bar{A}_i)_{i\geq 0}$ let the (i,j)-th entry of the $(m-r)\times r$ matrix K_l hold the probability that, given that the original process starts in state (k,i), with $i \in \mathcal{S}^-$, its first transition to a state with phase in $\mathcal{S}^+$ occurs to the state $(k+l,j)$, for $j \in \mathcal{S}^+$, $k > 1$ and $l \in \{-1,0,1,...\}$. Hence, the matrices $(K_i)_{i\geq -1}$ are given by

$$\begin{array}{ll} K_{-1} & = (I - A_1^{--})^{-1} A_0^{-+}, \\ K_0 & = (I - A_1^{--})^{-1}(A_1^{-+} + A_2^{--} K_{-1}), \\ K_1 & = (I - A_1^{--})^{-1}(A_2^{-+} + A_2^{--} K_0), \\ K_i & = (I - A_1^{--})^{-1} A_2^{--} K_{i-1}, \quad i \geq 2. \end{array} \tag{1.9}$$

To define K_{-1} we observe that the chain starts in level k and spends some time in the states of this level with phase in $\mathcal{S}^-$. Afterward the chain has to move to a state $(k-1,j)$, with $j \in \mathcal{S}^+$. The only other possible state that the chain could visit after its sojourn in level k, avoiding states with phase in $\mathcal{S}^+$, is to move to a state in level $k+1$ and phase in $\mathcal{S}^-$. However, for the chain to visit level $k-1$ it first has to go back from level $k+1$ to level k, and this can only be done through a state with phase in $\mathcal{S}^+$. Therefore, this path is not possible if the first state with phase in $\mathcal{S}^+$ to be visited must be part of level $k-1$. The definition of the other matrices can be understood in a similar manner. Now we can define the blocks $(\bar{A}_i)_{i\geq 0}$ in terms of the matrices $(K_i)_{i\geq -1}$ as

$$\begin{array}{ll} \bar{A}_0 & = A_0^{++} + A_1^{+-} K_{-1}, \\ \bar{A}_1 & = A_1^{++} + A_1^{+-} K_0 + A_2^{+-} K_{-1}, \\ \bar{A}_2 & = A_2^{++} + A_1^{+-} K_1 + A_2^{+-} K_0, \\ \bar{A}_i & = A_1^{+-} K_{i-1} + A_2^{+-} K_{i-2}, \quad i \geq 3. \end{array} \tag{1.10}$$

To define $\bar{A}_0$ we see that the transition from a state (k,i) to a state $(k-1,j)$, with $i,j \in \mathcal{S}^+$, can only occur in two ways: either the chain goes directly to $(k-1,j)$ with transition matrix A_0^{++}; or it moves first to a state in level k with phase in $\mathcal{S}^-$ and, after a sojourn in these states, it moves downward avoiding other states in $\mathcal{S}^+$ (with transition matrix $A_1^{+-}K_{-1}$). A transition to level $k+1$ is not allowed since the chain cannot return to $k-1$ without passing through a state in level k with phase in $\mathcal{S}^+$. The other matrices

can be defined similarly. Notice, to compute the matrices $\bar{A}_i$ it suffices to store two K_i matrices at a time. The $r \times r$ matrices $\bar{A}_i$ are sequentially computed from $i = 0$ to c, where c is the smallest positive integer such that $\sum_{i=0}^{c} \bar{A}_i e > (1-\epsilon)e$, with e a column vector of ones and $\epsilon = 10^{-14}$. These blocks can then be used to compute the matrix G_+ using the CR algorithm [19].

1.2.2 Computing G_0

Given the structure of the matrices A_0 and G we can rewrite Equation (1.3) as

$$\begin{bmatrix} G_+ & 0 \\ G_0 & 0 \end{bmatrix} = \begin{bmatrix} A_0^{++} & 0 \\ A_0^{-+} & 0 \end{bmatrix} + \begin{bmatrix} A_1^{++} & A_1^{+-} \\ A_1^{-+} & A_1^{--} \end{bmatrix} \begin{bmatrix} G_+ & 0 \\ G_0 & 0 \end{bmatrix} + \begin{bmatrix} A_2^{++} & A_2^{+-} \\ A_2^{-+} & A_2^{--} \end{bmatrix} \begin{bmatrix} G_+^2 & 0 \\ G_0 G_+ & 0 \end{bmatrix}. \quad (1.11)$$

Extracting the lower-left block we find

$$G_0 - (I - A_1^{--})^{-1} A_2^{--} G_0 G_+ = (I - A_1^{--})^{-1} (A_0^{-+} + A_1^{-+} G_+ + A_2^{-+} G_+^2), \quad (1.12)$$

which is a Sylvester matrix equation [44, 47] of the type $AXB + X = E$, that can be solved in $O((m-r)^3)$ time with the Hessenberg-Schur method proposed in [47]. A brief description of this method is included in Appendix A.4 together with a discussion on some additional considerations that influence the computation time of G_0. Next, we consider two special cases where additional restrictions on the transition probabilities allow us to limit the number of blocks of the reduced process, which result in further reductions in the computation times.

1.2.3 Restricted downward transitions and $A_2^{--} = 0$

Let the matrix A_0 have the structure shown in Equation (1.5). Additionally, assume that upward transitions from states with phase in $\mathcal{S}^-$ take the process to a state with phase in $\mathcal{S}^+$, i.e., the matrix A_2 has the form

$$A_2 = \begin{bmatrix} A_2^{++} & A_2^{+-} \\ A_2^{-+} & 0 \end{bmatrix}.$$

With this additional structure, the maximum number of upward transitions between two visits to $\mathcal{S}^+$ is two, since an upward transition from $\mathcal{S}^-$ must end in $\mathcal{S}^+$. Therefore the reduced process of the M/G/1 type, constructed by observing the original process when the phase is in $\mathcal{S}^+$, has only four nonzero blocks defined as

$$\begin{aligned} \bar{A}_0 &= A_0^{++} + A_1^{+-}(I - A_1^{--})^{-1} A_0^{-+}, \\ \bar{A}_1 &= A_1^{++} + A_1^{+-}(I - A_1^{--})^{-1} A_1^{-+} + A_2^{+-}(I - A_1^{--})^{-1} A_0^{-+}, \\ \bar{A}_2 &= A_2^{++} + A_1^{+-}(I - A_1^{--})^{-1} A_2^{-+} + A_2^{+-}(I - A_1^{--})^{-1} A_1^{-+}, \\ \bar{A}_3 &= A_2^{+-}(I - A_1^{--})^{-1} A_2^{-+}. \end{aligned}$$

The definition of these blocks can be obtained directly from the equations (1.9) and (1.10) as follows: $A_2^{--} = 0$ implies that $K_i = 0$ for $i \geq 2$, which therefore means that $\bar{A}_i = 0$ for $i > 3$. Additionally, the fact that $A_2^{--} = 0$ also simplifies the expressions for K_0

and K_1, which are used in the definition of the matrices $\bar{A}_1$, $\bar{A}_2$ and $\bar{A}_3$. This additional structure reduces both the time to compute the blocks and the time to find G_+ using CR. Additionally, to find G_0 we consider again Equation (1.11) and by extracting its lower-left block we find

$$G_0 = (I - A_1^{--})^{-1} \left(A_0^{-+} + A_1^{-+} G_+ + A_2^{-+} G_+^2 \right).$$

Therefore, there is no need to solve a Sylvester matrix equation, since G_0 can be determined directly from G_+ and other already computed matrices. With this additional constraint the problem of finding the $m \times m$ matrix G is replaced by the determination of just four $r \times r$ matrices and the solution of Equation (1.7) using these smaller matrices.

1.2.4 Restricted downward and upward transitions

Now we assume that the matrices A_0 and A_2 of the QBD have the structure described in equations (1.5) and (1.6), respectively. In this case the process obtained by observing the QBD when the phase is in the set $\mathcal{S}^+$, is again a QBD with parameters

$$\begin{aligned}
\bar{A}_0 &= A_0^{++} + A_1^{+-}(I - A_1^{--})^{-1} A_0^{-+}, \\
\bar{A}_1 &= A_1^{++} + A_1^{+-}(I - A_1^{--})^{-1} A_1^{-+} + A_2^{+-}(I - A_1^{--})^{-1} A_0^{-+}, \\
\bar{A}_2 &= A_2^{++} + A_2^{+-}(I - A_1^{--})^{-1} A_1^{-+}.
\end{aligned}$$

To obtain these expressions, in addition to the simplifications due to $A_2^{--} = 0$ explained above, we notice that K_1 becomes zero since both A_2^{-+} and A_2^{--} are equal to zero. Hence $\bar{A}_3$ also becomes zero and the resulting process is again a QBD (but of a smaller block size). Moreover, the matrix G_0 is given by

$$G_0 = (I - A_1^{--})^{-1} \left(A_0^{-+} + A_1^{-+} G_+ \right).$$

The reduction in computation time is evident since now it is enough to find the solution to Equation (1.3) with matrices of size r instead of m. Additionally, the number of matrix multiplications required to compute the blocks of the new QBD process and the matrix G_0 is fixed and small compared to the solution of Equation (1.3).

1.3 QBDs with restricted upward transitions

We now turn to the case where the matrix A_2 has only $r \ll m$ nonzero rows as in Equation (1.6), restricting the upward transitions to occur only when the phase variable is in $\mathcal{S}^+$, while A_0 is no longer in the form (1.5). In a QBD the (i, j)-th entry of the rate matrix R can be interpreted as the expected number of visits to the state $(k+1, j)$ starting from state (k, i) before visiting any other state at level k [91]. To visit a state in level $k+1$ starting from level k, while avoiding level k, the first transition must take the chain from level k to level $k+1$. However, due to the structure of A_2, no upward transition can be made if the phase variable is in $\mathcal{S}^-$. Hence the last $m - r$ rows of the matrix R are equal to zero, and R can be written as

$$R = \begin{bmatrix} R_+ & R_0 \\ 0 & 0 \end{bmatrix},$$

where R_+ and R_0 are matrices of size $r \times r$ and $r \times (m-r)$, respectively. In a similar way as in the previous case, we define a new process by observing the original QBD MC when the phase variable is in $\mathcal{S}^+$. In this case the level cannot increase in the phases outside $\mathcal{S}^+$, but it can decrease several levels between two visits to $\mathcal{S}^+$. Therefore, the new process is a Markov chain of the GI/M/1 type. Using this process we can find the matrices R_+ and R_0 separately, as shown next.

1.3.1 Computing R_+

The behavior of the censored process, obtained by observing the original QBD MC when the phase is in $\mathcal{S}^+$, is characterized away from the boundary by the set of $r \times r$ matrices $(\hat{A}_i)_{i\geq 0}$. Let $\hat{R}$ be the minimal nonnegative solution of the Equation (1.8). Then the (i,j)-th entry of the matrix $\hat{R}$ can be interpreted as the expected number of visits to state $(k+1,j)$, starting from state (k,i), before the first return to level k [91], for $(i,j) \in \mathcal{S}^+$ and $k > 1$. This is the same interpretation as the (i,j)-th entry of R_+; therefore $R_+ = \hat{R}$. To find $\hat{R}$ we first need to specify the blocks $(\hat{A}_i)_{i\geq 0}$, which is done in terms of the matrices $(W_{-i})_{i\geq 0}$.

Let the entry (i,j) of the $(m-r) \times r$ matrix W_{-l} be the probability that, given that the original process starts in state (k,i) with $i \in \mathcal{S}^-$, its first transition to a state with phase in the set $\mathcal{S}^+$ occurs in the state $(k-l,j)$, for $j \in \mathcal{S}^+$, $k > l \geq 0$. Hence, the matrices $(W_{-i})_{i\geq 0}$ are given by

$$\begin{aligned} W_0 &= (I - A_1^{--})^{-1} A_1^{-+}, \\ W_{-1} &= (I - A_1^{--})^{-1} (A_0^{-+} + A_0^{--} W_0), \\ W_{-i} &= (I - A_1^{--})^{-1} A_0^{--} W_{-(i-1)}, \quad i \geq 2. \end{aligned}$$

The blocks $(\hat{A}_i)_{i\geq 0}$ can be defined in terms of the matrices $(W_{-i})_{i\geq 0}$ as

$$\begin{aligned} \hat{A}_0 &= A_2^{++} + A_2^{+-} W_0, \\ \hat{A}_1 &= A_1^{++} + A_1^{+-} W_0 + A_2^{+-} W_{-1}, \\ \hat{A}_2 &= A_0^{++} + A_0^{+-} W_0 + A_1^{+-} W_{-1} + A_2^{+-} W_{-2}, \\ \hat{A}_i &= A_0^{+-} W_{-i+2} + A_1^{+-} W_{-i+1} + A_2^{+-} W_{-i}, \quad i \geq 3. \end{aligned}$$

The blocks $\hat{A}_i$ are computed from $i = 0$ to c, where c is the smallest positive integer such that $\sum_{i=0}^{c} \hat{A}_i e > (1-\epsilon)e$. In this case it suffices to keep track of the three matrices $\{W_{-i+2}, W_{-i+1}, W_{-i}\}$ when computing the matrix A_i. As stated before, we need to compute the dual process of the GI/M/1 type MC characterized by $(\hat{A}_i)_{i\geq 0}$ in order to apply the CR algorithm. We use the dual relationship to compute the M/G/1-type blocks and, after solving a matrix equation of the type (1.7), retrieve R_+ from the G matrix of the dual. Since there are two different duals that can be used (see Appendix A.5), we consider both options and compare their performance in Section 1.4.

1.3.2 Computing R_0

By writing Equation (1.2) in block form and extracting the upper-right corner, we find

$$R_0 - R_+ R_0 A_0^{--}(I - A_1^{--})^{-1} = (A_2^{+-} + R_+ A_1^{+-} + R_+^2 A_0^{+-})(I - A_1^{--})^{-1}. \quad (1.13)$$

This is also a Sylvester matrix equation of the type $AXB + X = E$, which can be solved in $O((m-r)^3)$ time using the Hessenberg-Schur method proposed in [47] (see Appendix A.4).

1.3.3 Restricted upward transitions and $A_0^{--} = 0$

When the matrix A_2 of the QBD MC has only r nonzero rows as in Equation (1.6) and additionally the block A_0^{--} is equal to zero, we can further improve the new algorithm in a manner similar to Section 1.2.3. We omit the details as both cases are analogous.

1.4 Examples and Numerical Experiments

In this section we consider four different queueing systems in which the structures analyzed in the previous sections arise (and a standard uniformization argument is applied to transform a continuous-time problem to discrete time when necessary). We start by considering a priority queue with two customer classes that can be modeled as a QBD process with restricted downward transitions. Next we present a general MAP/PH/1 queue (see Appendix A.1), which can be modeled as a QBD process that can be induced to have restricted downward transitions. Then we illustrate the case of a QBD process with restricted upward transitions through an overflow queue. Finally, we consider the model of a relay node in a wireless network introduced in [25], where the QBD process used to evaluate the node's performance also falls within our framework. In all these cases cases we compare the times required to compute the R or G matrix using the full-size QBD and the approach proposed in this chapter. For the overflow queue we also compare with the approach introduced in [50], while for the relay node model we include a comparison with the method proposed in [25]. For further reference recall that the Kronecker product of the matrices A and B, denoted $A \otimes B$, is the block matrix with block (i, j) equal to $A_{ij}B$ [48]. The Kronecker sum $A \oplus B$ is defined as $A \otimes I + I \otimes B$, where I is an identity matrix of appropriate size.

1.4.1 A Priority Queue

Our first example is a continuous-time priority queue with two classes of customers. Class-1 customers have preemptive priority over class-2 customers. Therefore, customers of class 2 can only be served if there are no class-1 customers in the queue and the service of a class-2 customer is interrupted if a customer of class 1 arrives. The high-priority arrivals are described by a MAP characterized by (m_a^1, C_0^1, C_1^1) while the MAP of the low-priority arrivals has parameters (m_a^2, C_0^2, C_1^2). These two processes can be combined in a single marked MAP with parameters $D_0 = C_0^1 \oplus C_0^2$, $D_1 = C_1^1 \otimes I$ and $D_2 = I \otimes C_1^2$, where D_0, D_1

and D_2 are square matrices of size $m_a = m_a^1 m_a^2$. The service times of class-1 (resp. class-2) customers follow a PH distribution with parameters (m_s^1, α, T) (resp. (m_s^2, β, S)). To model this queue as a QBD with restricted downward transitions we take the level as the number of low-priority customers in the queue, and assume a finite buffer of size C for the class-1 customers. This assumption places no restriction in the analysis since this buffer can be dimensioned such that the blocking probability of the high-priority customers is below a certain threshold, allowing us to truncate its infinite size. Given the preemptive nature of the priority queue, this can be done using a QBD MC that ignores the low-priority customers. The second dimension of the QBD therefore holds the number of class-1 customers, the phase of the arrival process and the phase of the customer in service. In addition, if there is a class-1 customer in service, the service phase includes both the current phase of the customer in service and the phase in which the next class-2 customer will (re-)start its (possibly preempted) service. This is not necessary if the customer in service is of class 2, since in that case there are zero class-1 customers in the system. Therefore the blocks have size $m = m_a m_s^2 (1 + C m_s^1)$ and are given by

$$A_0 = \begin{bmatrix} I \otimes s\beta & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}, \quad A_2 = \begin{bmatrix} D_2 \otimes I_{m_s^2} & 0 & \dots & 0 \\ 0 & D_2 \otimes I_{m_s} & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & D_2 \otimes I_{m_s} \end{bmatrix},$$

$$A_1 = \begin{bmatrix} D_0 \oplus S & D_1 \otimes I \otimes \alpha & 0 & \dots & 0 \\ I \otimes t & (D_0 \otimes I) \oplus T & D_1 \otimes I & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & (D_0 \otimes I) \oplus T & D_1 \otimes I \\ 0 & 0 & \dots & I \otimes t\alpha & (\bar{D} \otimes I) \oplus T \end{bmatrix},$$

where $\bar{D} = D_0 + D_1$, $t = -Te$, $s = -Se$, $m_s = m_s^1 + m_s^2$ and the size of the identity matrix has been made explicit in those places where it might be unclear from the context. Since low-priority service completions can only occur when there are no high-priority customers in the queue, downward transitions are limited to occur when the process is in one of the first $r = \bar{m}$ phases and such transitions trigger the process to the same set of phases. Therefore the structure of A_0 can be exploited as shown in Section 1.2.

For the numerical results shown next we consider a high-priority buffer of size $C = 120$ and, for both customer classes, hyper-exponential service times with mean one and squared coefficient of variation (SCV) equal to two. The parameters of the service distribution are computed using the moment-matching method in [120], that results in a PH representation of order 2. The arrival processes are built using the method in [40, 56] that allows the matching of the first two moments of the inter-arrival distribution and the decay rate of the autocorrelation function γ with a MAP of size 2. In this case both MAPs have the same mean, fixed by the load ρ, and SCV equal to five. For this queue the load is given by $\rho = \lambda_1/\mu_1 + \lambda_2/\mu_2$, where λ_i and μ_i are the arrival and service rates of type-i customers, respectively, for $i = 1, 2$. Since the service rates are equal to one and

the arrival rates are equal, then $\lambda_1 = \lambda_2 = \rho/2$. We consider two scenarios, in the first the inter-arrival times are independent ($\gamma = 0$), while in the second γ is equal to 0.9. The selection of $C = 120$ guarantees the high-priority customers' loss rate to be below 10^{-15} (resp. 10^{-9}) for the case with $\gamma = 0$ (resp. $\gamma = 0.9$). With this set of parameters the block size is 1928 and the number of nonzero columns in A_0 is 8.

ρ	QBD-CR	Bl	# Bl	MG1-CR	Sylv	MG1	Ratio
0.1	476.9	16.4	18	0.09	47.5	64.0	7.5
0.2	545.0	16.9	31	0.11	25.8	42.8	12.7
0.3	544.9	17.5	48	0.11	47.5	65.1	8.4
0.4	613.0	18.4	70	0.16	25.8	44.4	13.8
0.5	613.0	19.5	98	0.17	25.8	45.5	13.5
0.6	680.9	20.9	135	0.27	47.5	68.6	9.9
0.7	680.9	22.9	185	0.27	47.5	70.6	9.6
0.8	749.1	25.4	250	0.28	25.9	51.5	14.5
0.9	817.1	28.9	338	1.09	25.5	55.4	14.7

Table 1.1: Computation times (sec) for the priority queue with $\gamma = 0$

In Table 1.4.1 we show the time required to compute the matrix G using the full-size QBD with the CR algorithm (QBD-CR), the time to compute the M/G/1-type blocks (Bl), the number of those blocks (# Bl), the time to compute the matrix G_+ with CR (MG1-CR) and the time to solve the Sylvester matrix equation to get G_0 (Sylv). The total computation time using the reduced process is shown in column MG1, and the last column has the ratio between the columns QBD and MG1. Clearly, the M/G/1-type based method outperforms the full-size approach, which can take 7 to 15 times longer to compute G. Also, when the load ρ increases both methods require more computation time, particularly the CR algorithm and the computation of the M/G/1-type blocks. A large load has two major effects: first, it increases the rate of upward transitions per time unit; second, since the set $\mathcal{S}^+$ includes only the phases in which there are no high-priority customers in the queue, a larger load increases the likelihood of having long sojourn times in $\mathcal{S}^-$. These two effects together imply that the number of blocks to compute increases and the CR algorithm to solve Equation (1.7) requires more time. In contrast, the Hessenberg-Schur method to solve Equation (1.12) seems to be less sensitive to the load of the queue.

Table 1.4.1 contains the same information as the previous one, but in this scenario the arrival processes are highly autocorrelated, with decay rate of the autocorrelation function $\gamma = 0.9$. As can be observed, the correlation, together with the load, has a large effect on the number of M/G/1-type blocks that describe the reduced process and therefore on the time required to compute those blocks and to find G_+. On the other hand, the correlation has little effect on the time to find G_0 with the Hessenberg-Schur method. We see that the reduced process still offers a reduction in computation times, but this gain is affected by the system parameters. A similar behavior will be observed in the subsequent examples.

In addition to the computation times, it is relevant to consider the behavior of the

ρ	QBD-CR	Bl	# Bl	MG1-CR	Sylv	MG1	Ratio
0.1	477.6	16.4	19	0.11	25.8	42.4	11.3
0.3	613.7	18.1	61	0.14	25.8	44.0	13.9
0.5	681.8	22.0	161	0.25	25.6	47.8	14.3
0.7	818.0	32.0	419	0.52	69.2	101.7	8.0
0.9	954.3	59.3	1117	1.73	25.7	86.8	11.0

Table 1.2: Computation times (sec) for the priority queue with $\gamma = 0.9$

approach introduced in this chapter in terms of the residual error. Let the infinity norm of an $n \times m$ matrix K be given by $||K||_\infty = \max_{i=1}^{n} \sum_{j=1}^{m} K_{ij}$. Let $\tilde{G}$ be the matrix that solves Equation (1.3) obtained with the approach of Section 1.2. Then the residual error is defined as

$$||\tilde{G} - A_0 + A_1\tilde{G} + A_2\tilde{G}^2||_\infty,$$

which gives a measure of the goodness of $\tilde{G}$ as a solution for Equation (1.3). In all the instances considered here the residual error was always below 10^{-14}, revealing the good behavior of the approach proposed. This behavior is to be expected since the algorithms on which our method relies (Cyclic Reduction and the Hessenberg-Schur method for the Sylvester equation) are numerically stable. A similar result in terms of the residual error holds for the other examples.

1.4.2 The MAP/PH/1 queue

The MAP/PH/1 queue receives customers according to a MAP characterized by the parameters (m_a, D_0, D_1), which are processed by a single server, and the service time is described by a PH distribution characterized by (m_s, α, T). This queue can be modeled as a QBD MC by choosing the number of customers in the queue to be the level. This selection assures that the level increases and decreases by at most one in a single transition since only one service completion or a single arrival can occur at a time. The blocks of this QBD MC are given by

$$A_0 = t\alpha \otimes I_{m_a}, \quad A_1 = T \oplus D_0, \quad A_2 = I_{m_s} \otimes D_1, \tag{1.14}$$

where $t = -Te$ and I_n is the identity matrix of size n. From this definition it is clear that the block size is $m = m_s m_a$ and that the number of nonzero columns in A_0 depends on the number of nonzero elements in the vector α. In fact, if α has only one nonzero element, then A_0 has only $r = m_a$ nonzero columns, i.e., the block size is m_s times larger than the number of nonzero columns in A_0. This is the case if the service times are described by an Acyclic PH distribution (APH) [33]. This class of distributions (which includes the Erlang and the hyper-exponential distributions as special cases) has a canonical form introduced in [33] where all the mass of the initial probability vector is concentrated in the first phase. Therefore, in this case the vector α has only one nonzero entry and the matrix A_0 has m_a nonzero columns. In general, the vector α may have any number of

nonzero entries, but we can always find a representation of size $m_s + 1$ such that the initial probability vector has only one nonzero entry, as shown in the next theorem.

Theorem 1.1. *Any continuous PH distribution with representation (m_s, α, T) also has a representation $(m_s + 1, e_1, \bar{T})$, where e_1 and $\bar{T}$ are given by*

$$e_1 = [1\ 0_{m_s}] \quad \textit{and} \quad \bar{T} = \begin{bmatrix} -c & c\alpha P \\ 0 & T \end{bmatrix},$$

where 0_n is the $1 \times n$ zero vector, c is the diagonal entry of T of largest absolute value, i.e., $c = \max\{|T_{ii}|, 1 \leq i \leq m_s\}$, and P is the uniformized version of the subgenerator matrix T, i.e., $P = \frac{1}{c}T + I_{m_s}$.

Proof. We start by uniformizing the absorbing MC that underlies the PH distribution characterized by (m_s, α, T). Since the rate corresponding to the absorbing state is zero, we can use c to uniformize the chain and therefore P holds the transition probabilities among the transient states in the uniformized chain. Also, let $\bar{P}$ be the uniformized version of the subgenerator $\bar{T}$, which is equal to

$$\bar{P} = \frac{1}{c}\bar{T} + I_{m_s+1} = \begin{bmatrix} 0 & \alpha P \\ 0 & P \end{bmatrix}.$$

Now we can write the CDF of the new representation $G(\cdot)$ as

$$\begin{aligned}
G(x) &= 1 - e_1 \exp(\bar{T}x)e = 1 - e_1 \sum_{n\geq 0} \frac{x^n}{n!}\bar{T}^n e = 1 - e_1 \sum_{n\geq 0} \frac{(cx)^n}{n!}(\bar{P} - I_{m_s+1})^n e, \\
&= 1 - e_1 \sum_{n\geq 0} \frac{(cx)^n}{n!} \sum_{k=0}^{n} \binom{n}{k} \bar{P}^k (-I_{m_s+1})^{n-k} e, \\
&= 1 - \sum_{n\geq 0} \frac{(cx)^n}{n!} \sum_{k=0}^{n} \binom{n}{k} e_1 \begin{bmatrix} 0 & \alpha P^k \\ 0 & P^k \end{bmatrix} (-I_{m_s+1})^{n-k} e, \\
&= 1 - \sum_{n\geq 0} \frac{(cx)^n}{n!} \sum_{k=0}^{n} \binom{n}{k} \alpha P^k (-I_{m_s})^{n-k} e, \\
&= 1 - \alpha \sum_{n\geq 0} \frac{(cx)^n}{n!} (P - I_{m_s})^n e, \\
&= 1 - \alpha \exp(Tx)e, \quad x \geq 0,
\end{aligned}$$

which is equal to the CDF of the original representation. Therefore (m_s, α, T) and $(m_s + 1, e_1, \bar{T})$ are two different PH representations of the same distribution. A similar result holds for discrete PH distributions. □

Using this result we can replace α and T by e_1 and $\bar{T}$, respectively, in Equation (1.14). As a consequence the block A_0 has only $r = m_a$ nonzero columns, and the new block size is $(m_s + 1)m_a$, which is exactly the structure we have referred to as restricted downward transitions. To illustrate the applicability of this result we consider a specific case of a MAP/PH/1 queue, namely a system that provides reliable messaging services. In

particular, we consider the Web Services Reliable Messaging (WSRM) protocol, which is used to ensure message transmission in web-based service oriented architectures [14]. This protocol has been analyzed in [100], and there the authors have used PH distributions to approximate the effective transmission time in a WSRM implementation. They consider different methods to obtain the PH representation, which is then used as input in an M/PH/1 queue that models the arrival and transmission of messages over WSRM. Here we consider the more general case where the arrivals are modeled as the combination of one, two or three streams, each one represented by a MAP. The transmission times are represented by a hyper-Erlang distribution with $m_s = 153$ phases, which corresponds to the case S_{2JK} considered in [100]. The parameters of this distribution were downloaded from [99]. Although this distribution is acyclic, its initial probability vector has many nonzero entries. As stated before, it is possible to use the results in [33] to obtain a canonical representation where the initial probability vector has a single nonzero entry. However, we have opted for using Theorem 1.1 to illustrate the computational gains obtained by exploiting the restricted-transitions structure, even if a slightly larger representation of the service process is needed to induce that structure.

	QBD-CR			MG1			Ratio		
ρ \ r	2	4	8	2	4	8	2	4	8
0.1	2.3	17.8	140.2	0.3	1.5	11.9	8.8	12.0	11.8
0.3	2.6	20.2	157.9	0.3	2.0	15.5	7.7	10.2	10.2
0.5	3.0	22.5	175.5	0.4	2.4	18.1	7.1	9.5	9.7
0.7	3.3	24.8	193.2	0.5	2.7	20.1	6.6	9.3	9.6
0.9	3.9	29.4	228.5	0.6	3.2	24.2	6.8	9.1	9.4

Table 1.3: Computation times (sec) for the MAP/PH/1 queue

As stated above, the arrivals come from the superposition of one, two or three sources, each one represented by a MAP of size two. This implies that the size m_a of the arrival process representation, and the number of nonzero columns r, is equal to two, four and eight, respectively. The total arrival rate λ is set to match a given load $\rho = \lambda/\mu$, where μ is the mean transmission rate. The total arrival rate is equally divided among all the sources, while each of them has SCV equal to five and the decay of their autocorrelation function is set at 0.5. These characteristics are matched by using the method introduced in [40, 56]. Table 1.4.2 shows the times required to compute the matrix G using CR directly on the blocks of size $m = m_s m_a$ (QBD-CR), and using the approach introduced in this chapter to exploit the (induced) restricted-transitions structure (MG1). It also shows the ratio between these two times (Ratio), which tells us how many times slower the general approach is compared to our specific method. We observe how in both cases the computation times are affected by the increase in the size of the arrival process representation, as is to be expected since this size affects both the original block size m and the number of nonzero columns. When there is a single source ($r = m_a = 2$), the QBD-CR method is between 6 and 9 times slower than MG1. When the number of sources increases to 2 and 3, this figure increases to between 9 and 12. We also notice that

the difference is larger for small loads and, although both methods are negatively affected by the increase of the load, this parameter has a larger effect on the MG1 method. In this case, a larger load implies the computation of a larger number of blocks in the censored process, which also means that it is necessary to solve Equation (1.7) with a larger number of nonzero coefficients. These procedures are therefore affected by the load, while finding G_0 by solving Equation (1.12) is almost insensitive to this parameter.

1.4.3 An Overflow Queue

We now consider an overflow queueing system consisting of two queues. The arrival process to the first queue is a MAP characterized by (m_a, D_0, D_1). Customers arriving at the first queue are attended in FCFS order by a single server with service times following a PH distribution characterized by the parameters (m_s^1, α, T). This queue has a finite buffer of size C and a customer that finds the buffer full is sent to the second queue. The second queue receives *only* overflow arrivals from the first queue and attends them in FCFS order with a single server. The service times in this queue follow a PH distribution with parameters (m_s^2, β, S). Hence, the arrival process at the second queue can be described by a MAP with parameters (m_o, C_0, C_1) given by $m_o = (C+1)m_a m_s^1$,

$$C_0 = \begin{bmatrix} D_0 \otimes I & D_1 \otimes I & 0 & \cdots & 0 & 0 \\ I \otimes t\alpha & D_0 \oplus T & D_1 \otimes I & \cdots & 0 & 0 \\ 0 & I \otimes t\alpha & D_0 \oplus T & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & D_0 \oplus T & D_1 \otimes I \\ 0 & 0 & 0 & \cdots & I \otimes t\alpha & D_0 \oplus T \end{bmatrix}, C_1 = \begin{bmatrix} 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & D_1 \otimes I \end{bmatrix},$$

where $t = -Te$. Assuming an infinite buffer at the second queue, we can model the queueing system as a QBD where the level describes the number of customers in the second queue. The second dimension holds the phase of the current customer in service and the phase of the arrival process at the second queue. The parameters of the QBD are $A_0 = I \otimes s\beta$, $A_1 = C_0 \oplus S$, $A_2 = C_1 \otimes I$, with $s = -Se$. In this case, the restricted upward transitions are a result of the overflow process, as can be seen in the structure of C_1, where the arrivals to the second queue can only occur in the last $m_a m_s^1$ phases. The inclusion of a separate arrival stream directed to the second queue would suppress this structure. The block size in this case is $m = m_o m_s^2$ and the number of nonzero rows in A_2 is $r = m_a m_s^1 m_s^2$.

As with the previous examples, we make use of the moment-matching methods in [40, 56, 120] to obtain PH and MAP representations of the service and arrival processes, respectively. The arrival process at the first queue has arrival rate and SCV equal to five while the service time has mean one and SCV equal to two. Therefore the first queue is heavily loaded and many customers are overflowed to the second queue. The arrival rate at the second queue (λ_2) is the arrival rate of the MAP with parameters (C_0, C_1). Therefore for a given load at the second queue (ρ_2) the service rate at this queue is fixed by the relation $\rho_2 = \lambda_2/\mu_2$. In this queue the service times have SCV equal to two, as in the first queue. The results are presented for different values of ρ_2 and a buffer size of

$C = 100$ in the first queue. With these parameters the block size is $m = 808$ while the number of nonzero rows in A_2 is $r = 8$.

ρ_2	Q-CR	Bl	# Bl	G-CR-R	G-CR-B	Sylv	GM1-R	GM1-B	Rat-R	Rat-B
0.1	31.91	24.92	2791	17.11	8.14	2.25	44.28	35.31	0.72	0.90
0.3	42.3	8.63	980	2.33	0.53	2.25	13.20	11.41	3.20	3.71
0.5	47.38	5.63	601	1.17	0.64	3.94	10.73	10.2	4.41	4.64
0.7	52.47	4.41	442	0.81	0.58	2.24	7.45	7.22	7.04	7.27
0.9	62.67	3.70	350	0.66	0.45	2.23	6.59	6.39	9.51	9.81

Table 1.4: Computation times (sec) for the overflow queue with $C = 100$

Table 1.4.3 shows the computation times in a similar fashion as in Section 1.4.1. Here the column Q-CR holds the times to solve the original QBD MC using the CR algorithm. Also, the columns G-CR-R (resp. G-CR-B) includes the time to compute the blocks of the Ramaswami (resp. Bright) dual process and the time to solve the dual with the CR algorithm. The columns GM1-R and GM1-B show the total computation times to find R using the two different duals, while the columns Rat-R and Rat-B hold the ratio between the Q-CR and the GM1-R and GM1-B columns, respectively. Again, the load has an important effect on the computation times, but in this case the consequences are reversed. When the load is low the original process can make many downward transitions between two visits to the set $\mathcal{S}^+$, increasing the number of GI/M/1-type blocks. As before, a large number of blocks increases the computation time of the CR algorithm for the M/G/1-type MC (the dual process), but it has little effect on the solution of the Sylvester equation and the full-size QBD. For loads between 0.2 and 0.9 in this scenario, the solution of the full-size QBD may take between 2 and 10 times as long as the solution of the reduced process. When the load is equal to one the process is null recurrent and Q-CR requires much longer times than for lower loads. This effect can be reduced by using the shift technique [18], resulting in times similar to those shown for loads up to 0.9. When comparing the two alternative duals, it is clear how the Bright dual outperforms the Ramaswami dual, being specially effective when the load is low, i.e., when the number of GI/M/1-type blocks is large. This effect is to be expected since for $\rho_2 < 1$ the GI/M/1-type MC is positive recurrent, and therefore the Ramaswami dual is transient, while the Bright dual is positive recurrent (see [109]).

In Figure 1.1 we include, for the full-size QBD, the computation times of CR (QBD-CR), the original U-based algorithm (QBD-U) and the modified version of the U-based algorithm (QBD-GT) proposed by Grassman and Tavakoli [50] to exploit the special structure of A_2. We also include the total time required to solve the reduced process using the Ramaswami dual (GM1-R) and the Bright dual (GM1-B). The scenario is the same as in the previous case with the only exception that the buffer size in the first queue is $C = 50$. This means that the block size is $m = 408$ while the number of nonzero rows in A_2 does not change. This reduction is done because of the long computation times experienced with the U-based method, as can be observed in the figure. From these results the substantial gain obtained by the QBD-GT method compared to the original QBD-U is evident, as the latter requires about 5 times as much computation time. In

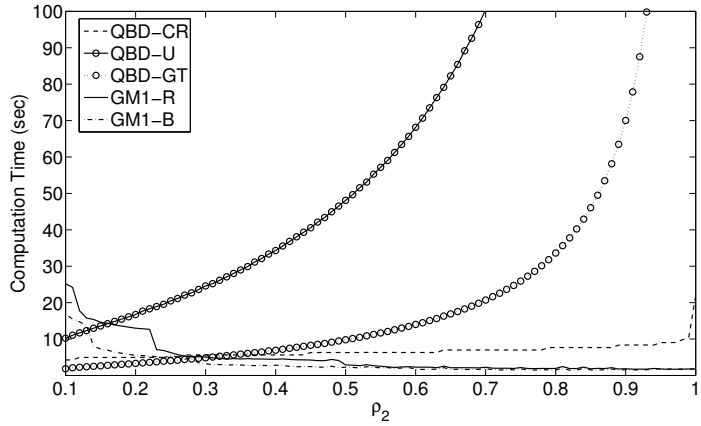


Figure 1.1: Computation times (sec) for the overflow queue with $C = 50$

spite of this gain, the QBD-GT method performs better than the QBD-CR only for small values of ρ_2. In contrast, the GI/M/1-type based approach performs better than CR on the full-size QBD, except for low values of ρ_2. For the remaining part of the load range (except $\rho_2 = 1$) the QBD-CR takes up to 6 times as much time as the GI/M/1-type based approach. In this case the time to compute R is smaller using the Bright dual than the Ramaswami dual. The difference is significant for low loads, when the number of blocks is large (2800 for $\rho_2 = 0.1$), and it vanishes as the load increases. Therefore, the use of the Bright dual implies an important reduction in computation times in the range of the load that is more critical for the reduced process.

1.4.4 Wireless Relay node

Our last example is a model introduced in [25] to evaluate the packet-level performance in a wireless node implementing user relaying. In a wireless network a source node can hear what another source node is sending to the destination node, and it could therefore retransmit the information to the destination node. The node that retransmits the packets sent by another source node is called the relay node. The main purpose of enabling the source nodes to act as relay nodes is to provide multiple channels to transmit the same information, helping to alleviate the network's capacity loss caused by, for instance, channel fading [25]. To model a single relay node, the authors in [25] start by setting up an M/G/1-type MC where the level is the number of packets waiting for transmission in this node. The model is in discrete time and in a single slot only one packet can be transmitted while the number of packets that arrives at the relay node can be between zero and $K < \infty$. However, taking advantage of the finite nature of K the authors propose a re-blocking of the MC's transition matrix to transform it into a QBD with block size Km, where m is the block size of the original M/G/1-type MC. The re-blocking operation has two main advantages: first, it is computationally less expensive to determine the stationary probability vector of a QBD MC than that of an M/G/1-type MC, once the matrix G of each of them has been computed, due to the matrix-geometric

property; and second, in this particular case the matrix A_0 of the QBD MC has a few nonzero columns, a structure that can be exploited with the methods introduced in this chapter. More specifically, the model in [25] considers three possible cases for the behavior of the relay node: in the first case the matrix A_0 has only one nonzero column, allowing the matrix G to be expressed directly in terms of the system parameters; in the other two cases the matrix A_0 has two nonzero columns, which prevents expressing G in closed form. Regarding the operation of the relay node, the three cases differ on how the node is allowed to participate in relaying the transmission of another source, and when it can transmit its own packets.

K	100					200				
ρ	FI	QBD-CR	MG1	Ratio-F	Ratio-C	FI	QBD-CR	MG1	Ratio-F	Ratio-C
0.1	2.4	0.3	0.1	17.2	2.4	13.5	2.5	0.6	23.3	4.4
0.3	4.1	0.4	0.1	33.0	3.4	22.8	3.2	0.6	40.5	5.7
0.5	6.6	0.5	0.1	52.6	4.0	36.3	3.2	0.6	62.7	5.5
0.7	11.8	0.5	0.1	83.3	3.5	64.8	3.8	0.6	112.2	6.7
0.9	34.5	0.7	0.2	219.7	4.2	189.7	5.2	0.6	337.0	9.2

Table 1.5: Computation times (sec) for the wireless relay node - Case 2

K	100					200				
ρ	FI	QBD-CR	MG1	Ratio-F	Ratio-C	FI	QBD-CR	MG1	Ratio-F	Ratio-C
0.1	5.2	2.5	0.4	12.3	6.0	32.8	19.3	2.7	12.1	7.1
0.3	8.9	3.2	0.4	21.2	7.6	55.4	24.3	2.8	20.0	8.8
0.5	14.0	3.8	0.4	33.1	9.1	88.4	24.3	2.8	32.2	8.8
0.7	24.8	3.8	0.4	58.9	9.1	158.2	29.3	2.8	57.5	10.6
0.9	72.4	5.2	0.4	171.1	12.2	460.1	39.2	2.8	163.6	13.9

Table 1.6: Computation times (sec) for the wireless relay node - Case 3

We consider the two cases where G cannot be found explicitly, to compare the performance of our method (MG1) not only with the CR algorithm on the full-size QBD (QBD-CR), but also with the approach proposed in [25], which also exploits the structure of A_0. This latter approach relies on a functional iteration to find the components of the G matrix, and therefore it has been labeled FI. We assume uncorrelated arrivals, as in [25], although the model can be easily generalized to allow for MAP arrivals while preserving the restricted-downward-transitions structure. The distribution of the number of packets arriving in a single slot is assumed to be uniformly distributed between one and K, and the probability of having zero arrivals is set according to the load ρ of the node. The results for the two cases are shown in Tables 1.4.4 and 1.4.4, which are labeled Case 2 and Case 3, respectively, as in [25]. The block-size m of the original M/G/1-type MC for cases 2 and 3 is equal to 2 and 4, respectively, and recall that the size of the QBD blocks is Km. From these results we observe that in both cases there is a significant gain that can be obtained by exploiting the restricted-transitions structure with the methods

introduced here. Moreover, the FI method is dramatically slower than both QBD-CR and MG1, even though this method takes advantage of the special structure of the block A_0. The main reason for this behavior, as with the QBD-GT method in the previous example, is that the gain obtained at each iteration of the algorithm is not enough to compensate for the large number of iterations required to find G. Additionally, the number of iterations, and therefore the computation time, is badly affected by the load of the node. For instance, for the third case and $K = 100$ the FI algorithm requires around 5 seconds to run if the load is 0.1, but more than one minute if the load equals 0.9. The computation time therefore increases by a factor of fourteen, while for the same scenario the QBD-CR method requires about twice as much time when $\rho = 0.9$ compared to the case when $\rho = 0.1$. Furthermore, for the same instance the MG1 method is almost insensitive to the load and requires less than half a second to complete the computation of G.

Additionally, when comparing the performance of the various methods in the two cases, we observe that all of them require significantly more time in Case 3 than in Case 2. However, the difference between these two cases is not proportional for all the methods. In fact, for the FI method the average ratio between the times for Case 3 to those for Case 2 is slightly above two. For QBD-CR this figure is well above seven, while for MG1 it is around four. In spite of this relatively better behavior of the FI method, in absolute terms it requires substantially more time than MG1. Actually, for Case 3 the FI approach is between 12 and 170 times slower than our method. For the the QBD-CR, this figure ranges between 6 and 14, confirming the benefits of exploiting the restricted-transitions structure with the approach introduced in this chapter.

Notice that during the numerical examples we have encountered three different behaviors of the MG1 method in relation to the load of the system under analysis: for the MAP/PH/1 and the priority queues the computation time increases with the load, for the overflow queue it decreases and for the relay node it is almost unaffected by the load. For the current example we see that the level i of the QBD holds the set of states where the number of packets waiting to be transmitted is between $(i-1)K+1$ and iK. Within level i, the set $\mathcal{S}^+$ holds the states where the number of waiting packets is the largest in the level (iK). When the load increases we expect the chain to make longer excursions toward higher levels, meaning the censored process has more blocks, but there is no particular reason for the chain to avoid or prefer the states with phase in $\mathcal{S}^+$. This is in contrast with the priority and the overflow queues, where a larger load implies that visits to the states with phase in $\mathcal{S}^+$ become less and more likely, respectively.

By means of the examples considered in this section, we have shown that the computation times to find the R or G matrix of the QBD MC can be substantially reduced with the approach proposed in this chapter. As expected, for every case the gain increases with the ratio m/r, but it also depends on other factors related to the parameters of the system modeled. Even though for some cases the reduced-process approach may take longer than solving the full-size QBD, exploiting the structure of the matrices A_0 or A_2 may reduce the computation times substantially. To determine whether the approach introduced here can be useful for a particular system or not, attention must be paid to the expected sojourn times in $\mathcal{S}^-$ related to those in $\mathcal{S}^+$. If the sojourn times in $\mathcal{S}^-$ are too long compared to the sojourn times in $\mathcal{S}^+$, the reduced process will need many blocks

to be described. This increases both the time required to compute the blocks and the time to find G_+ or R_+. However, to analyze the performance of a particular system it is usual to consider a broad range of conditions (load, variability, etc.), and it is likely that for a considerable part of this range our approach provides important reductions in computation times. An additional gain can be obtained for the GI/M/1-type case by using the Bright dual, which helps to reduce the computation times specially in those cases where the reduced process requires more time, i.e., when the number of blocks is large.

Chapter 2

M/G/1-type Markov chains with restricted downward transitions

In the previous chapter, we introduced M/G/1-type Markov chains (MCs) as part of the analysis of QBD MCs with restricted transitions. In this chapter we will consider M/G/1-type MCs with restricted transitions and therefore we recall their definition. Also, the development here will be done in continuous time, as a complement to the discrete-time setting used in the previous chapter. A continuous-time M/G/1-type MC [92] is a two-dimensional process $\{(N_t, X_t), t \geq 0\}$ where the *level* variable N_t takes values on $\mathbb{N}$, while the *phase* variable X_t takes values on a finite set of size m_b or m, depending on whether the level is equal to or greater than zero. The transition rates are level-independent and the level can only decrease by one during a single transition. Therefore, the generator matrix Q of an M/G/1-type MC is of the form

$$Q = \begin{bmatrix} B_0 & B_1 & B_2 & B_3 & \cdots \\ C_0 & A_1 & A_2 & A_3 & \cdots \\ & A_0 & A_1 & A_2 & \cdots \\ & & A_0 & A_1 & \cdots \\ 0 & & & \ddots & \ddots \end{bmatrix}, \tag{2.1}$$

where A_i, for $i \geq 0$, are $m \times m$ matrices, B_0 is an $m_b \times m_b$ matrix, C_0 is an $m \times m_b$ matrix and B_i, for $i \geq 1$, are $m_b \times m$ matrices. All these matrices have nonnegative real entries, with the exception of the diagonal entries of the matrices B_0 and A_1, which are negative and such that the matrix Q has zero row sums. There are two main steps to determine the stationary probability vector of this MC. The first step is to find the matrix G that is the minimal non-negative solution of the matrix equation

$$\sum_{i=0}^{\infty} A_i G^i = 0. \tag{2.2}$$

This equation can be solved using iterative algorithms such as the (linearly-convergent) functional iterations [76,92] or the (quadratically-convergent) Cyclic Reduction algorithm [18, 19]. The second step is to compute the stationary probability vector by means of Ramaswami's formula [97], which relies on the matrix G.

Although it is possible to analyze a broad range of systems using M/G/1-type MCs, they suffer from the curse of dimensionality, as discussed earlier. To circumvent this problem we consider M/G/1-type MCs where a downward transition can only trigger the phase to a small subset of the phase space. In other words, the matrix A_0 is assumed to have only $1 < r \ll m$ nonzero columns, a structure referred to as restricted downward transitions. The main contribution of this chapter is to analyze and exploit this structure to reduce the total computation time to find the stationary probability vector of the chain. For this purpose, we define a new M/G/1-type MC by observing the original chain when the phase variable is in one of the r phases corresponding to the nonzero columns of A_0. The blocks of this new MC are of size r and therefore the solution of this chain can be carried out significantly faster. Moreover, we show how the G matrix of the original MC can be obtained from the G matrix of the censored chain. After finding G we further exploit its structure to speed-up the computation of the stationary probability vector of the MC.

Even though at first sight the form assumed for A_0 appears to be rather restrictive, this structure actually arises or can be induced in several, even well-studied, queueing models. Section 2.3 will show how the BMAP/PH/1 queue and the BMAP[2]/PH[2]/1 preemptive priority queue can be modeled as M/G/1-type MCs with restricted downward transitions. Moreover, these queues provide some additional structure which can be exploited in combination with our general approach to further reduce the computation time to find the stationary probability vector. These two queueing systems will also be used to illustrate numerically the substantial reduction in computation time that can be obtained by using the methods introduced in this chapter. In fact, for the BMAP[2]/PH[2]/1 preemptive priority queue we found that our approach is not only faster than solving the original M/G/1-type MC, but it is also faster than other methods previously proposed to find the queue-length distribution of this particular queue [129]. Although we consider these two queues in detail, there are many other systems where the restricted transitions arise, such as the meteor burst packet model presented in [30].

In the next section we will show how the computation of the matrix G can be sped up by exploiting the restricted-transitions structure. After finding G, this structure can be used to accelerate the computation of the stationary probability vector, as discussed in Section 2.2. The performance of our approach is illustrated through the numerical results presented in Section 2.3. Although this chapter deals with MCs in continuous time, all the results can be easily translated and applied to discrete-time MCs. Besides, the approach presented here can be applied *mutatis mutandis* to compute the matrix R of a GI/M/1-type MC with restricted upward transitions. In these MCs an upward transition, which increases the level by at most one, can only occur if the phase variable is in a small subset of the phase space. After computing the matrix R the stationary probability vector can be easily obtained as it has the matrix-geometric property.

2.1 Analyzing an M/G/1-type MC with restricted transitions

In this section we focus on the analysis of a general M/G/1-type MC with restricted downward transitions. We focus on the computation of the matrix G, the minimal nonnegative solution of Equation (2.2). The development here follows the lines of Section 1.2, where a similar analysis was carried out for QBD processes with restricted downward transitions. To start with, let us partition the phase space of the M/G/1-type MC, i.e., the set $\{1,\ldots,m\}$, into two subsets: $\mathcal{S}^+ = \{1,\ldots,r\}$ and $\mathcal{S}^- = \{r+1,\ldots,m\}$. We can partition the matrices A_i according to these sets as

$$A_i = \begin{bmatrix} A_i^{++} & A_i^{+-} \\ A_i^{-+} & A_i^{--} \end{bmatrix}, \; i = 0,\ldots,N, \tag{2.3}$$

where N is the smallest integer such that $A_i = 0$ and $B_i = 0$ for $i > N$, and A_i^{++} and A_i^{--} are square matrices of size r and $m-r$, respectively. We assume that the MC is irreducible and therefore the matrices A_1^{++} and A_1^{--} are transient generators and their inverses exist. As stated in the previous sections, our focus is on the case where the block A_0 has only a few nonzero columns. Here we assume without loss of generality that those columns are the first $r \ll m$, and therefore the matrix A_0 can be written as

$$A_0 = \begin{bmatrix} A_0^{++} & 0 \\ A_0^{-+} & 0 \end{bmatrix}. \tag{2.4}$$

In general, to compute the matrix G one must rely on iterative algorithms like functional iterations or Cyclic Reduction (CR) [18, 76, 92]. However, if the matrix A_0 has the structure in (2.4) and if $r = 1$ ($\mathcal{S}^+$ is unitary), the matrix G can be computed explicitly [76]. Here we assume that A_0 has $r > 1$ nonzero columns. The main consequence of this structure is that the matrix G also has r nonzero columns. To see this recall that the (i,j)-th entry of the matrix G holds the probability that, if the chain starts in state (k,i), the first visit to level $k-1$ occurs by visiting state $(k-1,j)$, for $k > 1$, $1 \le i,j \le m$ [76]. In addition, in an M/G/1-type MC, if the chain starts in level k the first visit to level $k-1$ must be a downward transition from level k to level $k-1$, which is governed by A_0. If this matrix has the structure in (2.4), the first visit to level $k-1$, starting from level k, must be to a state with phase in $\mathcal{S}^+$. Therefore, only the first r columns of G are different from zero and this matrix can be written as

$$G = \begin{bmatrix} G_+ & 0 \\ G_0 & 0 \end{bmatrix},$$

where G_+ (resp. G_0) is an $r \times r$ (resp. $(m-r) \times r$) matrix. In the following subsections we show how the matrices G_+ and G_0 can be computed separately such that the total time required to compute G can be significantly reduced when $r \ll m$.

Remark 2.1. *In addition to the structure in* (2.4)*, we have found M/G/1-type MCs where the blocks feature an additional structure, e.g. when modeling the batch queues introduced in Section 2.3. The most relevant feature is that the matrices* $(A_i^{--})_{i=2}^N$ *can be written as*

the product of a scalar and a common matrix, i.e., $A_i^{--} = c_i K$, for $2 \leq i \leq N$. Another observation is that the blocks $(A_i^{+-})_{i=2}^N$ and $(A_i^{-+})_{i=2}^N$ are actually equal to zero. This additional structure will be exploited in those places where it may provide a significant gain in the computation time to find the stationary probability vector. Apart from this, the remainder of the chapter focuses mostly on the structure in (2.4)*, so that the approach proposed here can be used in other applications where only this structure arises.*

Remark 2.2. *As mentioned in the introduction to this Part, the approach in [30] also deals with M/G/1-type MCs with restricted transitions. However, this method imposes an additional restriction on the blocks of the MC. This restriction can be translated into our notation by imposing that*

$$A_i = \begin{bmatrix} A^{++} & A^{+-} \\ A^{-+} & A^{--} \end{bmatrix} \begin{bmatrix} a_i^+ I & 0 \\ 0 & a_i^- I \end{bmatrix}, \quad i \geq 0,$$

where $(a_i^+)_{i\geq 0}$ and $(a_i^-)_{i\geq 0}$ are nonnegative scalars and $a_0^- = 0$. The most restrictive implication of this assumption is that the block A_1 (transitions within the same level) must have the same pattern (up to a multiplication by a scalar) than the blocks $(A_i)_{i\geq 2}$ (transitions to upper levels). This means that, for instance, the BMAP[2]/PH[2]/1 priority queue cannot be modeled with the approach in [30] as the transitions within the same level (arrivals and service completions of high-priority customers) and those to upper levels (arrivals of low-priority customers) trigger very different transitions on the phase variable (see Section 2.3).

2.1.1 Computing G_+

The computation of the matrix G is split in two steps: we first compute G_+ by using a censoring argument and afterward we obtain G_0 by solving a linear system. To compute G_+ we define a new process by observing the original M/G/1-type MC only when the phase variable is in $\mathcal{S}^+$. Since in the original chain any downward transition takes the chain to a state with phase in $\mathcal{S}^+$, the level in the new chain can decrease by at most one in a single transition. Also, in the original chain there can be many upward transitions between two visits to states with phase in $\mathcal{S}^+$. Therefore, the chain that results from observing the original MC when the phase variable is in $\mathcal{S}^+$ is also of the M/G/1 type. In this case however the size of the blocks is equal to r, which is assumed to be significantly smaller than m. Let the $r \times r$ blocks $(\bar{A}_i)_{i\geq 0}$ characterize the behavior, away from the boundary, of the new M/G/1-type MC, and let $\bar{G}$ be the associated minimal nonnegative solution of Equation (2.2). As stated before, the (i,j)-th entry of $\bar{G}$ holds the probability that, given that the chain starts in state (k,i), the first visit to level $k-1$ occurs by visiting state $(k-1,j)$, for $1 \leq i,j \leq r$. However, this is the same definition of the (i,j)-th entry of G_+, since in the original chain the first visit to level $k-1$, starting from level k, can only occur to a state with phase in $\mathcal{S}^+$. Therefore, to compute G_+ we can compute the blocks $(\bar{A}_i)_{i\geq 0}$ of the censored process and then solve Equation (2.2) to obtain $\bar{G}$, which is equal to G_+.

To determine the blocks $(\bar{A}_i)_{i\geq 0}$ we define the $(m-r) \times r$ matrices $(W_l)_{l\geq -1}$. The (i,j)-th entry of W_l holds the probability that, given that the chain starts in level $k > 1$

and phase $i \in \mathcal{S}^-$, the first passage to a state with phase in $\mathcal{S}^+$ occurs by visiting state $(k+l, j)$, for $j \in \mathcal{S}^+$ and $l \geq -1$. Relying on these matrices, the blocks $(\bar{A}_i)_{i\geq 0}$ can be computed by conditioning on the first transition in the original chain. For instance, to define the matrix $\bar{A}_0$ there are two possible sets of transitions: the chain may move directly from (k, i) to $(k-1, j)$ with $i, j \in \mathcal{S}^+$ (according to A_0^{++}); or it may first move within the same level to a state (k, l) with $l \in \mathcal{S}^-$ (according to A_1^{+-}), and then make a number of transitions that will take the chain to a state $(k-1, j)$ with $j \in \mathcal{S}^+$ while avoiding any state with phase in $\mathcal{S}^+$ (according to W_{-1}). Therefore, the matrix $\bar{A}_0$ can be obtained as $\bar{A}_0 = A_0^{++} + A_1^{+-}W_{-1}$. Other paths from level k to $k-1$ are ruled out because they would involve either a direct transition to a state with phase in $\mathcal{S}^+$, or a transition to a state with phase in $\mathcal{S}^-$, but level greater than k, from which, to return to $k-1$, it would be necessary to visit states in level k with phase in $\mathcal{S}^+$. By proceeding with a similar analysis we find that the blocks $(\bar{A}_i)_{i\geq 0}$ are given by

$$\bar{A}_i = \begin{cases} A_i^{++} + \sum_{j=1}^{i+1} A_j^{+-}W_{i-j}, & 0 \leq i \leq N-1, \\ A_N^{++} + \sum_{j=1}^{N} A_j^{+-}W_{N-j}, & i = N, \\ \sum_{j=1}^{N} A_j^{+-}W_{i-j}, & i \geq N+1. \end{cases} \tag{2.5}$$

Recall, $A_i = 0$ for $i > N$.

We now turn to the computation of the matrices $(W_l)_{l\geq -1}$. We start by noticing that, to go from state (k, i) to state $(k-1, j)$, with $i \in \mathcal{S}^-$ and $j \in \mathcal{S}^+$, while avoiding any states with phase in $\mathcal{S}^+$, the chain must make a downward transition immediately after a sojourn in the set of states (k, l) with $l \in \mathcal{S}^-$. This results in that W_{-1} is given by $W_{-1} = (-A_1^{--})^{-1}A_0^{-+}$. In a similar manner we find that W_0 is given by $W_0 = (-A_1^{--})^{-1}(A_1^{-+} + A_2^{--}W_{-1})$. Here the chain has the additional option of going from level k to level $k+1$ (according to A_2^{--}) and then returning to level k while avoiding states with phase in $\mathcal{S}^+$ (according to W_{-1}). Following the same argument we find that the matrices $(W_l)_{l\geq -1}$ can be computed recursively as

$$W_i = \begin{cases} (-A_1^{--})^{-1}\left(A_{i+1}^{-+} + \sum_{j=2}^{i+2} A_j^{--}W_{i-j+1}\right), & -1 \leq i \leq N-2, \\ (-A_1^{--})^{-1}\left(A_N^{-+} + \sum_{j=2}^{N} A_j^{--}W_{N-j}\right), & i = N-1, \\ (-A_1^{--})^{-1}\sum_{j=2}^{N} A_j^{--}W_{i-j+1}, & i \geq N. \end{cases} \tag{2.6}$$

From Equation (2.5) we observe that to compute $\bar{A}_i$ we need to keep track of the matrices $\{W_{i-N}, \ldots, W_{i-1}\}$, for $i > N$. Therefore, we must store N matrices of size $(m-r)\times r$. After obtaining $\bar{A}_i$, the value of W_i is computed as a function of $\{W_{i-N+1}, \ldots, W_{i-1}\}$, for $i \geq N$. Then, the value of W_{i-N} can be discarded since the set $\{W_{i-N+1}, \ldots, W_i\}$ suffices to determine $\bar{A}_{i+1}$. This procedure continues until the matrix $\bar{A}_M$ is computed, where M is the smallest positive integer such that $\sum_{i=0}^{M} \bar{A}_i e > -\epsilon e$, with $\epsilon = 10^{-14}$ and e a column vector of ones. Once the blocks $(\bar{A}_i)_{i=0}^{M}$ are computed, the CR algorithm [19] can be used to solve Equation (2.2) to obtain G_+.

2.1.2 Computing G_0

Once G_+ has been computed, we can obtain G_0 by solving a linear system. This results by considering the partition in (2.3) and the structure in (2.4) to rewrite Equation (2.2)

as

$$-\begin{bmatrix} A_0^{++} & 0 \\ A_0^{-+} & 0 \end{bmatrix} = \sum_{i=1}^{N} \begin{bmatrix} A_i^{++} & A_i^{+-} \\ A_i^{-+} & A_i^{--} \end{bmatrix} \begin{bmatrix} G_+ & 0 \\ G_0 & 0 \end{bmatrix}^i = \sum_{i=1}^{N} \begin{bmatrix} A_i^{++} & A_i^{+-} \\ A_i^{-+} & A_i^{--} \end{bmatrix} \begin{bmatrix} G_+^i & 0 \\ G_0 G_+^{i-1} & 0 \end{bmatrix}.$$

Now, extracting the lower-left block we find

$$-\sum_{i=1}^{N} A_i^{--} G_0 G_+^{i-1} = \sum_{i=0}^{N} A_i^{-+} G_+^i, \tag{2.7}$$

which is a general linear system of the form $\sum_{i=1}^{N} A_i X B_i = C$, where G_0 is the only unknown term. This system has $(m-r)r$ unknowns and equations, therefore its solution by general procedures has a time complexity of $O((m-r)^3 r^3)$. Hence, this system can be solved directly if r is very small. Another possibility is to use an iterative approach as those proposed in [17], although these are not guaranteed to converge to the actual solution. However, the system (2.7) has a special characteristic: the matrices that post-multiply the unknown matrix G_0 are all powers of the same matrix G_+. We have devised a way to exploit this fact, reducing the time complexity to $O((m-r)^3 r)$. Also, there are two special cases where the general system (2.7) can be reduced to a Sylvester matrix equation, as in Chapter 1, and can therefore be solved in $O((m-r)^3)$ time with the Hessenberg-Schur algorithm proposed in [47] (see Appendix A.4). In the next sections we explain how the general and the two special cases can be approached.

The general case

The key to solve Equation (2.7) is to apply a real Schur decomposition [48] to G_+, i.e., to find an orthogonal matrix $U \in \mathbb{R}^{r \times r}$ such that $U' G_+ U = T$, where $'$ denotes the transpose operator. Recall that a matrix U is called orthogonal if $U'U = UU' = I$ [48]. The matrix $T \in \mathbb{R}^{r \times r}$ is upper quasi-triangular, meaning it is block upper triangular and the diagonal blocks are of size one or two [48]. We now post-multiply (2.7) by U to obtain

$$-\sum_{i=1}^{N} A_i^{--} G_0 U U' G_+^{i-1} U = \sum_{i=0}^{N} A_i^{-+} G_+^i U,$$

which, since $U' G_+^j U = T^j$ for any nonnegative integer j, can be rewritten as

$$-\sum_{i=1}^{N} A_i^{--} G_0 U T^{i-1} = \sum_{i=0}^{N} A_i^{-+} G_+^i U.$$

Now let $Y = \sum_{i=0}^{N} A_i^{-+} G_+^i U$, which is a known matrix, and let $X = G_0 U$, to obtain

$$-\sum_{i=1}^{N} A_i^{--} X T^{i-1} = Y. \tag{2.8}$$

This system can be equivalently written in a column-wise form as

$$-\sum_{i=1}^{N} A_i^{--} \sum_{j=1}^{r} [T^{i-1}]_{jk} X_j = Y_k, \tag{2.9}$$

for $k = 1, \ldots, r$, where M_k and $[M]_{i,j}$ are the k-th column and the (i,j)-th entry of a matrix M, respectively.

However, in Equation (2.8) the matrices that post-multiply X are all upper quasi-triangular matrices, and all they have the same block structure as they are powers of T. Therefore it is possible to iteratively compute the columns X_k, starting with X_1. Let us assume that we have already found $\{X_1, \ldots, X_{k-1}\}$ and we want to compute X_k, for some $1 \leq k \leq r$. Given the upper quasi-triangular nature of T there are two possibilities. The first is that the entry $[T]_{k+1,k}$ is zero, meaning that Equation (2.9) can be rewritten as

$$-\sum_{i=1}^{N} A_i^{--}[T^{i-1}]_{kk} X_k = Y_k + \sum_{i=1}^{N} A_i^{--} \sum_{j=1}^{k-1} [T^{i-1}]_{jk} X_j.$$

Therefore we can obtain the column X_k by solving a linear system of size $m - r$, which requires $O((m-r)^3)$ time. The second case is when $[T]_{k+1,k} \neq 0$, which, thanks to the upper quasi-triangular structure (the diagonal blocks are at most of size two) implies that $[T]_{k+2,k+1} = 0$. Therefore we can find the columns X_k and X_{k+1} simultaneously by solving the system

$$-\begin{bmatrix} \sum_{i=1}^{N} A_i^{--}[T^{i-1}]_{kk} & \sum_{i=1}^{N} A_i^{--}[T^{i-1}]_{k+1,k} \\ \sum_{i=1}^{N} A_i^{--}[T^{i-1}]_{k,k+1} & \sum_{i=1}^{N} A_i^{--}[T^{i-1}]_{k+1,k+1} \end{bmatrix} \begin{bmatrix} X_k \\ X_{k+1} \end{bmatrix} = \begin{bmatrix} \hat{Y}_k^{k-1} \\ \hat{Y}_{k+1}^{k-1} \end{bmatrix},$$

where $\hat{Y}_k^l = Y_k + \sum_{i=1}^{N} A_i^{--} \sum_{j=1}^{l} [T^{i-1}]_{jk} X_j$, for $1 \leq l \leq k-1$ and $1 \leq k \leq r$. This is a linear system with $2(m-r)$ unknowns that requires $O((m-r)^3)$ time to be solved. As a result we can start by finding the first (two) column(s) of X and iteratively compute the others. After we have computed X, G_0 is obtained from $G_0 = XU'$. Since the Schur decomposition of G_+ requires $O(r^2)$ time, the computation of G_0 has a time complexity of $O(r(m-r)^3)$.

Two special cases

There are two special cases where an $O((m-r)^3)$ algorithm can be used to compute G_0 from G_+. In the first case it is assumed that only one of the $(A_i^{--})_{i=2}^{N}$ matrices is different from zero. Let A_k^{--} be the only of these matrices different from zero, for some $2 \leq k \leq N$. Then Equation (2.7) can be written as

$$G_0 - (-A_1^{--})^{-1} A_k^{--} G_0 G_+^{k-1} = (-A_1^{--})^{-1} \sum_{i=0}^{N} A_i^{-+} G_+^i. \tag{2.10}$$

This is a Sylvester matrix equation [44, 47] of the type $AXB + X = C$, which can be solved in $O((m-r)^3)$ time with the Hessenberg-Schur method proposed in [47].

The second case arises when the matrices $(A_i^{--})_{i=2}^{N}$ can be written as the product of a scalar and a common matrix, i.e., $A_i^{--} = c_i K$, for $2 \leq i \leq N$. As highlighted in Remark 2.1, this structure arises, for instance, in the batch queues introduced in Section

2.3. Using this assumption we can rewrite Equation (2.7) as

$$G_0 - (-A_1^{--})^{-1} K G_0 \sum_{i=2}^{N} c_i G_+^{i-1} = (-A_1^{--})^{-1} \sum_{i=0}^{N} A_i^{-+} G_+^i. \tag{2.11}$$

This is also a Sylvester matrix equation of the type $AXB + X = C$, with $A = (A_1^{--})^{-1}K$ and $B = \sum_{i=2}^{N} c_i G_+^{i-1}$. To solve this equation, the Hessenberg-Schur method [47] relies on the computation of a Hessenberg decomposition of the matrix A and a Schur decomposition of the matrix B. These operations transform the problem into the solution of a quasi-triangular system that can be solved efficiently by forward substitution. A brief description of this solution method is given in Appendix A.4, and the details can be found in [47].

2.2 Computing the stationary probability vector

We have shown how the matrix G can be obtained by first computing G_+ from a censored process and then solving a linear system to determine G_0. Once G has been obtained the stationary probability vector of the original M/G/1-type MC can be computed by means of Ramaswami's formula [97], as follows. Let the matrices $(\tilde{A}_i)_{i\geq 1}$ and $(\tilde{B}_i)_{i\geq 0}$ be defined as

$$\tilde{A}_i = \sum_{j=i}^{\infty} A_j G^{j-i},\ i \geq 1, \quad \tilde{B}_i = \sum_{j=i}^{\infty} B_j G^{j-i},\ i \geq 1, \quad \tilde{B}_0 = B_0 + \tilde{B}_1(-\tilde{A}_1)^{-1} C_0. \tag{2.12}$$

Let π be the stationary probability vector of the MC with generator Q, i.e., the vector such that $\pi Q = 0$ and $\pi e = 1$. This vector can also be partitioned according to the levels as $\pi = [\pi_0, \pi_1, \dots]$. Then, Ramaswami's formula states that the vectors π_i can be found recursively as

$$\pi_i = \left(\pi_0 \tilde{B}_i + \sum_{j=1}^{i-1} \pi_j \tilde{A}_{i-j+1} \right) (-\tilde{A}_1)^{-1}, \quad i \geq 1, \tag{2.13}$$

and π_0 is such that

$$\pi_0 \tilde{B}_0 = 0, \quad \pi_0 \kappa = 1, \tag{2.14}$$

where $\kappa = e + \left(\sum_{j=1}^{\infty} \tilde{B}_j\right)\left(-\sum_{j=1}^{\infty} \tilde{A}_j\right)^{-1} e$. However, one may wonder whether the structure of the matrix A_0 can be exploited to make the computation of π faster. This is the purpose of this section, where three main approaches to compute π are described. The first method makes use of the structure of A_0 to accelerate the computation of the matrices $(\tilde{A}_i)_{i=1}^N$ and $(\tilde{B}_i)_{i=1}^N$. The other two methods compute the vector π from the stationary probability vector of the censored process. While the second method applies in general for any M/G/1-type MC with restricted downward transitions, the last one exploits the additional structure discussed in Remark 2.1, which applies for the MCs that describe the batch queues to be introduced in Section 2.3.

2.2.1 Computing π from the original process

In this section we consider the problem of exploiting the structure of A_0 to compute the matrices $(\tilde{A}_i)_{i=1}^N$ and $(\tilde{B}_i)_{i=1}^N$. To compute these matrices one typically starts with $\tilde{A}_N = A_N$ and $\tilde{B}_N = B_N$, and computes iteratively $\tilde{A}_i = A_i + \tilde{A}_{i+1}G$ and $\tilde{B}_i = B_i + \tilde{B}_{i+1}G$, for $i = N-1, \ldots, 1$. However, if we rewrite Equation (2.12) in block form according to the partition in (2.3), we find that

$$\begin{aligned}\tilde{A}_i = \begin{bmatrix} \tilde{A}_i^{++} & \tilde{A}_i^{+-} \\ \tilde{A}_i^{-+} & \tilde{A}_i^{--} \end{bmatrix} &= \begin{bmatrix} A_i^{++} & A_i^{+-} \\ A_i^{-+} & A_i^{--} \end{bmatrix} + \sum_{j=i+1}^{N} \begin{bmatrix} A_j^{++} & A_j^{+-} \\ A_j^{-+} & A_j^{--} \end{bmatrix} \begin{bmatrix} G_+ & 0 \\ G_0 & 0 \end{bmatrix}^{j-i} \\ &= \begin{bmatrix} A_i^{++} & A_i^{+-} \\ A_i^{-+} & A_i^{--} \end{bmatrix} + \sum_{j=i+1}^{N} \begin{bmatrix} A_j^{++}G_+^{j-i} + A_j^{+-}G_0G_+^{j-i-1} & 0 \\ A_j^{-+}G_+^{j-i} + A_j^{--}G_0G_+^{j-i-1} & 0 \end{bmatrix},\end{aligned}$$

for $1 \le i \le N$. Therefore, the last $m-r$ columns of the matrix $\tilde{A}_i$ are identical to those of the matrix A_i, for $1 \le i \le N$. To compute the first r columns of these matrices we simply start with $\tilde{A}_N^{++} = A_N^{++}$ and $\tilde{A}_N^{-+} = A_N^{-+}$ and then iteratively compute

$$\begin{aligned}\tilde{A}_i^{++} &= A_i^{++} + \tilde{A}_{i+1}^{++}G_+ + A_{i+1}^{+-}G_0, \\ \tilde{A}_i^{-+} &= A_i^{-+} + \tilde{A}_{i+1}^{-+}G_+ + A_{i+1}^{--}G_0,\end{aligned}$$

for $i = N-1, \ldots, 1$. A similar observation can be made for the matrices $(\tilde{B}_i)_{i=1}^N$ since these, as well as $(B_i)_{i=1}^N$, can be partitioned in two blocks depending on whether the transitions from level zero take the chain to a state with phase in $\mathcal{S}^+$ or $\mathcal{S}^-$. Therefore, we can write

$$\tilde{B}_i = \begin{bmatrix} \tilde{B}_i^+ & \tilde{B}_i^- \end{bmatrix} = \begin{bmatrix} B_i^+ & B_i^- \end{bmatrix} + \sum_{j=i+1}^{N} \begin{bmatrix} B_j^{++} & B_j^- \end{bmatrix} \begin{bmatrix} G_+ & 0 \\ G_0 & 0 \end{bmatrix}^{j-i}.$$

From this equation we find that $\tilde{B}_i^- = B_i^-$, for $1 \le i \le N$. Also, $\tilde{B}_i^+$ can be obtained by starting with $\tilde{B}_N^+ = B_N^+$, and sequentially computing $\tilde{B}_i^+ = B_i^+ + \tilde{B}_{i+1}^+G_+ + B_{i+1}^-G_0$, for $i = N-1, \ldots, 1$. Once we have obtained these matrices, we can compute the vector π directly using Ramaswami's formula as in Equation (2.13). One could also rely on the fast-Fourier-transform-based implementation of Ramaswami's formula proposed by Meini [84]. With either approach, we are computing π by simply using the structure of G to speed up the computation of the matrices $(\tilde{A}_i)_{i=1}^N$ and $(\tilde{B}_i)_{i=1}^N$. In the next section we consider a different approach in which we make use of the censored process to obtain the stationary probability vector π.

2.2.2 Computing π from the censored process

In this section we show how the censored process can be used to reduce the time required to compute π by separately computing $(\pi_i^+)_{i\ge1}$ and $(\pi_i^-)_{i\ge1}$, where these vectors arise by partitioning π_i according to $\mathcal{S}^+$ and $\mathcal{S}^-$, i.e., $\pi_i = [\pi_i^+ \ \pi_i^-]$, for $i \ge 1$. As stated before, we obtain the matrix G_+ from a censored process of the M/G/1-type whose G matrix is actually equal to G_+. To do so, we described the censored process by means

of the blocks $(\bar{A}_i)_{i=0}^{M}$, which only consider the behavior of the process away from the boundary. We now complete the description of this process by adding a boundary level, which is identical to the boundary level of the original process. This means that the new process is built by observing the original process only when it is in the subset of states $\bigcup_{i\geq 1}\{(i,j), j \in \mathcal{S}^+\} \cup \{(0,j), 1 \leq j \leq m_b\}$. The boundary behavior of this process is described by the $m_b \times m_b$ matrix $\bar{B}_0$ (transitions within level 0), the $m_b \times r$ matrices $(\bar{B}_i)_{i=1}^{M_0}$ (transitions from level zero to level $i > 1$) and the $r \times m_b$ matrix $\bar{C}_0$ (transitions from level 1 to level 0). These blocks, together with the matrices $(\bar{A}_i)_{i=0}^{M}$, completely characterize the censored process, whose generator $\bar{Q}$ is built from these blocks in a similar manner as the generator Q of the original process in (2.1).

To define the blocks $(\bar{B}_i)_{i=0}^{M_0}$ we rely on the matrices $(W_i)_{i\geq -1}$ as defined in (2.6). As stated in the previous section, we partition the original blocks $B_i = [B_i^+ \; B_i^-]$, where B_i^+ (resp. B_i^-) holds the transition rates that, from level zero, trigger the chain into level i and phase in $\mathcal{S}^+$ (resp. $\mathcal{S}^-$). Similarly, the matrix C_0 can be partitioned into C_0^+ and C_0^-, which hold the transition rates to level zero from states in level one and phase in $\mathcal{S}^+$ and $\mathcal{S}^-$, respectively. Using these definitions we find that $\bar{B}_0 = B_0 + B_1^-(-A_1^{--})^{-1}C_0^-$, which considers the two alternative paths for a transition starting and ending in level zero in the new process. This may occur directly according to B_0 or by a transition to level one and a transition backward, avoiding states with phase in $\mathcal{S}^+$. Carrying out a similar analysis we find that the remaining boundary blocks are given by

$$\bar{B}_i = \begin{cases} B_i^+ + \sum_{j=1}^{i+1} B_j^- W_{i-j}, & 1 \leq i \leq N-1, \\ B_N^+ + \sum_{j=1}^{N} B_j^- W_{N-j}, & i = N, \\ \sum_{j=1}^{N} B_j^- W_{i-j}, & i \geq N+1. \end{cases} \tag{2.15}$$

These matrices are computed sequentially from $i = 0$ to M_0, where M_0 is the smallest positive integer such that $\sum_{i=0}^{M_0} \bar{B}_i e > -\epsilon e$. Finally, we observe that the transitions from level one to level zero in the censored process are governed by $\bar{C}_0 = C_0^+ + A_1^{+-}(-A_1^{--})^{-1}C_0^-$.

With these definitions we are ready to compute, using Ramaswami's formula, the stationary probability vector of the censored process $\bar{\pi}$, which can be partitioned in blocks as $\bar{\pi} = [\bar{\pi}_0, \; \bar{\pi}_1, \bar{\pi}_2, \ldots]$. The importance of this vector is that it is proportional to the stationary vector π of the original process [66, 76], i.e., $\bar{\pi}_0 \propto \pi_0$ and $\bar{\pi}_i \propto \pi_i^+$, for $i \geq 1$. Actually, we would like to normalize the vector $\bar{\pi}$ by a constant such that $\bar{\pi}_0 = \pi_0$ and $\bar{\pi}_i = \pi_i^+$, for $i \geq 1$. This can be accomplished by computing π_0 as in Equation (2.14) and assigning $\bar{\pi}_0 = \pi_0$. This forces $\bar{\pi}_0$ to be normalized by the appropriate constant. The terms $(\bar{\pi}_i)_{i\geq 1}$ can then be obtained, using Ramaswami's formula, after computing the matrices $(\tilde{\bar{A}}_i)_{i\geq 1}$ and $(\tilde{\bar{B}}_i)_{i\geq 1}$ for the censored process, as in Equation (2.12). Notice, to obtain π_0 from Equation (2.14) we first need to compute the matrices $(\tilde{A}_i)_{i=1}^{N}$ and $(\tilde{B}_i)_{i=1}^{N}$.

Up to this point, we have computed π_0 and $(\pi_i^+)_{i\geq 1}$, and we can use these to find the vectors $(\pi_i^-)_{i\geq 1}$ in order to completely determine π. This can be done by rewriting Equation (2.13) in block form as

$$\begin{bmatrix}\pi_i^+ & \pi_i^-\end{bmatrix}\begin{bmatrix}-\tilde{A}_1^{++} & -\tilde{A}_1^{+-} \\ -\tilde{A}_1^{-+} & -\tilde{A}_1^{--}\end{bmatrix} = \pi_0\begin{bmatrix}\tilde{B}_i^+ & \tilde{B}_i^-\end{bmatrix} + \sum_{j=1}^{i-1}\begin{bmatrix}\pi_j^+ & \pi_j^-\end{bmatrix}\begin{bmatrix}\tilde{A}_{i-j+1}^{++} & \tilde{A}_{i-j+1}^{+-} \\ \tilde{A}_{i-j+1}^{-+} & \tilde{A}_{i-j+1}^{--}\end{bmatrix}, \quad i \geq 1.$$

From this equation we can express π_i^- as

$$\pi_i^- = \left(\pi_0 B_i^- + \pi_i^+ A_1^{+-} + \sum_{j=1}^{i-1} \left(\pi_j^+ A_{i-j+1}^{+-} + \pi_j^- A_{i-j+1}^{--} \right) \right) (-A_1^{--})^{-1}, \quad i \geq 1. \quad (2.16)$$

Notice that this equation makes use of the original blocks $\{A_i\}_{i\geq 1}$ and $\{B_i\}_{i\geq 1}$ instead of the blocks $\{\tilde{A}_i\}_{i\geq 1}$ and $\{\tilde{B}_i\}_{i\geq 1}$ since their last $m-r$ columns are identical, as shown in Section 2.2.1. Using this equation we can compute π_i^- in terms of π_0, $(\pi_j^+)_{j=1}^i$ and $(\pi_j^-)_{j=1}^{i-1}$, for $i \geq 1$, thus concluding the computation of π. This approach is summarized in Algorithm 2.1. One may consider a slight modification to this algorithm by replacing the use of Ramaswami's formula in step 7 with the Fast Ramaswami's Formula (FRF) proposed in [84], which is based on the fast Fourier transform. This may provide an important computational gain when the number of blocks $(\bar{B}_i)_{i\geq 1}$ and $(\bar{A}_i)_{i\geq 1}$ is large. This approach will also be considered in the numerical experiments in Section 2.3.

Algorithm 2.1 Computing π from the censored process

Input: Matrices $(\bar{A}_i)_{i\geq 1}$ and G

1: Compute the matrices $(\tilde{A}_i)_{i\geq 1}$ and $(\tilde{B}_i)_{i\geq 0}$ using the iterative scheme in Section 2.2.1.
2: Find π_0 solving Equation (2.14) and set $\bar{\pi}_0 = \pi_0$.
3: Compute the matrices $(\bar{B}_i)_{i\geq 0}$ according to Equation (2.15).
4: Compute the matrices $(\tilde{\bar{A}}_i)_{i\geq 1}$ and $(\tilde{\bar{B}}_i)_{i\geq 0}$ as in Equation (2.12).
5: Set $\sigma = \bar{\pi}_0 e$ and $i = 1$.
6: **while** $\sigma < 1 - \epsilon$ **do**
7: Compute $\bar{\pi}_i$ using (Fast) Ramaswami's formula.
8: Set $\pi_i^+ = \bar{\pi}_i$ and compute π_i^- as in Equation (2.16).
9: Update $\sigma = \sigma + \pi_i^+ e + \pi_i^- e$, and $i = i + 1$.
10: **end while**

2.2.3 Computing π from the censored process under additional structure

In this section we consider how to exploit the additional structure described in Remark 2.1, which arises in the MCs that describe the batch queues to be introduced in Section 2.3. Recall that this additional structure implies that: first, the matrices $(A_i^{--})_{i=2}^N$ can be written as the product of a scalar and a common matrix, i.e., $A_i^{--} = c_i K$, for $2 \leq i \leq N$; second, the blocks $(A_i^{+-})_{i=2}^N$ and $(A_i^{-+})_{i=2}^N$ are equal to zero. We start by noting that, thanks to the additional structure, Equation (2.16) can be written as

$$\pi_i^- = \left[\pi_0 B_i^- + \pi_i^+ A_1^{+-} + \left(\sum_{j=1}^{i-1} c_{i-j+1} \pi_j^- \right) K \right] (-A_1^{--})^{-1}, \quad i \geq 1. \quad (2.17)$$

The advantage of this expression is that each term of the sum is just a vector multiplied by a scalar, while in the general case each term in the sum requires a vector-matrix

multiplication. Therefore, one can simply modify step 8 in Algorithm 2.1 replacing the use of Equation (2.16) by Equation (2.17). However there is an additional fact that can be exploited if we make use of the Fast Ramaswami's Formula (FRF) to compute $(\pi_i^+)_{i\geq 1}$. The FRF, as introduced in [84], computes not one but many vectors π_i^+ at a time. Specifically, in each iteration the FRF computes $\bar{M}$ terms, where $\bar{M}$ is a power of two such that $\bar{M} \geq M$ and $\bar{M} \geq M_0$. Therefore, after the first iteration we already have the terms $(\pi_i^+)_{i=1}^{\bar{M}}$ and it suffices to compute the corresponding terms $(\pi_i^-)_{i=1}^{\bar{M}}$. In the case of the MCs with the structure described in Remark 2.1 we could do this by using Equation (2.17) directly. But we can also use the following observation. Let us assume that we have computed the first $(\pi_i^-)_{i=1}^{N}$ by means of (2.17). For $i \geq N+1$, we can write

$$\pi_i^- = \left(\pi_i^+ A_1^{+-} + \sum_{j=i-N+1}^{i-1} \left(c_{i-j+1}\pi_j^-\right) K \right) (-A_1^{--})^{-1},$$

as $B_i = 0$ and $A_i = 0$ for $i > N$. Now, if we focus on the computation of $(\pi_i^-)_{i=N+1}^{2N-1}$ we can write the previous expression as

$$\pi_i^- = \pi_i^+ \hat{A}_1^{+-} + \sum_{j=i-N+1}^{N} \left(c_{i-j+1}\pi_j^-\right) \hat{K} + \sum_{j=N+1}^{i-1} \left(c_{i-j+1}\pi_j^-\right) \hat{K}, \;\; N+1 \leq i \leq 2N-1, \tag{2.18}$$

where $\hat{A}_1^{+-} = A_1^{+-}(-A_1^{--})^{-1}$ and $\hat{K} = K(-A_1^{--})^{-1}$. In this equation the first sum in the right-hand side only involves the terms $(\pi_i^-)_{i=2}^{N}$, while the second sum and the left-hand side involve the terms $(\pi_i^-)_{i=N+1}^{2N-1}$. Therefore, we can write this expression as a system of equations. To do so, let $\bar{N} = N-1$ and let $\hat{\pi}_i^+$ and $\hat{\pi}_i^-$ be

$$\hat{\pi}_i^+ = \begin{bmatrix} \pi^+_{1+(i-2)\bar{N}+1} \\ \pi^+_{1+(i-2)\bar{N}+2} \\ \vdots \\ \pi^+_{1+(i-1)\bar{N}} \end{bmatrix} \text{ and } \hat{\pi}_i^- = \begin{bmatrix} \pi^-_{1+(i-2)\bar{N}+1} \\ \pi^-_{1+(i-2)\bar{N}+2} \\ \vdots \\ \pi^-_{1+(i-1)\bar{N}} \end{bmatrix}, \; i \geq 2.$$

Therefore, we can write the set of $\bar{N}$ equations in (2.18) in matrix form as

$$\hat{\pi}_3^- = \hat{\pi}_3^+ \hat{A}_1^{+-} + R_1 \hat{\pi}_2^- \hat{K} + R_2 \hat{\pi}_3^- \hat{K},$$

where the $\bar{N} \times \bar{N}$ matrices R_1 and R_2 are given by

$$R_1 = \begin{bmatrix} c_N & c_{N-1} & \dots & c_2 \\ 0 & c_N & \dots & c_3 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & c_N \end{bmatrix} \text{ and } R_2 = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ c_2 & 0 & \dots & 0 & 0 \\ c_3 & c_2 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{N-1} & c_{N-2} & \dots & c_2 & 0 \end{bmatrix}.$$

In general, we can find the vector $\hat{\pi}_i^-$ in terms of $\hat{\pi}_{i-1}^-$ and $\hat{\pi}_i^+$ by solving the system

$$\hat{\pi}_i^- - R_2 \hat{\pi}_i^- \hat{K} = \hat{\pi}_i^+ \hat{A}_1^{+-} + R_1 \hat{\pi}_{i-1}^- \hat{K}, \quad i \geq 3. \tag{2.19}$$

This is also a Sylvester matrix equation of the type $AXB + X = C$, that can be solved using the Hessenberg-Schur method mentioned before [47]. Among the two decompositions that must be carried out to use this method, the Schur decomposition is the most expensive, requiring $10b^3$ operations for a square matrix of size b. On the other hand, the Hessenberg decomposition only requires $\frac{5}{3}b^3$ operations [47]. In this case the square matrix $A = -R_2$ is of size $\bar{N}$, while the square matrix $B = \hat{K}$ is of size $m - r$. Since we expect a large value for $m-r$, it is actually better to apply this method to the transposed system, i.e., to $B'X'A' + X' = C'$. In this manner we apply the Hessenberg decomposition to the larger matrix, i.e., to $B' = -\hat{K}'$. Additionally, since the matrix $A' = -R_2'$ is already in real Schur form [48] (it is upper triangular), there is no need to use the QR algorithm to obtain the Schur decomposition. Therefore, after finding the Hessenberg decomposition of $B' = -\hat{K}'$ we can find the solution to the transposed Sylvester matrix equation by solving $\bar{N}$ Hessenberg systems of size $m - r$, each one requiring $O((m - r)^2)$ time. Moreover, to find all the vectors $(\hat{\pi}_i^-)_{i\geq 3}$ we need to solve a series of equations as in (2.19), but the matrices R_2 and $\hat{K}$ remain unchanged for $i \geq 3$. Therefore, to solve these equations we only need to apply the Hessenberg decomposition once and update the right-hand side of (2.19). This concludes the present section on the computation of π. The next section illustrates the performance of the various methods introduced in this chapter by means of two queueing examples.

2.3 Examples and Numerical Experiments

The purpose of this section is to show how two different queueing systems can be modeled as M/G/1-type MCs with restricted downward transitions. The first system is a BMAP/PH/1 queue where the block A_0 is forced to have a few nonzero columns by using a slightly larger representation of the service-time distribution. The second example is a BMAP[2]/PH[2]/1 preemptive priority queue with two types of customers where, by adequately ordering the state space, we obtain the desired structure for the block A_0. These two examples will be used to illustrate the computational gains obtained by exploiting this structure. In addition, when the batch sizes are i.i.d. random variables independent of the arrival process, the MCs used to model these queues show the structure described in Remark 2.1. Therefore, these examples will also be useful to demonstrate the gains obtained by exploiting this additional structure.

We focus on the total time required to compute the vector π, which includes the computation of G. As a benchmark, we compute the matrix G of the original M/G/1-type MC using Cyclic Reduction (CR) and use the label **B** to refer to this approach for both queues. To compute π, after finding G, we consider Ramaswami's formula (RF) and its fast implementation (FRF). Therefore, for the total time to compute π we have two benchmarks: **B-RF** and **B-FRF**. For the BMAP[2]/PH[2]/1 preemptive priority queue we also compare with the approach introduced in [129], here labeled **INF**, where both the high- and the low-priority buffers are assumed to be infinite. The methods introduced in this chapter are labeled **O**, **C** and **C***. In all the methods the censored process is used to exploit the structure of the matrix A_0 to compute G_+. In the methods labeled **O** and **C** we only consider this structure, while in **C*** we also take into account the additional

structure described in Remark 2.1. Therefore, in the methods **O** and **C** the matrix G_0 is found by solving the linear system (2.7) for the general case. In the method **O**, the vector π is found from the original process as described in Section 2.2.1. This can be done either with RF or FRF and therefore we have both **O-RF** and **O-FRF**. On the other hand, for the method **C** the vector π is computed by means of the censored process as discussed in Section 2.2.2. In this case we also have both **C-RF** and **C-FRF**. In the method labeled **C*** we consider the additional structure in Remark 2.1, and therefore the matrix G_0 can be found by solving Equation (2.11). Additionally, we can compute π relying on the censored process and applying Equation (2.17). As this can be done by using either RF or FRF to compute $(\pi_i^+)_{i\geq 1}$, we have the two alternatives **C*-RF** and **C*-FRF**. Moreover, if FRF is used to compute $(\pi_i^+)_{i\geq 1}$, we can also make use of Equation (2.19) to obtain $(\pi_i^-)_{i\geq 1}$. This approach will be labeled **C*-FRF$_2$**.

2.3.1 The BMAP/PH/1 queue

Our first example is a single-server queue where the customers arrive according to a BMAP characterized by the $m_a \times m_a$ matrices $\{D_0, D_1, \dots D_{\bar{L}}\}$, with $\bar{L}$ the maximum batch size. The service times follow a PH distribution with parameters (m_s, α, T). For details on the BMAP and the PH distributions see Appendix A.1. The BMAP/PH/1 queue can be modeled as an M/G/1-type MC by choosing the number of customers in the queue to be the level. This selection assures that the level decreases by at most one in a single transition since only one service completion can occur at a time. On the other hand the level can increase by up to $\bar{L}$ as it is triggered by a batch arrival. Let $N(t)$ be the number of customers in the queue at time t, $S(t)$ the phase of the service-time distribution at time t if there is a customer in service, and $J(t)$ the phase of the arrival process at time t. Then, the tuple $\{N(t), S(t), J(t), t \geq 0\}$ forms a CTMC that fully describes the state of the BMAP/PH/1 queue. The state space of this MC can be described as follows: the level zero is the set of states $\Omega_0 = \{(0, j), 1 \leq j \leq m_a\}$, where in state $(0, j)$ the queue is empty and the arrival process is in phase j; the level $k \geq 1$ is the set of states $\Omega_k = \{(k, i, j), 1 \leq i \leq m_s, 1 \leq j \leq m_a\}$, where in state (k, i, j) there are k customers in the queue, the service in progress is in phase i and the arrival process is in phase j. The complete state space is therefore given by $\Omega = \bigcup_{k\geq 0} \Omega_k$. Since this MC is of the M/G/1 type, its rate matrix has the structure in (2.1) with blocks given by

$$A_0 = t\alpha \otimes I_{m_a}, \quad A_1 = T \oplus D_0, \quad A_{j+1} = I_{m_s} \otimes D_j, \ j = 1, \dots, \bar{L}, \tag{2.20}$$

where $t = -Te$, and I_n is the identity matrix of size n. Here $\otimes$ and $\oplus$ stand for Kronecker product and sum [48], respectively. From this definition it is clear that the block size is $m = m_s m_a$ and that the number of nonzero columns in A_0 depends on the number of nonzero elements in the vector α. In fact, the definition of A_0 is the same as the one used in the previous chapter, Section 1.4.2, to describe a MAP/PH/1 queue by means of a QBD MC. The difference here is that we are considering batch arrivals instead of the single arrivals in the previous chapter, and therefore the queue is modeled as an M/G/1-type MC, instead of a QBD MC. As the structure of A_0 is the same as in the previous chapter, we can also make use of Theorem 1.1 to obtain an M/G/1-type MC of slightly

larger block size, but where the matrix A_0 has only m_a nonzero columns. Recall that Theorem 1.1 states that any CPH distribution with representation (m_s, α, T) also has a representation $(m_s + 1, e_1, \bar{T})$, where e_1 and $\bar{T}$ are given by

$$e_1 = [1\ 0_{m_s}] \quad \text{and} \quad \bar{T} = \begin{bmatrix} -\lambda & \lambda\alpha P \\ 0 & T \end{bmatrix}.$$

Here λ is the diagonal entry of T of largest absolute value, i.e., $\lambda = \max\{|T_{ii}|, 1 \leq i \leq m_s\}$, and P is the uniformized version of the sub-generator matrix T, i.e., $P = \frac{1}{\lambda}T + I$. Using this result we can replace α and T by e_1 and $\bar{T}$, respectively, in Equation (2.20). As a consequence the block A_0 has only m_a nonzero columns, and the new block size is $(m_s + 1)m_a$. If m_s is large, the structure of the block A_0 can be exploited to speed up the computation of the matrix G and the stationary probability vector of the chain.

In addition, the structure described in Remark 2.1 arises if the batch sizes are i.i.d. random variables independent of the arrival process. Let r_i be the probability that a batch is of size i, for $1 \leq i \leq \bar{L}$, and let the times between batch arrivals be described by a MAP characterized by (m_a, D_0, D_+). The matrices $\{D_1, \ldots, D_{\bar{L}}\}$ of the BMAP are thus given by $D_i = r_i D_+$, for $1 \leq i \leq \bar{L}$. Using these parameters for the BMAP we find that $A^{--}_{i+1} = I_{m_s} \otimes D_i = r_i\,(I_{m_s} \otimes D_+)$, for $1 \leq i \leq \bar{L}$. Therefore the matrices $(A^{--}_i)_{i=2}^{\bar{L}}$ can be written as the product of a common matrix $I_{m_s} \otimes D_+$ and a set of scalars $(r_i)_{i=2}^{\bar{L}}$. Furthermore, from Equation (2.20) we also see that $(A^{+-}_i)_{i=2}^{\bar{L}}$ and $(A^{-+}_i)_{i=2}^{\bar{L}}$ are all zero.

To illustrate the behavior of our methods we choose the following PH distribution as service time: we assume that the total service time is made up of a random number of i.i.d. elementary operations; each of these operations is described by a continuous PH distribution with parameters $(\bar{m}_s, \gamma, C)$; the number of elementary operations that make up the total service time is described by a discrete PH distribution with parameters (n_s, β, S); as a result [76], the total service time is a continuous PH distribution with parameters (m_s, α, T) given by $m_s = n_s\bar{m}_s$, $\alpha = \gamma \otimes \beta$ and $T = C \otimes I + c\gamma \otimes S$, where $c = -Ce$. Notice that we can easily alter the size of this representation by changing n_s, the size of the discrete PH distribution. Also, this representation is not acyclic, but we can obtain a (slightly larger) representation where the initial probability vector has only one nonzero entry, as mentioned above.

The time to complete an elementary operation is assumed to have mean one and squared coefficient of variation (SCV) equal to two. The moments are matched with a PH distribution of size $\bar{m}_s = 2$ using the method in [120]. Hence, the service time distribution has a representation of size $2n_s$, which we convert into a representation of size $2n_s+1$ to induce the block A_0 to have m_a nonzero columns. The number of elementary service operations is assumed to be uniformly distributed between one and n_s. Therefore the mean service time is equal to $(n_s+1)/2$, which is therefore fixed when n_s is specified. In addition to quantify the effect of n_s (the block size) on the computation times, we also consider the effect of the load. For this queue the load is given by $\rho = \lambda E[L]/\mu$, where λ is the mean arrival rate (inverse of the mean IAT), μ is the mean service rate (inverse of the mean service time) and $E[L]$ is the mean batch size. We set the batch size distribution to be uniform between one and $\bar{L}$, hence $E[L] = (\bar{L} + 1)/2$. As μ and $E[L]$ are fixed by specifying the values of n_s and $\bar{L}$, we use the rate λ to match a determined

load ρ. The arrival process is assumed to have rate $\lambda = \rho\mu/E[L]$, SCV equal to 5 and decay rate of the autocorrelation function of the sequence of IATs equal to 0.5. These characteristics are matched with a MAP(m_a, D_0, D_+) of order $m_a = 2$ with the method introduced in [40]. The matrices $\{D_1, \dots D_{\bar{L}}\}$ are obtained as $D_j = r_j D_+$, for $1 \le j \le \bar{L}$, where $r_j = 1/\bar{L}$ is the probability that a batch is of size j. Since $m_a = 2$, the block size of the original chain is $4n_s$ and the number of nonzero columns in A_0 is only two.

ρ	B-RF	B-FRF	O-RF	O-FRF	C-RF	C-FRF	C*-RF	C*-FRF	C*-FRF$_2$
0.1	1.6	1.6	0.2	0.2	0.3	0.3	0.4	0.3	0.4
0.3	2.6	2.7	0.1	0.2	0.2	0.1	0.2	0.1	0.1
0.5	3.5	3.6	0.2	0.3	0.3	0.3	0.3	0.2	0.1
0.7	5.3	5.4	0.5	0.9	1.4	0.8	0.9	0.3	0.3
0.9	13.7	8.6	8.4	2.7	11.6	8.2	10.6	7.2	1.7

Table 2.1: Computation times (sec) for $\bar{L} = 10$, $n_s = 10$

In the first scenario we set both the maximum batch size $\bar{L}$ and the maximum number of elementary service operations n_s equal to 10. The results for different values of the load are shown in Table 2.1. In this case the use of the FRF provides an important gain for the **B**, the **O**, the **C** and the **C*** methods. However, compared to the **B-RF** method, the reduction that can be achieved by using the **O-RF** method is significantly larger than the one that can be obtained by using the **B-FRF** method. In particular, we see that the **O-RF** method provides an important gain for all the load values considered, while the use of the FRF provides a gain for high loads only. Also, we observe that in this case the performance of **C-RF** and **C*-RF** is actually worse than the simpler **O-RF**, with a larger difference for higher loads. In fact, **O-RF** is the best method among those that do not consider the structure in Remark 2.1. We will comment more on this later on, after presenting the results for other scenarios. The best method is **C*-FRF$_2$**, which outperforms all the other methods for every value of the load considered, except $\rho = 0.1$. This table shows that even for $n_s = 10$ the methods introduced in this chapter are able to significantly reduce the times to compute the vector π.

ρ	B-RF	B-FRF	O-RF	O-FRF	C-RF	C-FRF	C*-RF	C*-FRF	C*-FRF$_2$
0.1	129.2	132.6	0.5	3.9	0.4	0.4	0.3	0.3	0.4
0.3	145.3	149.6	0.5	5.5	0.5	0.5	0.4	0.3	0.4
0.5	201.4	210.8	1.0	10.4	1.1	1.0	0.6	0.5	0.6
0.7	240.9	264.3	4.3	27.5	4.9	4.3	2.9	2.4	1.5
0.9	388.3	447.4	55.3	113.3	58.7	55.5	49.2	46.4	10.8

Table 2.2: Computation times (sec) for $\bar{L} = 10$, $n_s = 50$

We now consider a scenario where the computation times become considerably larger due to an increase in the block size, as this is the case where we expect our methods to

provide more important gains. For the results shown in Table 2.2 we take the previous scenario and simply increase the value of n_s to 50. This makes the block size equal to 200, while the number of nonzero columns in A_0 remains equal to two. In this case we notice that the use of the FRF no longer provides a gain for the **B** nor the **O** methods. This is related to the increase in the block size, since the computational cost of the FRF is more sensitive to the block size than the RF [84]. On the other hand, we see how the **O-RF** method shows a dramatical reduction in computation times, especially under low and mid loads. In this case, the **C*-RF** method provides further improvements although this reduction is rather limited for high loads. Even the use of **C*-FRF** does not provide a substantial reduction under high loads, compared to **C-RF**. The only method that is able to show a significant reduction for $\rho = 0.9$ is **C*-FRF**$_2$, which again outperforms all the other methods. Here, as well as in the previous case and for most of the load values considered, we find that the **C** approach is not able to reduce the times obtained with the **O** method. In this scenario we also observe important absolute differences: while the **B** approach may take between two and four minutes to compute π under low to mid loads, the best of our methods requires less than two seconds for the same computation; also, for a load of 90% the solution by general techniques requires more than 6 minutes while our best method takes around 10 seconds.

ρ	B-RF	B-FRF	O-RF	O-FRF	C-RF	C-FRF	C*-RF	C*-FRF	C*-FRF$_2$
0.1	196.7	203.8	1.8	9.0	1.6	1.9	0.5	0.5	0.6
0.3	251.7	261.7	3.1	12.8	3.2	2.8	0.8	0.7	0.8
0.5	291.0	311.1	8.2	23.5	7.5	7.1	1.8	1.4	1.3
0.7	393.7	424.4	34.3	62.5	35.7	34.5	11.0	9.3	3.6
0.9	740.5	717.5	285.0	252.0	210.0	200.6	182.2	172.9	28.2

Table 2.3: Computation times (sec) for $\bar{L} = 20$, $n_s = 50$

As a final scenario we consider an increase in the maximum batch size, making $\bar{L}$ equal to 20, while the other parameters are kept as in the previous scenario. The results are shown in Table 2.3, where the first obvious observation is that all the methods require longer computation times than in the previous scenario. Although there is a small increase in the time to compute the matrix G, the difference between these scenarios is mostly due to the significantly longer times required to compute π. In this case the number of blocks $\bar{L}$ and the number of terms π_i are larger, but the block size is the same as in the previous case. Therefore, the FRF-based methods, compared to their RF-based counterparts, show a relative better performance than before. However, for the **B** and the **O** approaches the use of the FRF only provides a computational gain when the load is 0.9. On the other hand, the use of the FRF under the **C** and **C*** approaches results in reductions for almost all the load values considered. Moreover, the **C*-FRF**$_2$ method is able to significantly reduce the computation times for high loads, outperforming the other approaches once more. We also notice that in this case the **C-RF** and the **O-RF** methods show a similar performance, except when the load is very high (0.9). In this latter case

the **C-RF** method requires around 25% less time than **O-RF**. If we consider their FRF versions, we find that **C-FRF** is always better than **O-FRF** and, moreover, **C-FRF** is the best among the **O** and the **C** approaches. This is in contrast with the previous results, where **O-RF** was typically the best performing among these four methods.

2.3.2 The BMAP[2]/PH[2]/1 preemptive priority queue

We now illustrate how the BMAP[2]/PH[2]/1 preemptive priority queue can be modeled as an M/G/1-type MC with restricted downward transitions. In this queue the arrivals also occur in batches and there are two types of customers, each one associated with a different priority level. We assume that each batch is made of customers of only one type, and the maximum size of a batch of high- (resp. low-) priority customers is $\bar{L}_1$ (resp. $\bar{L}_2$). Hence the arrival process can be modeled as a BMAP[2] characterized by the matrices $\{D_0, D_1^{j_1}, D_2^{j_2}, 1 \leq j_1 \leq \bar{L}_1, 1 \leq j_2 \leq \bar{L}_2\}$. In this case the matrix D_1^j (resp. D_2^j) holds the transition rates associated with the arrival of a batch of j high- (resp. low-) priority customers, for $1 \leq j \leq \bar{L}_1$ (resp. $1 \leq j \leq \bar{L}_2$). As before, we can define a version of this process where the batch sizes are i.i.d. random variables and are independent of the IATs. For this purpose let p_i be the probability that a batch of high-priority customers is of size i, for $1 \leq i \leq \bar{L}_1$. Similarly, let q_i be the probability that a batch of low-priority customers is of size i, for $1 \leq i \leq \bar{L}_2$. Also, let the batch IATs be described by a MAP with markings for each customer type, i.e., characterized by the $m_a \times m_a$ matrices $\{D_0, D_1, D_2\}$. Here the matrix D_1 (resp. D_2) holds the rates at which the underlying chain triggers the arrival of a batch of high- (resp. low-) priority customers. Combining the marked MAP and the batch-size distributions we obtain a BMAP[2] characterized by the $m_a \times m_a$ matrices $\{D_0, D_1^{j_1}, D_2^{j_2}, 1 \leq j_1 \leq \bar{L}_1, 1 \leq j_2 \leq \bar{L}_2\}$, where $D_1^{j_1} = p_{j_1} D_1$ and $D_2^{j_2} = q_{j_2} D_2$, for $1 \leq j_1 \leq \bar{L}_1$ and $1 \leq j_2 \leq \bar{L}_2$. Notice that, although the batch sizes are i.i.d. random variables, the IATs of both types of batch arrivals can be correlated. On the other hand, the service times for both the high- and the low-priority customers are PH-distributed. For the high- (resp. low-) priority customers the parameters of the service-time distribution are (m_1, α, T) (resp. (m_2, β, S)).

To model this priority queue as an M/G/1-type MC the level variable must be chosen such that during a single transition it can decrease by at most one. Therefore, one may take the level either as the number of high- or low-priority customers, since in both cases in a single transition the level can only decrease by one (service completion), but it can increase by at most $\bar{L}_1$ or $\bar{L}_2$ (batch arrivals). In previous works [6, 86, 129] the number of high-priority customers has been used as the level. This choice induces a structure that can be exploited when both priority classes are assumed to have an infinite buffer. However, under BMAP arrivals, this approach requires the determination of $\bar{L}_1$ infinite block-matrices, which is done with a linearly-convergent algorithm [129] that may require extremely long computation times. Here we opt for the second alternative, that is to take the number of *low-priority* customers as the level. We further assume that the high-priority customers have a finite buffer of size C. This approach induces a completely different structure (restricted downward transitions), which can be exploited to obtain the stationary probability vector of the MC in an efficient manner for moderate values of

C, e.g., 50 or 100. The assumption of a finite high-priority queue (of moderate size) does not necessarily limit the applicability of the model since high-priority queues are typically fairly short as opposed to low-priority queues. Hence, there is little difference between having a, sufficiently large, finite or an infinite high-priority buffer.

Since the number of low-priority customers is chosen to be the level, the phase keeps track of the number of high-priority customers, and the state of the arrival and the service processes. Let $N_1(t)$ and $N_2(t)$ be the number of high- and low-priority customers in the system at time t, respectively. Also, let $J(t)$ be the state of the arrival process at time t, which takes values in the set $\{1, \ldots, m_a\}$. Let $S_1(t)$ (resp. $S_2(t)$) be the phase of the high- (resp. low-) priority service at time t when there is a customer of this class being served. When a low-priority customer is in service there is no need to keep track of $S_1(t)$. However, when a high-priority customer is being served, $S_2(t)$ keeps track of the phase at which the first low-priority customer in the queue will (re-)start its service. Then, when a low-priority customer is preempted by the arrival of a high-priority one, the service phase of the preempted customer is kept in $S_2(t)$ such that its service can be resumed from this phase. The queue is therefore described by the CTMC $X(t) = \{(N_2(t), N_1(t), J(t), S_2(t), S_1(t)), t \geq 0\}$. Its state space is ordered lexicographically and is described as follows. In level zero there is no need to keep track of $S_2(t)$ since there are no low-priority customers in the system. The states in this level are subdivided in two subsets: the first considers the case where there are no high-priority customers either ($N_1(t) = 0$), so it is enough to keep track of the phase of the arrival process. This case is covered by the set of states $\Delta_{00} = \{j, 1 \leq j \leq m_a\}$. The second subset is $\Delta_{0+} = \{(i, j, s_1), 1 \leq i \leq C, 1 \leq j \leq m_a, 1 \leq s_1 \leq m_1\}$, which considers the case where there are $i > 0$ high-priority customers, i.e., the server is busy with a customer of this type, and the service phase is s_1. Therefore, the set of states corresponding to level zero is $\Delta_0 = \Delta_{00} \cup \Delta_{0+}$. For every level $k \geq 1$ there are also two subsets. The first is $\Delta_{+0} = \{(j, s_2), 1 \leq j \leq m_a, 1 \leq s_2 \leq m_2\}$, which describes the case where there are no high-priority customers and the server is busy with a low-priority one. The second subset of level $k \geq 1$ is $\Delta_{++} = \{(i, j, s_2, s_1), 1 \leq i \leq C, 1 \leq j \leq m_a, 1 \leq s_2 \leq m_2, 1 \leq s_1 \leq m_1\}$, which considers the general case with $k \geq 1$ low-priority and $i \geq 1$ high-priority customers, the arrival process is in phase j, the service of the high-priority customer is in phase s_1 and the next low-priority customer to be served will start (or resume) in phase s_2. The set of states in level $k \geq 1$ is given by $\Delta_k = (k, \Delta_{+0} \cup \Delta_{++})$ and the full state space of the MC is $\Delta = \bigcup_{k \geq 0} \Delta_k$.

A complete description of the blocks that characterize the M/G/1-type MC of the BMAP[2]/PH[2]/1 priority queue can be found in Appendix A.2. Of particular importance, however, is the block A_0, which holds the transition rates from level k to level $k-1$. These transitions are triggered by the completion of a low-priority service, which can only occur in the absence of high-priority customers. Therefore, the block A_0 is

$$A_0 = \begin{bmatrix} I_{m_a} \otimes s\beta & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 \end{bmatrix}.$$

This matrix exhibits a very special structure, that is, only its first $m_a m_2$ columns are different from zero and this number is relatively small compared to the complete block size $m = m_a m_2 + C(m_a m_2 m_1)$. In other words, the block size is $(1 + Cm_1)$ times larger than the number of nonzero columns in A_0. Therefore, by modeling the priority queue in this manner we induce the restricted-downward-transitions structure. In addition, if the batch-sizes are i.i.d. random variables independent of the arrival process we have that $D_2^i = q_i D_2$. Then, by looking at the blocks $(A_i)_{i\geq 2}$ described in Equation (A.6), we find that $A_{i+1}^{--} = q_i (I_C \otimes D_2 \otimes I_{m_a m_2 m_1})$, for $1 \leq i \leq \bar{L}_2$. Moreover, from Equation (A.6) we also find that $(A_i^{+-})_{i=2}^{\bar{L}_2}$ and $(A_i^{-+})_{i=2}^{\bar{L}_2}$ are zero. Therefore, these blocks have the structure defined in Remark 2.1, which can be exploited together with the restricted-transitions structure as described in Section 2.2.3.

We now consider some numerical instances of the BMAP[2]/PH[2]/1 preemptive priority queue to illustrate the performance of the methods introduced in this chapter. As before, one of the main parameters of this queue is the load ρ, which in this case is given by $\rho = \lambda_1 E[L_1]/\mu_1 + \lambda_2 E[L_2]/\mu_2$. For customers of type i, λ_i is their mean arrival rate, μ_i is their mean service rate and $E[L_i]$ is their mean batch size, for $1 \leq i \leq 2$. Recall that high- (resp. low-) priority customers are referred to as customers of type one (resp. two). We assume that high- and low-priority customers have the same mean service rate equal to one, i.e., $\mu_1 = \mu_2 = 1$. Also, both service-time distributions have SCV equal to two. We can match these two moments with a PH distribution of order 2 by using the method in [120]. The batch-size distribution is assumed to be the same for both customer types. Specifically, we set $\bar{L}_1 = \bar{L}_2 = \bar{L}$, and let the batch size of both types be uniformly distributed between one and $\bar{L}$, with mean $E[L]$. Here we first build a single arrival process with arrival rate $\lambda = \lambda_1 + \lambda_2$. With the assumptions stated above this arrival rate is given by $\lambda = \rho/E[L]$. We assume that the SCV of the batch IAT distribution is equal to five and the decay rate of the autocorrelation function of the sequence of batch IATs is 0.5. We use the method in [40] to match the first two moments of the IAT distribution and the decay rate of the autocorrelation function with an order-2 MAP. The resulting process is a MAP characterized by the 2×2 matrices D_0 and D_+, that describe the batch IATs irrespective of their type. The matrices D_1 and D_2 are obtained as $D_j = v_j D_+$, for $1 \leq j \leq 2$, where $v_1 + v_2 = 1$. In this scenario we also assume that both customer types have the same arrival rate ($v_1 = v_2 = 0.5$). As the size of the MAP is $m_a = 2$ and the size of the PH representation of the service-time distributions is $m_1 = m_2 = 2$, the number of nonzero columns is $r = 4$ and the block size of the original M/G/1-type MC is $m = 4 + 8C$.

For the results shown in Table 2.4 we set the maximum batch size equal to five and the size of the high priority buffer is equal to 20. Even this buffer size causes very few losses as illustrated by the loss rate (LR) of high-priority customers included in the last column of the table. Here we observe how the use of the censored process to compute π reduces the computation times dramatically compared with simply solving the original M/G/1-type MC. Moreover, we see that the main gain is obtained thanks to the fast computation of G, and additional gains can be realized by using the censored process to compute π, although only when the additional structure from Remark 2.1 is exploited. Specifically, we observe that **C-RF** requires more time than **O-RF** to compute π, **C-**

ρ	B-RF	O-RF	O-FRF	C-RF	C-FRF	C*-RF	C*-FRF	C*-FRF$_2$	LR
0.1	51.0	0.3	1.2	0.5	0.4	0.4	0.4	0.4	2.09E-09
0.3	112.5	0.4	2.1	0.5	0.6	0.4	0.4	0.4	2.25E-06
0.5	164.8	0.8	4.2	1.0	0.9	0.7	0.6	0.7	5.92E-05
0.7	222.0	1.8	10.3	2.9	1.8	2.5	1.4	1.4	4.67E-04
0.9	282.8	13.2	38.7	20.1	13.5	17.8	9.7	5.0	1.93E-03

Table 2.4: Computation times (sec) for $\bar{L} = 5$, $C = 20$

FRF shows similar times than **O-RF**, and **C*-RF** performs worse than **O-RF** for high loads. In fact, we have noted that in many cases, as in the previous section, the gain obtained by using **C*-RF** instead of **O-RF** decreases as the load increases. The reason for this behavior is that to compute π^+ in **C*-RF** we use Equation (2.13) with the blocks of the censored process, but the number of blocks increases with the load. For instance, in this scenario for loads 0.1, 0.5 and 0.9, the number of blocks $(\bar{A}_i)_{i\geq 0}$ is 45, 199 and 552, respectively. This effect is reduced when the ratio m/r becomes large (e.g., 50), since in this case the vector-matrix multiplications of the original blocks become very expensive. Therefore, the censored process has the drawback of having many blocks, requiring many vector-matrix multiplications to compute the sum in (2.13). However, when the number of blocks and the number of terms in the vector π are large, the FRF is expected to perform significantly better than the customary RF [84]. This is the case for the censored process when the load is high, and we therefore observe important gains when using **C-FRF** instead **C-RF**, and **C*-FRF** or **C*-FRF$_2$** instead of **C*-RF**.

ρ	INF	O-RF	O-FRF	C-RF	C-FRF	C*-RF	C*-FRF	C*-FRF2	LR
0.1	2	1.9	11.7	2.6	3.3	1.9	1.9	2.5	2.14E-18
0.3	343	2.7	17.2	4.3	5.6	2.7	2.8	3.8	8.10E-12
0.5	1010	5.0	32.2	7.7	10.4	4.3	4.5	6.1	1.25E-08
0.7	2427	12.7	79.2	17.9	21.7	9.5	9.7	11.3	1.41E-06
0.9	6832	71.1	308.9	95.3	81.4	59.9	45.2	31.7	3.70E-05

Table 2.5: Computation times (sec) for $\bar{L} = 5$, $C = 50$

We now consider a simple modification on the previous scenario by increasing the buffer size from 20 to 50. This makes the block size equal to 404. For this size the method **B** fails to compute the vector π because of lack of memory. At this point we must mention that all the times shown here were obtained using a personal computer with a 2Ghz processor and 2GB of RAM. Table 2.5 shows the computation times for the methods introduced in this chapter and for the approach introduced in [129]. The difference in computation times is extremely large for moderate to large loads. It must be noted that our methods do not scale easily with the size of the high-priority buffer C. If this parameter is large, our methods would require a large amount of memory, while

the **INF** method does not suffer from this problem. However, if the buffer is not so large our methods provide huge savings in computation time. For instance, Table 2.5 shows that, when the load is 0.9, the **INF** approach takes more than two hours to compute π, while **C-RF** requires only one and a half minute, and the best of our methods in that case (**C*-FRF**$_2$) takes about half a minute. Additionally, the loss rate assuming a finite buffer is almost negligible even for high loads. We also observe that the **C*-FRF** and **C*-FRF**$_2$ have a better performance for high and very high loads, while the **C*-RF** method is slightly better for low loads. Among the methods that do not use the extra structure introduced in Remark 2.1, the best choice is the **O-RF**, as the methods based on the censored process require more time, partly due to the computation of the blocks of the censored process.

In the next scenario we make the maximum batch size $\bar{L}$ equal to 15, such that both the original and the censored M/G/1-type MCs have more blocks. The computation times are shown in Table 2.6. Comparing with the results in Table 2.5, we see that all the methods require more time to compute π, but the increase is not proportional. Actually, the **INF** method takes more than ten times longer in the new scenario. Among our methods, **C*-FRF**$_2$ shows the best behavior for high loads, roughly doubling the computation time compared to the previous scenario. We also observe that for mid loads the difference among the **C*** methods is negligible, but for high loads the advantage of using **C*-FRF**$_2$ is large, even compared to **C*-FRF**. Here again **O-RF** is the best method among those that do not exploit the additional structure from Remark 2.1. Also, we notice that, in this and the previous scenarios, the use of the FRF provides no gain for the **O** method, as is to be expected given the large block size. For the other methods, the use of the FRF provides a significant gain for mid-to-high loads, as in this case the FRF is applied on the censored process, the size of which is by construction very small.

ρ	INF	O-RF	O-FRF	C-RF	C-FRF	C*-RF	C*-FRF	C*-FRF$_2$	LR
0.1	55	5.0	28.0	9.3	13.4	4.7	4.8	5.7	2.77E-10
0.3	4149	9.8	39.6	16.0	24.6	6.6	7.0	8.2	2.33E-07
0.5	10661	24.0	76.7	30.3	29.5	10.9	10.0	11.2	6.58E-06
0.7	31033	76.1	195.8	93.6	87.0	32.6	25.8	22.8	5.90E-05
0.9	82545	398.3	699.4	475.7	426.4	240.3	187.2	64.9	2.78E-04

Table 2.6: Computation times (sec) for $\bar{L} = 15$, $C = 50$

We conclude this section by looking at a scenario where the number of high-priority customers is a small fraction of the total number of customers. This scenario may arise, for instance, in a communication system where the high priority is assigned to a subset of the customers such that this subset will experience a better quality of service. One way to accomplish this is to assign high priority only to a small proportion of the customers. To consider this we simply set $v_1 = 0.1$, meaning that the high-priority customers represent only 10 percent of the customers. In Table 2.7 we see that under this configuration the **INF** approach has a better performance than in the previous scenario, where the

proportions of high- and low-priority customers were equal. This gain is due to the fact that, in this case, the most expensive operation in all the methods is the actual computation of π, after obtaining π_0 and the set of R matrices (in the **INF** approach) or the matrix G (in our approach). Since now there are relatively fewer high-priority customers in the system and the **INF** method uses the number of these customers as the level, this method needs to compute fewer terms of the vector π. On the other hand, our methods use the number of low-priority customers as the level and therefore they now need to compute many more terms of π. In spite of this gain, under high loads the **O-RF** and **C*-FRF**$_2$ methods are still around 10 and 100 times faster than the **INF** approach, respectively. For lower loads, the gain obtained by using **C*-RF** is even larger, except for very low loads. We also observe that, for loads up to 0.7, the **O-RF** method has the best performance among those that do not use the structure in Remark 2.1. However, for $\rho = 0.9$ the **C-FRF** method is able to outperform the **O-RF** approach. As in this case the proportion of high priority customers is smaller than in the previous scenario, the loss rate is also smaller and becomes negligible.

ρ	INF	O-RF	O-FRF	C-RF	C-FRF	C*-RF	C*-FRF	C*-FRF$_2$	LR
0.1	44	5.5	28.2	11.5	14.8	6.1	6.1	7.0	3.87E-14
0.3	1854	10.3	41.1	18.0	26.5	8.5	8.9	10.2	1.44E-11
0.5	2730	24.6	78.1	32.6	31.8	12.2	11.5	12.8	2.58E-10
0.7	4247	79.3	202.8	92.9	89.1	28.7	24.6	21.0	1.82E-09
0.9	6312	700.8	1035.2	740.0	628.8	390.8	306.0	71.6	8.06E-09

Table 2.7: Computation times (sec) for $\bar{L} = 15$, $C = 50$, $v_1 = 0.1$

2.4 Conclusion

From the results in the previous section we can conclude that the methods proposed in this chapter provide an important tool to evaluate the performance of the BMAP/PH/1 queue and the BMAP[2]/PH[2]/1 preemptive priority queue since they are able to analyze rather large systems in a fraction of the time required by other methods. In general we see that exploiting the restricted-downward-transitions property, we are able to reduce the total time to compute π. As discussed at the end of the previous chapter, the gains depend not only on the ratio m/r, but also on the parameters of the system, and how these affect the expected sojourn times in $\mathcal{S}^-$, compared to those in $\mathcal{S}^+$. However, we have seen in the previous examples that our methods are able to provide significant gains in computation time even for a rather small ratio m/r. Also, among the methods that only exploit the restricted-transitions structure, we have observed that typically the **O-RF** method provides the best results, while the **O-FRF** is badly affected by the large block size. The use of the censored process, either **C-RF** or **C-FRF**, can in some cases provide some additional gains. However, given the overhead required by the **C** methods, compared to the simpler **O-RF**, the latter appears as a better option when the additional

structure of Remark 2.1 is not present. Notwithstanding, when this additional structure is present, the use of the censored process leads to very important gains. In this case the **C*-RF** method has the best performance for low loads, while for high loads the **C*FRF**$_2$ approach is the best alternative to compute π.

Part II

Optical Grids

Optical Grids

An optical grid is a set of computing sites/stations interconnected by optical links. Each of these sites is itself connected to a number of users who request the processing of tasks/jobs. Originally, these grids were deployed to process and store the large amounts of information arising from research efforts in fields such as astrophysics, particle physics, chemistry, biomedicine, among others [121]. More recently, optical grids have also been recognized as an alternative to provide consumer-oriented services [78]. In either case, the users submit jobs for processing but they have no preference about where the jobs should be processed [35, 36]. The site that receives a job tries to process it locally but, if all its servers are busy, the job can be sent to any of the other sites. This is in clear contrast with traditional networks, where it is assumed that (a fraction of) the jobs originating in a particular source must be processed in a predetermined site. In an optical grid the site where a specific job ends up being processed depends on the state of the site where it originates, the state of the other sites, and a scheduling policy that determines which site will be used to process the job. As a result, the amount of traffic that must be transported between each pair of stations depends not only on the arrival process at each station and a predefined routing matrix, but also on the number of servers per site and the policy used to allocate those jobs that cannot be processed locally.

Dimensioning an optical grid is therefore a complex problem, typically involving tens to hundreds of sites, each equipped with tens to hundreds of servers. In addition, the job mean inter-arrival times (IATs) and their arrival variability may differ significantly among the different sites. There are many relevant questions related to grid dimensioning [35] such as how to select the places to locate the server capacity, how many servers must be placed per site, what scheduling algorithm to choose to redistribute the jobs that must be processed remotely, and how to determine the bandwidth of the inter-site links, among others. Solving all these problems altogether is hard (even single-period dimensioning for a given traffic matrix may already be NP-hard [87]) and therefore one must approach them separately. For example, in [113] divisible load theory is used to determine the number of wavelength channels per fiber connecting the sites, which are assumed to have been dimensioned to handle the locally generated jobs. Of special interest to us is the methodology put forward in [39], where the authors propose to split the problem in phases, using the result of each phase as input for the next one. Specifically, the approach in [39] starts with a graph representing the topology of the grid network, the job arrival process and the processing rate at each site as well as the target maximum job loss rate. The first step of the methodology is to select the best sites to place the servers, i.e., all the

sites generate jobs, but only some of them have local processing servers. The result of this step is a set of sites where the servers will be located and whose arrival process includes jobs from other sites that lack processing power. The second step consists of assigning the servers to the sites already selected. The third step takes the set of sites and their server capacities and determines the amount of traffic that flows between each pair of sites, referred to as the traffic matrix. The traffic depends not only on the server capacities at each site, but also on the scheduling policy used to determine the site where each job ends up being processed. The last step is to dimension the links that connect the stations so that they can carry the traffic estimated in the previous step.

In [39] it is shown that an integer linear program can be used to solve the first step for small- to medium-sized instances. For larger instances the K-means algorithm [64] can be used to rapidly obtain very good solutions. For the second and third steps the authors of [39] also propose a few rules to determine the number of servers per site and how to schedule the jobs that cannot be processed locally. To determine the traffic matrix in the third step, which must be used as input for the last phase, they rely on time-consuming simulations. For the last step, and since the traffic matrix has already been determined, it is possible to use dimensioning techniques that may even depend on the specific switching technology [39]. As can be seen, one of the critical steps in this methodology is to estimate the traffic matrix by means of simulation. In fact, modeling this system is not an easy task as the traffic that flows from one site to another depends on the state of all the sites. For instance, one of the possible strategies to schedule the jobs that must be processed remotely is to choose a server in the station that has the largest number of idle servers. To this end it is in principle necessary to keep track of the state of every station, which makes the problem highly intractable. In fact, in [39] a fixed point approximation based on the Erlang-B formula is proposed as an alternative to simulations, but it is shown to be far from accurate.

In the next two chapters we will introduce two different approaches to compute the traffic matrix. In Chapter 3 we focus on a grid network with a ring topology where each station is physically connected to only one other site through an unidirectional fiber link. If an incoming job finds all the local servers busy, the job is sent to the next station in the ring where it is served by an available server. If no servers are available in that station, the job is sent again to the next site. In this manner the job goes from one station to the next trying to find an idle server. After trying all the stations once, the job is dropped from the network. The analysis of this system, which has a very specific topology and a simple scheduling algorithm, is still a very hard problem as it is necessary to keep track of the state of all the stations to determine the site where an arriving job ends up being processed. Therefore we propose two different methods to approximate the performance of an optical grid with these characteristics. Both approaches rely on a marked Markovian representation of the overflow process at each station and on reducing this representation by moment-matching methods. The first method is based on approximating the inter-overflow time process, while the second separately characterizes the periods where jobs are overflowed and the periods where they are served locally. The results show that the methods accurately approximate the rate of locally processed jobs, one of the main performance measures.

Subsequently, in Chapter 4 we propose a mean field model (see Appendix A.3) to analytically derive the traffic matrix in a grid with a large number of sites. We assume that the sites, characterized by their server capacity and job arrival process, can be partitioned in a limited number of classes. The benefit of the mean field model is that it allows us to compute the traffic matrix when the number of sites is large. It also allows the scheduling algorithm to depend on the proportion of stations in each possible state. On the other hand, it is not possible to make the scheduling algorithm to depend on the state of a particular site, and therefore information related to the topology of the network is not considered. Additionally, the mean field model is exact when the number of sites is infinite, but we show that it approximates very well the behavior of a grid with a large but finite number of sites. Moreover, the traffic matrix can be computed by means of the mean field model with a fraction of the computational effort required by general simulation techniques. This is also the case for the method presented in Chapter 3. In summary, the methods to be introduced provide a way to efficiently and accurately estimate the traffic matrix in an optical grid with either a ring architecture or a large number of sites.

Chapter 3

A Grid network with a ring topology

In this chapter we consider an optical grid with a ring topology. If a job arrives it is always served by a local server if there is one available. If all the local servers are busy the job is sent to the next station in the ring, where it is served by an idle server. Here again, if no servers are available the job is sent to next station. The job goes from one station to the next until it finds an idle server or, if it visits all the stations once, the job must be dropped. Typically a grid network consists of many stations and, thus, the scalability of the methods to analyze these networks is of major importance. In this work we decompose the network in single nodes to perform the analysis of each station separately. This technique has been widely used in the analysis of queueing networks that lack a product-form solution. Particularly, moment-matching methods have played a key role in approximating the arrival, departure and overflow processes of the single nodes in the network, see [68,69,120] and references therein. This approach has also been used recently to approximate the departure process of queues with Markovian input [55, 57, 58, 60]. Since the traffic between stations in the ring forms an overflow process, the analysis of teletraffic networks with alternate routing is particularly relevant [67,68,82,83,122]. In those networks the incoming traffic is first offered to a trunk group with a finite buffer. If the buffer is full, the traffic is sent to an alternative trunk group, forming an overflow process. To compute the performance measures for each of the flows in the network separately, the methods in [68,83] rely on representing each of these flows as an independent process. This approach is well-suited for general networks with different routing patterns, but dimensionality problems arise when analyzing even small networks. In [82] the authors propose to group all but one overflow process into a single description in order to model the arrival process at each station with one Poisson and two overflow processes. With this method the dimensionality problems are avoided, but an additional approximation is introduced.

In this work we exploit the ring topology to approximate the overflow process at each station by means of a marked Markovian point process. The markings are used to differentiate each of the flows in the network, leading to a more compact representation of

the overflow process. This representation has to be further reduced to make the analysis of the network feasible. We propose two different methods to build an approximate (reduced) representation of the overflow process. The first approach aims at approximating the stationary inter-overflow time process by matching a set of (joint) moments that uniquely characterize the reduced point process. The second approach separately characterizes the periods where jobs are overflowed and the periods where they are served locally, combining their reduced representation into a single one. Additionally, the first approach is introduced stepwise. We start by approximating the stationary inter-overflow time distribution using renewal approximations matching three moments with Phase-Type (PH) distributions. Then we analyze the effect of capturing more moments by relying on the class of Matrix-Exponential (ME) distributions. Next we include information of the joint moments of successive inter-overflow times by using recent matching methods based on the marked and unmarked versions of the Rational Arrival Process (RAP). These approximations are tested for different network parameters with particular emphasis on computing the rate of locally processed jobs (local rate) and the total traffic matrix. This is of vital importance when dimensioning the link capacities interconnecting the different stations [35, 38]. The results show that the methods are able to adequately approximate these measures, being especially accurate for the local rate.

This chapter is organized as follows. Section 3.1 includes a description of the grid network as well as a discussion of some characteristics common to both approximation methods. The approximation method based on the inter-overflow time process is presented in Section 3.2, while the second method, called ON-OFF, is introduced in Section 3.3. Section 3.4 compares the performance of both methods for various realistic network configurations.

3.1 The grid network

We consider a grid network that consists of N nodes arranged in a ring topology. The job arrivals at node i are represented by a Poisson process with rate λ_i (this assumption will be relaxed to allow for more general arrival processes as described at the end of this section). The Poisson assumption is based on Grid level measurements [31] and has been employed in previous works on Grid dimensioning [35, 38]. When a job arrives at node i, it is served by any of the C_i servers in that node, if at least one is available. In case all of them are busy, the job is sent to the next node in the ring. As mentioned before, the job jumps from node to node until it finds an idle server. If a job originating in node i arrives at station $i-1$ and there are no available servers, it must be dropped. Thus, a job that has tried all the stations once is dropped in the station before the one where it was generated. The service time of a job in station i is assumed to be exponentially distributed with rate μ_i. This means that the service time depends on the station serving the job, but not on the station where it entered the grid. From now on we will refer to a job that originally arrives at station i as a job of type i, for $i = 1, \ldots, N$.

Even in the case where the jobs of each type arrive according to a Poisson process, the actual flow that arrives to a particular station is a complex mixture of all the job types

present in the network. Hence we will approximate the input process[1] at station i as a marked Markovian Arrival Process [52] (MMAP[N]) characterized by a set of $m_i \times m_i$ matrices $\{D_0^i, D_1^i, \ldots, D_N^i\}$ (see Appendix A.1). Here the matrix D_j^i describes the transition rates related to an arrival of type (originated in station) j, for $j = 1, \ldots, N$. These matrices will be built through an iterative process in order to incorporate information about the overflowing jobs along the network into the analysis of each station. Let $\{N_i(t), t \geq 0\}$ (resp. $\{J_i(t), t \geq 0\}$) be the number of busy servers (resp. the phase of the arrival process) at station i at time t. Then $\{(N_i(t), J_i(t)), t \geq 0\}$ is a CTMC on the state space $\{(k, l), 0 \leq k \leq C_i, 1 \leq l \leq m_i\}$, that describes the state of the station i, based on the approximated arrival process. Its generator matrix is given by

$$Q^i = \begin{bmatrix} D_0^i & D_+^i & 0 & \ldots & 0 & 0 \\ \mu_i I & D_0^i - \mu_i I & D_+^i & \ldots & 0 & 0 \\ 0 & 2\mu_i I & D_0^i - 2\mu_i I & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & D_0^i - (C_i - 1)\mu_i I & D_+^i \\ 0 & 0 & 0 & \ldots & C_i \mu_i I & D^i - C_i \mu_i I \end{bmatrix}, \tag{3.1}$$

where $D_+^i = \sum_{j=1}^N D_j^i$ and $D^i = D_0^i + D_+^i$, and I is the identity matrix.

If the matrices $\{D_0^i, D_1^i, \ldots, D_N^i\}$ are specified, the steady state probability vector can be easily computed by exploiting the structure of matrix Q^i as a finite Quasi-Birth-and-Death (QBD) process [46, 76, 91]. The time and memory complexity of the algorithm in [46] to compute the steady-state distribution are $O(C_i m_i^3)$ and $O(C_i m_i^2)$, respectively. Nonetheless, the arrival process to a particular station is determined by the superposition of the new arrivals arriving at that station and the overflow process coming from the previous station in the network. Even though these could be exactly represented by including in the arrival process the state of all the stations in the network, the size of such a representation is huge even for a small number of stations and servers per station. Previous work by Meier-Hellstern [83] considers the problem of approximating the arrival process at each station by representing each of the flows in the network as a Markov Modulated Poisson Process (MMPP) and combining them for each station. An MMPP can be seen as an MMAP[1] where only the matrix D_1 is specified and it has zero off-diagonal elements. In [83] the algorithm of Heffes [54] is used to reduce each of the MMPPs to be of size 2. Even though this approximation is well suited to represent different routing strategies, it does not scale well with the number of stations because the input process to a particular station is of size 2^{N-1}. Our approximation methods make use of the ring structure to avoid this large size while keeping meaningful information about the overflow process at each station.

The overflow process at station i can be represented as an MMAP[N] characterized

[1] In Section 3.2.1 we will relax this approximation to a marked Rational Arrival Process (MRAP[N])

by the matrices $\{E_0^i, E_1^i, \ldots, E_N^i\}$, given by

$$E_0^i = \begin{bmatrix} D_0^i & D_+^i & 0 & \ldots & 0 & 0 \\ \mu_i I & D_0^i - \mu_i I & D_+^i & \ldots & 0 & 0 \\ 0 & 2\mu_i I & D_0^i - 2\mu_i I & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & D_0^i - (C_i - 1)\mu_i I & D_+^i \\ 0 & 0 & 0 & \ldots & C_i \mu_i I & D_0^i + D_s^i - C_i \mu_i I \end{bmatrix}, \quad (3.2)$$

$$E_j^i = \begin{bmatrix} 0 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & D_j^i \end{bmatrix}, j = 1, \ldots, N, j \neq s,$$

and $E_s^i = 0$, where s denotes the next station after node i, i.e., $s = (i \mod N) + 1$, and 0 is a zero matrix of appropriate dimension. This representation reflects the fact that the jobs originating from station s must be dropped if they cannot be served in node i. Even though this representation is exact, it is not useful for practical implementation, because the size of the matrices that describe the overflow process becomes extremely large after a few stations. Thus, it is useful to reduce the size of these matrices such that the reduced representation keeps some characteristics of the original and can be used as part of the arrival process at the next station. We propose two different methods to find an approximate representation of the overflow process, which result in an MMAP[N] with matrices $\{\bar{D}_0^i, \bar{D}_1^i, \ldots, \bar{D}_N^i\}$. The first one is based on the approximation of the inter-overflow time process. The second method divides the representation of the overflow process into two sets: one characterizes the time periods where any arriving packet is sent to the next station (ON period), while the other captures the behavior of the station when there are servers available to process an incoming job (OFF period). Although each method relies on a different type of information, some main features are alike. For example, both methods represent the overflow process at station i by means of a reduced MMAP[N] (or a generalization of it) with parameters $\{\bar{D}_0^i, \bar{D}_1^i, \ldots, \bar{D}_N^i\}$. Other common features are presented in the next section. Notice, while these matrices describe the *overflow process* at station i, the matrices $\{D_0^i, D_1^i, \ldots, D_N^i\}$ describe the approximated *arrival process* at station i. Their relationship is described in the next section.

3.1.1 Common characteristics of the approximation methods

Iterative approach: The ring topology implies that all the nodes receive both newly generated and overflow jobs, meaning that the arrival process at each station depends on the analysis of the previous one. To contemplate this, both methods consider an iterative strategy in which an arbitrary station is first analyzed considering only its own traffic. The next station in the ring is then considered, including both newly generated arrivals and overflow packets only from the previous station. The analysis continues at each station and, after considering all the stations once, the overflow process contains packets of (possibly) all types. Then, it is possible to reanalyze the "first" station, but now the

arrivals include both newly generated jobs as well as jobs coming from (possibly) all the other stations in the ring. This sequential analysis is performed several times for each station until the traffic matrix (that contains the amount of traffic between each pair of stations) changes less than a predetermined value ϵ, e.g., $\epsilon = 10^{-8}$.

Moment matching: Another relevant issue for both methods is the reduction of the representation of an inter-event process by means of the moment-matching methods introduced in [21,110,111,115,120]. The reduction of the process always implies the reduction of the inter-event time distribution by matching some of its moments with a distribution with smaller representation (if the moments are attainable by the matching distribution). Some of these methods provide closed-form formulas for the parameters of the matching distribution [110,120], making them well suited for iterative procedures as those presented in the next sections. The other methods require more computational effort, but this is still negligible compared to the computation of the moments themselves. This is due to the large number of phases in the exact representation of the inter-event time distribution. The direct formulas to compute the moments [76] have a time and memory complexity of $O(C_i^3 m_i^3)$ and $O(C_i^2 m_i^2)$, respectively. However, the representation of the inter-event time distribution has a block-tridiagonal structure as the one shown in Equation (3.1). To exploit this structure we make use of the algorithm introduced in [46] to compute the first two moments of the first-passage time distribution to higher levels in a finite QBD, where level k corresponds to the set of states with k busy servers: $\{(k,l), 1 \leq l \leq m_i\}$. Using the generating function of the first-passage times described in [46] it is possible to determine higher moments of this distribution to use them as input for the moment-matching methods. Appendix A.6 discusses why the inter-event times used by the approximation methods in Sections 3.2 and 3.3 can be interpreted as first-passage times to higher levels in a finite QBD. It also describes an algorithm based on [46] to compute any number of moments of this distribution. This is particularly useful for the method in Section 3.3, where the size of the arrival process at each station grows linearly with the number of stations. The time and memory complexity of the algorithm to compute these moments are thereby reduced to $O(C_i m_i^3)$ and $O(C_i m_i^2)$, respectively.

Arrival process: To characterize the arrival process at each station, the overflow coming from the previous station and the new arrivals at this station must be combined. Recall that the overflow process at station i is described as an MMAP[N] with parameters $\{\bar{D}_0^i, \bar{D}_1^i, \ldots, \bar{D}_N^i\}$. Under the assumption of Poisson arrivals, the new incoming jobs at station s and the overflow from station i can be combined as an MMAP[N] with parameters

$$D_0^s = \bar{D}_0^i - \lambda_s I, \qquad D_j^s = \bar{D}_j^i, \quad j = 1, \ldots, N, j \neq s, \qquad D_s^s = \lambda_s I. \tag{3.3}$$

It will become clear that the approximation introduced in Section 3.2 can deal not only with Poisson arrivals, but with more general point processes as well. Let the arrivals at station s be described by a MAP characterized by $\{B_0^s, B_1^s\}$ (see Appendix A.1). Assuming this arrival process, the combined stream at station s can be represented as an MMAP[N]

with parameters

$$D_0^s = \bar{D}_0^i \oplus B_0^s, \qquad D_j^s = \bar{D}_j^i \otimes I, \quad j = 1, \ldots, N, j \neq s, \qquad D_s^s = I \otimes B_1^s,$$

where $\otimes$ and $\oplus$ stand for Kronecker product and sum [16], respectively. Even though the second approximation (Section 3.3) can in principle include this more general process, the size of the representation of the arrival process increases exponentially with the dimension of the MAP at each station. For the technique in Section 3.2, we do not encounter such an exponential increase.

3.2 An approximation based on inter-overflow times

The approximation introduced in this section is based on reducing the size of the representation of the inter-overflow time process. As explained above, the size of the exact representation of this process is extremely large, making the analysis of even small networks infeasible. Here we introduce different methods to determine a reduced approximate representation of the overflow process. We first consider the case where the overflow process is approximated by a renewal process, assuming independent inter-overflow times. Next we allow the new process to be non-renewal by including information about the joint moments of successive inter-overflow times.

3.2.1 Renewal Approximation

3-moment match

When matching the inter-overflow time distribution, we make use of Phase-Type (PH) distributions (see Appendix A.1). From Equation (3.2) it is clear that E_0^i is the subgenerator of the PH representation of the inter-overflow times at station i. To represent the overflow process with a PH renewal process, we consider the stationary version of the inter-overflow times, with representation given by (γ_i, E_0^i). To define γ_i we first need the steady state probability vector of Q^i, i.e., the vector π^i such that $\pi^i Q^i = 0$ and $\pi^i e = 1$. This vector can be partitioned as $\pi^i = [\pi_0^i, \pi_1^i, \ldots, \pi_{C_i}^i]$, where π_j^i corresponds to the states with j busy servers in station i, for $j = 0, \ldots, C_i$. Hence the steady state distribution of the phase of the arrival process after an overflow is given by

$$\beta^i = \frac{\pi_{C_i}^i D_+^i}{\pi_{C_i}^i D_+^i e}. \tag{3.4}$$

By partitioning γ_i in the same way as π^i and using the fact that overflows can only occur when the system is full, we obtain that

$$\gamma^i = [0, \ldots, 0, \beta^i].$$

Given the large number of phases in (γ_i, E_0^i), we consider moment-matching algorithms to obtain a PH representation with fewer phases. In particular, we make use of the algorithms described in [21, 110] to match the first three moments with an acyclic PH

distribution with minimal number of phases. In particular, when the squared coefficient of variation (SCV) of the inter-overflow time distribution is greater than or equal to $\frac{1}{2}$, the resulting PH distribution has only two phases. If the SCV is greater than one, the method given in [120] can also be used to get a 3-parameter hyper-exponential representation, which is a particular case of the PH class with two phases. For the specific case of the inter-overflow time distribution, we found that the SCV was always above one in our experiments. This is closely related to the high variability inherent in an overflow process where the jobs are only overflowed in a small set of the state space. Additionally, we observed that the two-phase representation was able to capture the third moment of the inter-overflow distribution in the numerical instances considered in Section 3.4 as well as in many other cases not presented here. However, this is not the case in general, since the set of moments to match $\{n_i, 1 \leq i \leq 3\}$ have to fulfill the following condition: if $n_1 > 0$ and the SCV is greater than 1, the third moment can be represented by an acyclic PH distribution [110] or a hyper-exponential distribution [120], both with two phases, if and only if $3n_2^2 \leq 2n_1n_3$. In case the set of moments cannot be represented by a PH distribution of order two, the method in [21] determines the minimum number of phases required to do it with an acyclic PH distribution.

Let (α_i, A_i) be the parameters of the reduced PH distribution obtained from the moment-matching method applied to the PH(γ^i, E_0^i) representation of the stationary inter-overflow time distribution at station i. In this reduced representation the off-diagonal elements of the matrix A_i describe transition rates without arrivals, while the transition rates related to arrivals are included in the matrix $-A_ie\alpha_i$. Therefore, the matrices $\{\bar{D}_0^i, \bar{D}_1^i, \ldots, \bar{D}_N^i\}$ can be approximated as

$$\bar{D}_0^i = A_i, \qquad \bar{D}_j^i = -A_ie\alpha_i \frac{\pi_{C_i}^i D_j^i e}{\pi_{C_i}^i \left(D_+^i - D_s^i\right) e}, \quad j = 1, \ldots, N, j \neq s, \tag{3.5}$$

and $\bar{D}_s^i = 0$. For $j \notin \{0, s\}$ the approximate $\bar{D}_j^i$ is the product of the matrix of transition rates involving an overflow of any type and the probability that a particular overflow is of type j. When the overflowing packet is coming from the station s (right after station i), it is dropped and none of those packets are actually part of the overflow process. This reduction will be labeled PH(3) in the results, making explicit that it relies on a PH representation that matches the first three moments of the inter-overflow time distribution.

Matching higher moments

In order to analyze the effect of capturing more moments of the inter-overflow time distribution we rely on the class of Matrix-Exponential (ME) distributions. An ME random variable has a density function given by $f(x) = \alpha e^{Tx}t$, for $x \geq 0$. Here α is a $1 \times m$ vector, T is a square matrix of size m, and t is an $m \times 1$ vector (m is called the order of the representation) [20]. An ME representation can always be chosen such that $t = -Te$ and $\alpha e = 1$ [7, 15]. Therefore, an ME distribution, as in the PH case, is characterized by the tuple (α, T), but the parameters are less restricted than in the PH class. More specifically, the matrix T must be invertible and the density must be non-negative and

integrate to 1. Even though the entries of α and T can be complex numbers, the ME class is equally broad if they are restricted to be real [10]. In this sense the ME class can be seen as a generalization of the PH class, since the tuple (α, T) must satisfy some extra conditions to be a representation of a PH distribution: α must be non-negative and sub-stochastic, while T must be the sub-generator of a CTMC, as explained in Appendix A.1 . A discussion and further properties of ME distributions can be found in [7,8,10,43,53,80].

As shown in [15], the traditional analysis of QBD processes can be extended to admit more general components, e.g., allowing ME instead of PH distributions. Therefore, we can extend the analysis of each station to allow for arrivals coming from a marked Rational Arrival Process (MRAP[N]) with parameters $\{D_0^i, D_1^i, \ldots, D_N^i\}$, as a generalization of the MMAP[N] [59]. These matrices have real-valued entries, their sum $D^i = \sum_{j=0}^{N} D_j^i$ must satisfy $D^i e = 0$, and the real part of the dominant eigenvalue of D_0^i (resp. D^i) must be negative (resp. equal to zero). In this case the stationary inter-overflow time distribution has an ME representation given by (γ^i, E_0^i) as defined above. This result relies on the fact that the time between successive overflows can be represented as the time until absorption in a finite-state semi-Markov process with ME holding times, which is itself ME distributed [10]. In this case E_0^i is a real matrix with negative dominant eigenvalue and the vector γ^i is no longer a probability mass function, but a vector of weights of measures after an overflow [8, 15].

Relying on the ME class we can use the method proposed in [111, 115] to reduce the order of the representation (γ^i, E_0^i) by matching $2n-1$ moments with an ME distribution of order n. This method is based on the algorithm for the partial realization problem proposed in [49]. Using the resulting representation and Equation (3.5), we get a smaller set of parameters to approximately represent the overflow process, matching not only three, but $2n-1$ moments of the stationary inter-overflow time distribution. The results based on an ME representation of order n will be labeled ME$(2n-1)$. Nevertheless, the algorithm in [115] does not assure that the kernel matrix obtained from a set of moments defines a distribution function, i.e., the function has the required moments, but it may be negative. To the best of our knowledge, a characterization of the moments representable by an ME distribution of arbitrary order is not available. To make use of these distributions we evaluate the density function at several points to numerically verify if it is non-negative. This test is done for every reduced representation (α_i, A_i) obtained from the matching algorithms.

From the description above it is clear that the size of the reduced ME representation can be defined *a priori* according to the number of moments to match. Another approach is to rely on the characterization of the minimal order of an ME distribution given in [53]. There the authors propose the use of a set of Hankel matrices to determine the minimal order of an ME distribution. One of the Hankel matrices is built from the moments of the distribution and its rank determines the minimal order of an ME distribution with the specified moments. If the minimal order is found to be n, then $2n-1$ moments can be used as input for the method in [111, 115] to obtain an ME representation of the stationary inter-overflow time distribution. With this method, the size of the ME representation could be made variable, however, the minimum order could be large, making the analysis of the next station a lengthy process. Nevertheless, determining the minimal order in

advance is useful to avoid the computation of a redundant amount of moments when using the moment-matching method in [115], since the method returns an ME representation of order n as long as n is smaller than or equal to the minimal order.

3.2.2 Non-Renewal Approximation

The approximation defined above matches a predefined number of moments of the stationary inter-overflow time distribution. Nevertheless, we can also include information related to the joint moments of consecutive inter-overflow times by means of a Rational Arrival Process (RAP) with parameters $\{H_0^i, H_1^i\}$ using the approach proposed in [111]. In this process the inter-event times are ME distributed, the matrix H_0^i describes the evolution of the process between events and H_1^i contains the arrival intensities. In the method of [111] the inter-overflow times are first approximated with an ME distribution using the algorithm in [115], as described above. Based on that result and the joint moments of the inter-overflow times, the method computes the matrices $\{H_0^i, H_1^i\}$ that describe the reduced RAP. If the reduced process is of order n, it not only matches $2n-1$ moments of the inter-overflow time distribution, but also $(n-1)^2$ joint moments of successive inter-overflow times. These matrices are used to approximate the overflow process as

$$\bar{D}_0^i = H_0^i, \qquad \bar{D}_j^i = H_1^i \frac{\pi_{C_i}^i D_j^i e}{\pi_{C_i}^i \left(D_+^i - D_s^i\right) e}, \quad j = 1, \ldots, N, j \neq s,$$

and $\bar{D}_s^i = 0$. The results obtained using a reduced RAP representation of order n are labeled RAP$(2n-1)$.

A further step consists of using the method in [59, 61] to directly approximate the matrices $\{\bar{D}_0^i, \bar{D}_1^i, \ldots, \bar{D}_N^i\}$ as the parameters of an MRAP[N]. To do so the inter-overflow time is again approximated by an ME distribution. However, in this case the description includes the joint moments of successive inter-overflow times for each type of arrival. This makes the method able to determine not just a matrix H_1^i describing the arrival intensities of any type, but a set of matrices $\{H_1^i, \ldots, H_N^i\}$ with the intensities for each type of arrival. These matrices, together with H_0^i, completely determine the approximate overflow process that is fed to the next station in the ring, i.e., $\bar{D}_j^i = H_j^i$, $j = 0, 1, \ldots, N$, where $H_s^i = 0$ by construction. The label MRAP$(2n-1)$ will be used to refer to the results obtained with an order-n MRAP[N] representation. Notice that the two methods discussed in this section rely on the algorithm in [115] to compute an approximate ME representation of the inter-overflow time distribution. Therefore, the density function of this ME representation is evaluated at several points to test its non-negativity, in a similar way as with the ME renewal approximation.

Before turning to the second approximation method, it is important to emphasize that in this method the size of the reduced representation of the overflow process does not depend on the size of the arrival process representation. Actually, it only depends on the number of moments and joint moments of the inter-overflow time process to match. Therefore, this method can be used when the arrival of new jobs at each station is described by a MAP, as explained in Section 3.1.1, since this generalization has no effect on the size of the overflow representation.

3.3 ON-OFF approximation

This approximation aims at capturing the behavior of the periods where a station does not generate overflow jobs (OFF periods) and the periods where it does (ON periods) separately. Furthermore, the reduction process is split into two steps: the first step is related to the reduction of the OFF period representation, and the second regards the reduction of the ON period representation. The first time each station is analyzed only the OFF period representation will be reduced. Thus, after analyzing all the stations once, the overflow process will include jobs from all the stations and those coming from the first station must be dropped. To eliminate those jobs we reduce the ON period representation by lumping the states related to the first station, i.e., the station generating the jobs that need to be eliminated. Hereafter the analysis of each station first reduces the representation of its OFF period, and then eliminates the jobs that must be dropped by reducing the ON period representation. The overflow process is again represented by an MMAP[N]. The first step computes the matrices $\{\hat{D}_0^i, \hat{D}_1^i, \ldots, \hat{D}_N^i\}$ of an MMAP[N] that still includes the jobs that must be dropped. The result of the second step is the set of matrices $\{\bar{D}_0^i, \bar{D}_1^i, \ldots, \bar{D}_N^i\}$ which characterizes the approximating overflow process. The size of the approximate process grows the first time each station is analyzed, but remains fixed in the subsequent iterations. In fact, if the external arrivals at each station follow a Poisson process, the size of the approximated process depends linearly on the number of stations. Assuming a more general process would cause the size of the approximated process to grow exponentially, making it intractable except for very small networks. Therefore, in the exposition to follow we assume that new incoming jobs at each station arrive according to a Poisson process.

To describe the behavior of the system during the OFF periods we consider, from the process with generator (3.1), those states where there is at least one idle server. The transient generator of these states is given by

$$E_{\text{OFF}}^i = \begin{bmatrix} D_0^i & D_+^i & 0 & \ldots & 0 & 0 \\ \mu_i I & D_0^i - \mu_i I & D_+^i & \ldots & 0 & 0 \\ 0 & 2\mu_i I & D_0^i - 2\mu_i I & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & D_0^i - (C_i-2)\mu_i I & D_+^i \\ 0 & 0 & 0 & \ldots & (C_i-1)\mu_i I & D_0^i - (C_i-1)\mu_i I \end{bmatrix}. \tag{3.6}$$

Since in steady state all the servers are busy with probability vector $\pi_{C_i}^i$, the distribution of the arrival phase after a service completion that causes the system to make a transition from the states $\{(C_i, l), 1 \le l \le m_i\}$ to the states $\{(C_i - 1, l), 1 \le l \le m_i\}$ is given by

$$\eta^i = \frac{\pi_{C_i}^i C_i \mu_i I}{\pi_{C_i}^i C_i \mu_i I e} = \frac{\pi_{C_i}^i}{\pi_{C_i}^i e}. \tag{3.7}$$

Thus, the stationary distribution of the duration of an OFF period can be described as a PH distribution with parameters $(\delta^i, E_{\text{OFF}}^i)$, where

$$\delta^i = \begin{bmatrix} 0 & \cdots & 0 & \eta^i \end{bmatrix}.$$

Since the size of this representation is $m_i C_i$, we can reduce it using the methods in [21, 110, 120] to match the first three moments of the OFF period length distribution, in the same manner as described in Section 3.2.1 for the inter-overflow time distribution. Let (α^i, A^i) be the reduced PH representation of the duration of the OFF period in station i, then α^i_j is the probability of starting an OFF period in phase j and $a^i = -A^i e$ is the rate at which an ON period starts in each phase of the OFF period representation. In addition, higher moments can be captured using an ME representation as illustrated in Section 3.2.1 for the inter-overflow time distribution. The results obtained with an approximation based on matching n moments of the OFF period length distribution will be labeled ON-OFF(n). However, in the remainder of this section we assume a PH representation as its interpretation is more intuitive.

On the other hand, because of the exponential service times, the duration of an ON period is exponentially distributed with rate $C_i \mu_i$. Furthermore, the stationary distribution of the arrival phase when an ON periods begins is given by

$$\omega^i = \frac{\pi^i_{C_i-1} D_+}{\pi^i_{C_i-1} D_+ e},$$

where $\pi^i_{C_i-1}$ is the stationary probability vector of having $C_i - 1$ busy servers in station i. Using this description we can connect the OFF and ON periods in a single MMAP[N] process with parameters $\{\hat{D}^i_0, \hat{D}^i_1, \ldots, \hat{D}^i_N\}$ given by

$$\hat{D}^i_0 = \begin{bmatrix} A^i & a^i \omega^i \\ C_i \mu_i e \alpha^i & D^i_0 - C_i \mu_i I \end{bmatrix}, \qquad \hat{D}^i_j = \begin{bmatrix} 0 & 0 \\ 0 & D^i_j \end{bmatrix}, \quad j = 1, \ldots, N. \tag{3.8}$$

The structure of the matrices $\{\hat{D}^i_j, 1 \leq j \leq N\}$ clearly shows that the description of the ON period remains unchanged. In contrast, the matrix $\hat{D}^i_0$ contains the reduced representation of the OFF period. The reduced process resides in the OFF states according to matrix A^i, from which it can move to the ON states with rates a^i and select a state in this set according to the vector ω^i. The transitions within the ON states are described by the matrices $\{D^i_j, 1 \leq j \leq N\}$ when an arrival is associated with the transition, and by the matrix D^i_0 when no arrivals are generated by the transition. Finally, the process may move from any of the ON states to the OFF set with rate $C_i \mu_i$. When this happens, a new OFF state is selected according to the vector α^i.

Let n_i be the order of the representation (α^i, A^i). Then, after analyzing station 1 for the first time (considering its own external Poisson traffic only), the approximate representation of the overflow process at this station is of size $n_1 + 1$. This process is combined with the Poisson process arriving at station 2 and, after reducing the OFF period representation at this station, the overflow process fed to station 3 is of order $n_2 + n_1 + 1$. Therefore, the size of the overflow process from station N to station 1 is $\sum_{k=1}^{N} n_k + 1$. Notice, the overflow process is built such that the first n_N states describe the OFF period in which no packets are sent to station 1. In the next n_{N-1} states, the station only overflows packets of type N since these states correspond to the case when the station N faces an ON period while station $N-1$ resides in an OFF period. Accordingly, in the next n_{N-2} states both jobs of type N and $N-1$ are overflowing and

from this set of states the process can jump to any of the previous states as well as to the next ones. This feature of the overflow process captures the actual behavior of the ring network where, for a job from station $N-1$ to be sent to station 1, both stations N and $N-1$ have to be in their ON periods. Furthermore, if the station N moves to the OFF set, both jobs of type N and $N-1$ stop overflowing to station 1.

To illustrate the transitions in the overflow process we consider a simple network made of three nodes. The approximate overflow process at the first station is characterized by the size-(n_1+1) matrices

$$\hat{D}_0^1 = \left[\begin{array}{c|c} A^1 & a^1 \\ \hline C_1\mu_1\alpha^1 & -(\lambda_1 + C_1\mu_1) \end{array}\right], \quad \hat{D}_1^1 = \left[\begin{array}{c|c} 0_{n_1} & 0 \\ \hline 0 & \lambda_1 \end{array}\right], \quad \hat{D}_2^1 = \hat{D}_3^1 = 0.$$

The size of the (square) diagonal blocks is indicated explicitly to prevent confusion. Notice, the vector ω^1 is not included explicitly in the definition of $\hat{D}_0^1$ since in this case the ON period is described by one state which is selected with probability one when a new ON period starts. At station two the arrival process is built by superposing the newly-generated and the overflow jobs, as shown in Equation (3.3). After reducing the OFF period in this station, the matrices that characterize the overflow from station two to station three are

$$\hat{D}_0^2 = \left[\begin{array}{c|c|c} A^2 & a^2\omega^2 & \\ \hline C_2\mu_2 e\alpha^2 & A^1 - (\lambda_2 + C_2\mu_2)I & a^1 \\ \hline & C_1\mu_1\alpha^1 & -\sum_{j=1}^{2}(\lambda_j + C_j\mu_j) \end{array}\right],$$

$$\hat{D}_1^2 = \left[\begin{array}{c|c|c} 0_{n_2} & 0 & \\ \hline 0 & 0_{n_1} & 0 \\ \hline & 0 & \lambda_1 \end{array}\right], \quad \hat{D}_2^2 = \left[\begin{array}{c|c|c} 0_{n_2} & 0 & \\ \hline 0 & \lambda_2 I_{n_1} & 0 \\ \hline & 0 & \lambda_2 \end{array}\right], \quad \hat{D}_3^2 = 0.$$

Finally, the reduction of the OFF period representation is applied to station three, which results in an overflow process described by an MMAP[3] of size $n_3 + n_2 + n_1 + 1$ with matrices

$$\hat{D}_0^3 = \left[\begin{array}{c|c|c|c} A^3 & a^3\omega^3 & & \\ \hline C_3\mu_3 e\alpha^3 & A^2 - (\lambda_3 + C_3\mu_3)I & a^2\omega^2 & \\ \hline & C_2\mu_2 e\alpha^2 & A^1 - \sum_{j=2}^{3}(\lambda_j + C_j\mu_j)I & a^1 \\ \hline & & C_1\mu_1\alpha^1 & -\sum_{j=1}^{3}(\lambda_j + C_j\mu_j) \end{array}\right],$$

$$\hat{D}_1^3 = \left[\begin{array}{c|c|c|c} 0_{n_3} & 0 & & \\ \hline 0 & 0_{n_2} & 0 & \\ \hline & 0 & 0_{n_1} & 0 \\ \hline & & 0 & \lambda_1 \end{array}\right], \quad \hat{D}_2^3 = \left[\begin{array}{c|c|c|c} 0_{n_3} & 0 & & \\ \hline 0 & 0_{n_2} & 0 & \\ \hline & 0 & \lambda_2 I_{n_1} & 0 \\ \hline & & 0 & \lambda_2 \end{array}\right], \quad \hat{D}_3^3 = \left[\begin{array}{c|c|c|c} 0_{n_3} & 0 & & \\ \hline 0 & \lambda_3 I_{n_2} & 0 & \\ \hline & 0 & \lambda_3 I_{n_1} & 0 \\ \hline & & 0 & \lambda_3 \end{array}\right].$$

The matrix $\hat{D}_0^3$ shows how the process can move from an OFF state to any of the ON states, even those where jobs of all types are generated. It also reveals how the service rate $C_i\mu_i$ determines the transition rates from the states where station i is in an ON period to those where it enters an OFF period. It is clear that the last state is the only one where jobs of every type are generated, including those coming from the first station, which must be dropped. Also, the last $n_1 + 1$ states describe the ON period of station two, where packets of type two are generated. This fact suggests that the overflow process may be reduced by combining the last $n_1 + 1$ states into one single state describing the ON period of station two.

In general, after analyzing all the stations once, the overflow process at station N still includes packets of type one. Since the relevant streams for this overflow process are those of type $\{2, \ldots, N\}$, the process could be reduced to have $\sum_{k=2}^{N} n_k + 1$ states, where the last $n_2 + 1$ states describe the OFF and ON periods of station two. Specifically, to reduce the representation of the overflow process and to eliminate the type-1 jobs, we consider lumping the last $n_1 + 1$ states. From [24], a CTMC is strongly lumpable with respect to some partition if, for every pair of sets $\mathcal{A}$ and $\mathcal{B}$ in the partition, the sum of the transition rates from a state in $\mathcal{A}$ to the states in $\mathcal{B}$ is the same for every state in $\mathcal{A}$. In fact, if we define the set $\mathcal{A}_1$ containing the last $n_1 + 1$ states of the process, and the partition $\mathcal{E} = \{1, 2, \ldots, \sum_{k=2}^{N} n_k, \mathcal{A}_1\}$, then the underlying chain of the overflow process is strongly lumpable with respect to $\mathcal{E}$. This can be easily seen in the construction of the overflow process. The transitions from and to the set $\mathcal{A}_1$ are contained only in the matrix $\hat{D}_0$, as can be seen from Equation (3.8). Since only the last $n_1 + 1$ states are lumped, the required condition for strong lumpability is automatically met for all the single-state sets in the partition. To see that the transition rates from any state $s \in \mathcal{A}_1$ to any state $t \notin \mathcal{A}_1$ are the same for every element in $\mathcal{A}_1$, define $\mathcal{A}_i$ as the set of states $\{\sum_{k=i+1}^{N} n_k + 1, \ldots, \sum_{k=i}^{N} n_k\}$, i.e., the set of states describing the OFF period of station i, for $2 \leq i \leq N$. Then the transitions from $\mathcal{A}_1$ to $\mathcal{A}_i$ occur with rates $C_i\mu_i e\alpha^i$, for $2 \leq i \leq N$, which do not depend on the specific state in $\mathcal{A}_1$. Thus, it is clear that the transition rate from any state $s \in \mathcal{A}_1$ to $t \notin \mathcal{A}_1$ is the same for all $s \in \mathcal{A}_1$. Clearly, this argument applies when the reduced representation of the OFF period is a PH distribution, as in this case the underlying process is a CTMC. When the reduced representation is an ME distribution this result no longer applies and we are not aware of a similar result involving this more general process. However, we apply the same reduction of the ON period, lumping the last n_1+1 states of the process, to eliminate the type-1 jobs from the overflow process at station N. This implies an additional approximation, but the results in Section 3.4 show that matching more moments, using ME distributions, improve the performance of the ON-OFF approximation.

Lumping the state space of the underlying CTMC (or the more general process based on ME distributions) of the overflow process is the key step to reduce the representation of the ON period and to determine the matrices $\{\bar{D}_0^N, \bar{D}_1^N, \ldots, \bar{D}_N^N\}$ from the matrices $\{\hat{D}_0^N, \hat{D}_1^N, \ldots, \hat{D}_N^N\}$. Let $\hat{D}_0^N(\mathcal{A}_j, \mathcal{A}_k)$ be the transition rates from the set $\mathcal{A}_j$ to the set $\mathcal{A}_k$ without arrivals. Also, let $\mathcal{A}_1'$ be the unitary set containing the last state of the overflow process created by lumping the states in $\mathcal{A}_1$. Then, the sub-generator matrix of

the overflow process at station N, partitioned according to $\{\mathcal{A}_N, \ldots, \mathcal{A}_2, \mathcal{A}_1'\}$, is given by

$$\bar{D}_0^N = \left[\begin{array}{cccc|c} \hat{D}_0^N(\mathcal{A}_N,\mathcal{A}_N) & \hat{D}_0^N(\mathcal{A}_N,\mathcal{A}_{N-1}) & \cdots & \hat{D}_0^N(\mathcal{A}_N,\mathcal{A}_2) & \hat{D}_0^N(\mathcal{A}_N,\mathcal{A}_1)e \\ \hat{D}_0^N(\mathcal{A}_{N-1},\mathcal{A}_N) & \hat{D}_0^N(\mathcal{A}_{N-1},\mathcal{A}_{N-1}) & \cdots & \hat{D}_0^N(\mathcal{A}_{N-1},\mathcal{A}_2) & \hat{D}_0^N(\mathcal{A}_{N-1},\mathcal{A}_1)e \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \hat{D}_0^N(\mathcal{A}_2,\mathcal{A}_N) & \hat{D}_0^N(\mathcal{A}_2,\mathcal{A}_{N-1}) & \cdots & \hat{D}_0^N(\mathcal{A}_2,\mathcal{A}_2) & \hat{D}_0^N(\mathcal{A}_2,\mathcal{A}_1)e \\ \hline C_N\mu_N\alpha^N & C_{N-1}\mu_{N-1}\alpha^{N-1} & \cdots & C_2\mu_2\alpha^2 & -\sum_{k=2}^N (C_i\mu_i+\lambda_i) \end{array}\right].$$

We now partition the state space into two sets, one including the first $\sum_{k=2}^N n_k$ states and the other being $\mathcal{A}_1'$. With this partition, the matrices $\{\bar{D}_1^N, \ldots, \bar{D}_N^N\}$ can be expressed as

$$\bar{D}_1^N = 0, \qquad \bar{D}_j^N = \left[\begin{array}{cc} \bullet & 0 \\ 0 & \lambda_i \end{array}\right], \quad j = 2, \ldots, N,$$

where $\bullet$ denotes the transitions in $\hat{D}_j^N$ from $\{\mathcal{A}_N, \ldots, \mathcal{A}_2\}$ to itself. This step concludes the reduction of the overflow representation related to the ON period. Furthermore, this reduction must be repeated at each station from the second iteration onwards, since for the first iteration the reduction is only related to the OFF period. Thus, the representation during any future iteration of the overflow process that arrives at station i is of order $\sum_{k=1,k\neq i}^N n_k + 1$. Given that it is usually possible to match the first three moments of the OFF periods in each station with a PH distribution of order 2, the size of the overflow process representation is $2N-1$, which increases linearly with the number of stations. In contrast to the approximation proposed in Section 3.2, the ON-OFF method is not able to efficiently include more general arrival processes at the stations because the size of the arrival process representation at each station would increase exponentially.

3.4 Performance Results and Comparisons

In this section we evaluate the performance of the approximations introduced in this chapter by comparing their results against those obtained by simulation, for different values of the network parameters. We consider different values for the overall load, the number of stations in the network, the number of servers per station and the squared coefficient of variation (SCV) of the arrival process at each station. The results of the approximations presented in Section 3.2 are labeled depending on the specific representation of the inter-overflow time process, which can be a PH or ME distribution, a RAP or a marked RAP (MRAP). In each case the number of moments of the inter-overflow time distribution that are matched is made explicit, e.g., ME(n) refers to an approximation based on an ME representation matching n moments. The results of the approximation introduced in Section 3.3 are labeled ON-OFF(n) when n moments of the OFF period length distribution are matched.

The main performance measures are the number of newly-generated jobs processed locally per unit of time (*local rate*), and the total traffic rate transmitted from one station to another (*link traffic*). These measures have been chosen as they are of particular

relevance when dimensioning the optical grid resources [35,38]. Since these measures may be different for each station in the network, we present the maximum *relative* error found when comparing them against the simulation results. Let r_i^{M} (resp. r_i^{S}) be the local rate for station i computed with the approximation method M (resp. the local rate estimated via simulation). Then, the index of the station with maximum *absolute relative* error on local rate for the method M is $j_M = \arg\max_i\{|r_i^{\mathrm{M}} - r_i^{\mathrm{S}}|/r_i^{\mathrm{S}}\}$. The maximum *relative* error is thus given by $(r_{j_M}^{\mathrm{M}} - r_{j_M}^{\mathrm{S}})/r_{j_M}^{\mathrm{S}}$. A similar definition applies for the maximum relative error on link traffic. Apart from the maximum relative error, for one scenario we also include a figure that displays the absolute error for each station. The widths of the 95%-confidence intervals obtained from simulation (computed with the batch means method) are always less than one percent of the respective mean. All the network configurations considered here are assumed to have an overall load between 75% and 95%. The load at each station is randomly chosen within the range $[x - 5\%, x + 5\%]$, where x is the overall load of the network. In prior studies the number of stations ranges from less than ten to hundred, with typical values between twenty and fifty [31,35,36,38]. Also, typical values for the number of servers per station are between twenty and fifty, although in some cases it can be larger than one hundred. We consider networks with 10, 20, and 40 stations, each having the same number of servers (20 or 40).

Load	**0.75**		**0.80**		**0.85**		**0.90**		**0.95**	
# Servers	20	40	20	40	20	40	20	40	20	40
PH(3)	0.10	0.02	0.19	0.07	0.20	0.22	3.02	0.34	14.59	7.69
ME(5)	0.05	0.03	0.15	0.03	0.13	0.14	2.51	0.25	12.49	6.70
RAP(5)	0.03	0.03	0.09	0.03	0.19	0.09	2.91	0.22	12.15	7.21
MRAP(5)	0.03	0.03	0.09	0.03	0.19	0.09	2.83	0.22	11.34	7.04
ON-OFF(3)	0.11	0.02	0.29	0.08	0.45	0.29	1.64	0.67	9.66	5.01

Table 3.1: Maximum *relative* error (%) in local rate for N = 10 and C = {20, 40}

Load	**0.75**		**0.80**		**0.85**		**0.90**		**0.95**	
# Servers	20	40	20	40	20	40	20	40	20	40
PH(3)	-7.96	-3.74	-12.57	-7.30	-17.94	-13.76	-15.53	-25.39	10.62	-6.93
ME(5)	-3.94	-0.96	-8.26	-3.26	-13.47	-8.80	-10.51	-19.64	12.63	-1.70
RAP(5)	-3.06	-0.88	-6.87	-2.75	-11.23	-7.71	-6.29	-17.32	16.59	3.47
MRAP(5)	-3.06	-0.88	-6.87	-2.75	-11.25	-7.71	-6.68	-17.33	13.44	2.79
ON-OFF(3)	-5.17	-2.06	-9.58	-4.99	-14.91	-10.88	-13.24	-22.04	4.57	-6.60

Table 3.2: Maximum *relative* error (%) in link traffic for N = 10 and C = {20, 40}

We start with a network made of ten nodes with 20 and 40 servers per station. The *relative* errors in the approximation of the local rate and link traffic are included in Tables 3.1 and 3.2, respectively. The approximate local rates are very close to those obtained with simulation, particularly for loads up to 90%. When the load is higher the approximations

that also match the joint moments of successive inter-overflow times perform significantly better than those based on renewal processes. The ON-OFF method shows a competitive performance, with similar results to those of the other methods for loads under 90%, but better behavior for higher loads. Additionally, all the methods perform better when the number of servers increases from 20 to 40, especially for high loads. In relation to the link traffic (Table 3.2), the errors in the approximations are clearly larger than for the local rate. Although for loads from 75% to 85% the errors grow with the load, this is no longer the case if the load is further increased. Similarly, for the same range of loads, the errors are smaller for the system with 40 servers per station than for the one with 20. Again, this behavior does not hold for higher loads. For loads up to 85% the methods provide a reasonable approximation, especially the ones based on the RAP and RAPK representation of the overflow process.

Load		**0.75**		**0.80**		**0.85**		**0.90**		**0.95**	
# Stations		20	40	20	40	20	40	20	40	20	40
ME	3	0.06	0.06	0.07	0.11	0.18	0.19	0.74	0.66	2.40	1.02
	5	0.05	0.05	0.04	0.06	0.11	0.12	0.64	0.56	1.68	1.63
	7	0.05	0.05	0.05	0.07	0.15	0.16	0.77	0.70	1.19	2.12
(M)RAP	3	0.06	0.06	0.06	0.10	0.16	0.17	0.63	0.56	2.97	0.40
	5	0.05	0.05	0.04	0.05	0.06	0.08	0.43	0.35	2.47	0.73
	7	0.05	0.05	0.04	0.05	0.10	0.11	0.56	0.49	1.86	1.32
ON-OFF	3	0.06	0.06	0.09	0.11	0.25	0.25	1.05	0.98	0.46	2.84
	5	0.05	0.04	0.06	0.08	0.19	0.19	0.94	0.87	0.47	2.82
	7	0.05	0.04	0.06	0.08	0.19	0.20	0.95	0.87	0.43	2.86

Table 3.3: Maximum *relative* error (%) in local rate for N = {20, 40} and C = 40

We now turn to the analysis of larger systems that consist of 20 and 40 stations, each with 40 servers. The approximation errors for the local rate are included in Table 3.3, while those for the link traffic are in Table 3.4. In this case we provide the results for an increasing number of matched moments of the distribution of both the inter-overflow time and the OFF period length. The results labeled (M)RAP correspond to both RAP and MRAP representations since both have very similar results. This similarity was also apparent in Tables 3.1 and 3.2, as well as in many other scenarios not presented here. The approximation of the local rate is again very close to the estimates obtained by simulation, with improved results when compared to the smaller network. In general, the error in approximating the local rate diminishes with the inclusion of higher moments, with the exception of high loads for the system with 40 stations. On the other hand, the errors in the link traffic approximation decrease consistently when matching more moments. This effect is more evident for the methods based on the inter-overflow time process (ME, RAP, MRAP) than for the ON-OFF method. For all the methods, the accuracy of the approximations improves drastically when matching five instead of three moments, and the results are even better when matching up to seven moments, although the difference is not as large as in the first case. It is important to note that the approximation methods

fail to include more than seven moments because the algorithm in [115] returns a kernel matrix that corresponds to a negative density. In many other cases we obtained a similar result when trying to include nine or more moments.

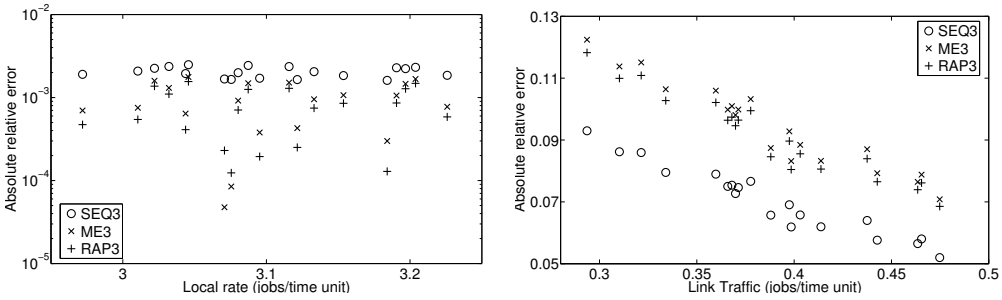
Load		**0.75**		**0.80**		**0.85**		**0.90**		**0.95**	
# Stations		20	40	20	40	20	40	20	40	20	40
ME	3	-5.64	-6.03	-9.38	-9.80	-12.24	-12.30	-24.44	-21.91	-30.35	-37.76
	5	-2.07	-2.57	-4.48	-4.92	-7.18	-7.24	-18.73	-16.01	-25.58	-33.50
	7	-1.62	-2.12	-3.83	-4.28	-6.46	-6.53	-17.83	-15.08	-24.67	-32.69
(M)RAP	3	-5.55	-5.92	-9.10	-9.53	-11.82	-11.88	-23.56	-21.00	-28.70	-36.29
	5	-1.86	-2.29	-3.79	-4.24	-6.09	-6.15	-16.41	-13.61	-21.28	-29.66
	7	-1.37	-1.78	-2.98	-3.43	-5.10	-5.16	-14.93	-12.08	-19.29	-27.87
ON-OFF	3	-3.59	-4.06	-6.54	-6.97	-9.30	-9.36	-21.04	-18.39	-27.30	-35.00
	5	-1.74	-2.26	-4.11	-4.55	-6.88	-6.94	-18.58	-15.85	-25.69	-33.55
	7	-1.57	-2.10	-3.87	-4.32	-6.62	-6.68	-18.27	-15.53	-25.43	-33.30

Table 3.4: Maximum *relative* error (%) in link traffic for N = {20, 40} and C = 40

When comparing the ON-OFF approach with the inter-overflow time method it is clear that neither of them outperforms the other. In particular, when the overall load is equal to 95% the ON-OFF method has a better performance in approximating the local rate for the scenario with 20 stations while it is worse for the one with 40 stations. For lower loads, both methods show a similar performance, with a slightly smaller error for the ME and (M)RAP methods. However, the link traffic results show a different behavior. As this measure includes a mixture of all the traffic in the network, the errors accumulate causing larger discrepancies between the approximate methods and the simulation results. These discrepancies tend to increase with the size and the overall load of the network, as each link carries more types of jobs when any of these parameters increases. In particular, the ON-OFF method shows a better performance (in estimating link traffic) than the other methods when matching three moments of the corresponding inter-event time distribution. When five moments are matched the (M)RAP method performs better than the ON-OFF, which now has similar errors than the ME approach. Finally, all the methods based on the inter-overflow time process perform better than the ON-OFF approach when seven moments are matched. This reveals that the inclusion of higher moments has a greater effect on the methods based on the inter-overflow time process, and this effect is more apparent for the non-renewal representations.

In addition to the maximum relative error, we now introduce Figure 3.1 to illustrate the absolute relative errors for all the stations in the network. Figure 3.1(a) displays the absolute relative error in local rate, while Figure 3.1(b) shows the absolute relative error in link traffic. Each point in the figures corresponds to one station or one link, for the network with 20 stations and 40 servers per stations. For the local rate, we observe that the errors are very similar, with little variation among the stations. Therefore, the maximum relative error shown in the tables is also close to the average relative error. On the other hand, we find that the error in estimating the link rate actually decreases when

the link rate increases. For instance, the largest error for the SEQ3 method is above 12% and corresponds to the link carrying the least amount of traffic. On the other end is the link with the highest traffic rate (above 0.47 jobs per time unit), for which the relative error is close to 5%. This example illustrates a behavior that can be found in all the other scenarios we have considered. This behavior is actually relevant for the applicability of the method since it shows that the errors in approximating the link traffic are smaller for the links that carry more traffic, for which a larger investment must be made to provide the necessary bandwidth.



(a) Error in Local Rate (b) Error in Link Traffic

Figure 3.1: Absolute errors for each station, $N = 20$, $C = 40$

In relation to the computation times, the inter-overflow time method has a clear advantage since the size of the approximate representation of the overflow process is independent of the number of stations. In the ON-OFF method this size grows linearly with the number of stations, which generates a proportional increase in the block size of the QBD that must be solved at each station. The relevance of the block size for the computation times comes from the fact that the time complexity of the algorithms to compute the stationary distribution and the moments of the distribution of the first-passage times to higher levels of the QBD is cubic in the block size. For the inter-overflow time method, the computation times were always below one minute in all the scenarios considered here. In the scenarios with up to 20 stations, the ON-OFF method required seven minutes in the worst case. However, this only occurred when 7 moments of the OFF-period length distribution were matched, under an overall load of 95%. When only three moments are matched, the computation times were always below one minute. For the scenarios with 40 stations, the ON-OFF method required up to six minutes when matching 3 moments. If more moments are matched the computation times become too long for practical purposes. If the number of nodes in the network is large it is better to use the inter-overflow time methods, since this number does not affect the computation times for this method. Notice that the load also affects the computation times as more iterations are required for the traffic matrix to converge when the load is higher. For the scenarios shown here, usually less than five iterations were enough when the load was less than 80%, but this figure was between 20 and 30 when the load was 95%.

For the results considered thus far, the arrival process of newly-generated jobs at each station is assumed to be Poisson. Even though this is a usual assumption in prior

Load		**0.75**		**0.80**		**0.85**		**0.90**		**0.95**	
SCV		5	20	5	20	5	20	5	20	5	20
ME	3	0.14	0.76	0.40	1.09	0.33	1.25	1.59	4.63	10.73	15.22
	5	0.09	0.34	0.17	0.62	0.33	0.37	0.66	3.28	9.27	13.25
	7	0.14	0.59	0.28	1.11	0.59	1.21	0.59	0.94	8.24	-
(M)RAP	3	0.16	0.80	0.42	1.17	0.42	1.39	1.82	4.90	11.42	15.77
	5	0.08	0.39	0.20	0.49	0.17	0.66	0.98	3.87	10.21	14.51
	7	0.11	0.51	0.22	0.94	0.47	0.91	0.28	1.71	9.00	8.90

Table 3.5: Maximum *relative* error (%) in local rate for N = 20, C = 40, SCV = {5, 20}

Load		**0.75**		**0.80**		**0.85**		**0.90**		**0.95**	
SCV		5	20	5	20	5	20	5	20	5	20
ME	3	-9.77	-20.91	-15.51	-29.90	-23.16	-38.03	-34.36	-42.86	-26.85	-34.42
	5	-5.79	-17.38	-10.46	-25.72	-17.87	-33.11	-28.87	-36.71	-20.77	-26.19
	7	-5.08	-16.16	-9.48	-24.32	-16.77	-31.53	-27.59	-34.48	-19.21	-
(M)RAP	3	-9.59	-20.69	-15.17	-29.61	-22.67	-37.67	-33.61	-42.41	-25.51	-32.62
	5	-5.28	-16.49	-9.48	-24.54	-16.49	-31.62	-26.71	-34.63	-16.74	-21.86
	7	-4.39	-14.84	-8.17	-22.54	-14.88	-29.23	-24.60	-31.25	-13.59	-16.20

Table 3.6: Maximum *relative* error (%) in link traffic for N = 20, C = 40, SCV = {5, 20}

studies on optical grid networks, more general processes can also be considered in order to capture the high variability of the arrival process [31]. As mentioned in Section 3.1, the method based on the inter-overflow time distribution can be extended to allow for more general arrival processes without experiencing an exponential increase in the size of the overflow process representation. To consider the case of high variability in the arrival process, we assume that each station receives newly-generated jobs coming from a hyper-exponential renewal process with different arrival rates at each station, but the same SCV. Specifically, the cases where the SCV is equal to 5 and 20 are analyzed for a network with 20 stations and 40 servers per station. The approximation errors in local rate and traffic link are included in Tables 3.5 and 3.6, respectively. Again, the RAP and MRAP representations are presented together as their results are very similar. The first clear result is that the increase in the SCV causes larger errors in the approximation of both the local rate and the link traffic. Not only are the errors for SCV equal to 20 larger than those for SCV equal to 5, but in both cases the approximation is worse than under Poisson arrivals (Tables 3.3 and 3.4). With respect to the local rate, the effect of matching higher moments becomes more evident for larger SCV. The approximations offer small errors for loads up to 90%, and for higher loads these are still below 10% (when matching seven moments of the inter-overflow time distribution). The results for the ME(7) approximation with an SCV equal to 20 are not included because the density function obtained with the algorithm in [115] was negative. For the link traffic, the

effect of matching higher moments is evident, with a stronger effect for the non-renewal approximations (RAP, MRAP). The errors tend to increase with the load, except for a load of 95%.

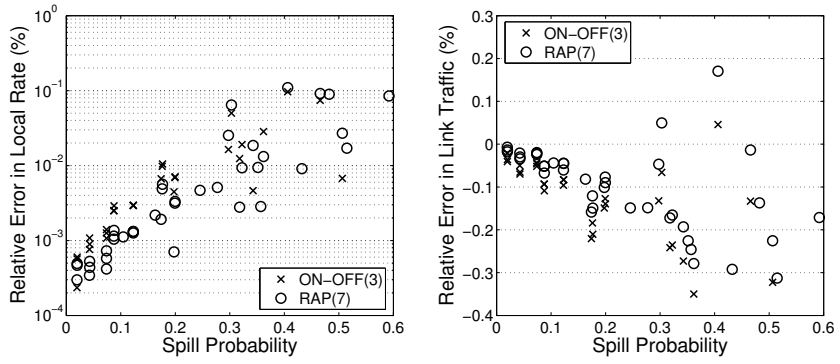


Figure 3.2: Maximum *relative* errors in local rate and link traffic as a function of the spill probability

From the instances analyzed here, and others not included, we have found that a larger SCV of the arrival process and higher overall load deteriorate the performance of the approximation methods. The number of servers has an opposite effect, i.e., the approximation errors are smaller when the number of servers is larger. These observations are related to the impact that these parameters have on the spill probability (the probability that a job has to be sent to a remote station for service). Clearly, both the arrival process SCV and the load increase the spill probability, while, under the same load, a larger number of servers reduces the spill probability as more jobs can be handled locally. Figure 3.2 shows the errors in local rate and link traffic as a function of the spill probability for two specific methods: ON-OFF(3) and RAP(7). There we observe that the approximations behave better when the spill probability is small, typically under 0.3. For a larger spill probability the approximation errors become more pronounced, especially for the link traffic. Besides, in most of the cases the approximation methods underestimate the actual link traffic. Thus, the results of these methods can be regarded as a practical lower bound of the actual link traffic.

Chapter 4

A Grid network with a large number of sites

In this chapter we introduce an analytical framework to compute the traffic matrix in a grid network with a large number of sites. As mentioned in the introduction to this Part, one of the greatest obstacles when modeling a Grid is the need to keep track of the state of all the sites, the number of which may be between some tens and a few hundreds. Therefore the methods used to analyze these networks (e.g. to compute the traffic matrix) must be able to overcome this limitation. In the previous chapter we presented a method to approximate the traffic matrix by exploiting a particular topology. In this chapter we take a different path by introducing a mean field model that is exact when the number of sites in the network tends to infinity. This model can therefore be used to approximate the performance of a network with a large but finite number of sites. We will show that this method is practical for a realistic grid dimensioning case and closely matches the results obtained with (time consuming) simulations.

Our mean field model is defined within the framework introduced in [79], where a general convergence result is obtained for a system of interacting objects. The idea of the mean field framework in [79] is that the behavior of a system made of N objects converges toward (and therefore can be approximated by) a deterministic dynamical system when N tends to infinity. The description of the deterministic system requires a state space with the same size as that of a single object in the system, meaning that the deterministic system lacks the dimensionality issues of modeling a system made of many objects. A basic description of the framework can be found in Appendix A.3 and for a detailed treatment we refer the reader to [79].

In our model the sites of the network are the objects that interact to redistribute the jobs that cannot be processed locally. The model is in discrete time, i.e., the time is divided in slots or epochs of fixed duration. The sites in the grid are characterized by the number of local servers they have, their arrival process and the job processing rate. These characteristics are used to segregate the sites in classes. When a job arrives to a site, it is served locally if there is a server available. If all the local servers are busy the job is sent to another site which is selected according to a scheduling algorithm, for which we consider

two alternatives: *random* and *mostfree*. In both cases, the scheduling is made without considering the topology of the network, as this aspect is not taken into account by the model. As discussed in Appendix A.3, in the mean field framework the state transitions of a single object (site) cannot depend on the state of another specific object, but on the fraction of objects in each possible state only. This limits the analysis of scheduling algorithms to those that do not take into account the topology of the network. For this type of algorithms the model to be introduced allows the computation of the traffic matrix, which can then be used to dimension the inter-site links. Moreover, we show that the mean field model can be used to speed up the dimensioning cycle by avoiding simulations.

This chapter is organized as follows. In Section 4.1 we start by pointing out the main characteristics and assumptions of the grid network model. Then, Section 4.2 introduces the mean field model assuming that all the sites in the network are identical, i.e., the single-class grid. This is followed by Section 4.3, where we generalize the model to allow for multiple classes of sites, which is the more realistic case. Here we also show how to compute the traffic matrix, which is the main objective of the model. Finally, in Section 4.4 we compare the performance of the two scheduling algorithms considered and investigate the effect of the load and the number of servers on the network's performance. We end the chapter by analyzing a realistic scenario and showing that the results of the mean field model match very well with those obtained by means of simulation. The material presented in this chapter is the result of joint work with the INTEC research group at Ghent University.

4.1 The Grid network model

We consider a grid network consisting of N sites, partitioned into K classes, assuming all sites belonging to the same class have the same characteristics:

1. a class-k site has $C^{(k)}$ identical servers,

2. the inter-arrival times (IATs) of the jobs originating at a class-k site are independent and identically distributed (i.i.d.) and follow a discrete phase-type (DPH) distribution with parameters $(\alpha^{(k)}, T^{(k)})$ (see Appendix A.1 for a description of this class of distributions),

3. processing a job at a class-k site takes a geometric amount of time with mean $1/p^{(k)}$.

This class partitioning may seem to limit the applicability of the mean field solution discussed below. However, as we will illustrate in Section 4.4 for a realistic scenario, clustering techniques can be used to achieve such partitioning into a limited number of classes.

The model is a discrete-time model where at each time epoch three sequences of events occur:

S1. *Service completions:* each class-k busy server becomes idle with probability $p^{(k)}$.

S2. *Arrivals:* at each site either 0 or 1 job arrives with a probability depending on the underlying phase of the arrival process at that particular site (the model can easily be extended to batch arrivals). If a job arrives at a site with at least one local server available after step S1, the job is processed locally. Otherwise, the job becomes part of the pool of excess jobs.

S3. *Excess redistribution:* all the excess jobs generated in step S2 are distributed among the servers that remained idle after step S2.

To redistribute j excess jobs among s idle servers in step S3, we consider two redistribution schemes: *mostfree* and *random*. Clearly, if $j > s$, all servers become occupied and we drop $j - s$ jobs. For $j \leq s$, the *mostfree* strategy will assign the j jobs one by one, each time selecting the site with the highest number of free servers (at the time of assignment). The *random* strategy simply selects the j servers at random among the s available ones, without considering the occupancy level of the site to which a server belongs.

The mean field analysis presented below computes exact results for the limiting system behavior when the number of sites per class goes to infinity. However, the number of sites per class does not have to be identical: if the number of class-k sites is defined as $\gamma_k N$ (where $\sum_k \gamma_k = 1$), then the limiting behavior corresponds to letting N approach infinity. Our case study will show that for practical site counts (some tens to a few hundreds), the limit behavior matches quite well with simulations for a finite number of sites.

4.2 A mean field solution for the single class Grid network

We first consider a Markovian model for the single class Grid network ($K = 1$); as such we can temporarily drop the superscript $^{(k)}$. For example, if the Grid dimensioning equally distributes server capacity over the chosen server locations, all these locations are identical in terms of server/processing capacity, each site having C servers. If we also assume all server locations have the same job arrival process, this amounts to a single class grid network.

The idea of the Markovian model is to associate $(C + 2)h$ states with each site. State $\langle i, j \rangle$, with $0 \leq i \leq C + 1$ and $1 \leq j \leq h$, indicates that i jobs are present at the site after step S2, while the arrival process is in phase j. Given that we have N sites, we get a total of $h^N(C + 2)^N$ states, which clearly can become huge. However, the mean field computation will be restricted to matrices of size $(C + 2)h$ and therefore turns out to be very effective (for details see Appendix A.3). The core of the model is the state transition matrix for a single object, and we now turn to its definition. To build this matrix we start by defining the transition matrices associated to each of the three steps in a slot described in the previous section. This is the topic of the next sections, after which we will combine these matrices into a single one to describe the slotted evolution of a single object.

4.2.1 Step S1, service completions

Given that i of the C servers in a site are busy, i' of them will become available with probability $s_{i,i-i'} = \binom{i}{i'} p^{i'}(1-p)^{i-i'}$, for $0 \leq i' \leq i$ and $0 \leq i \leq C$. Therefore, the state of a single site evolves in this step according to the $(C+1)h \times (C+1)h$ triangular matrix S, given by

$$S = \begin{bmatrix} s_{0,0} & 0 & \dots & 0 \\ s_{1,0} & s_{1,1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ s_{C,0} & \dots & s_{C,C-1} & s_{C,C} \end{bmatrix} \otimes I_h,$$

where I_h is the identity matrix of size h (reflecting the fact that the phase of the arrival process is not influenced by the service completions) and $\otimes$ denotes the Kronecker product between matrices.

4.2.2 Step S2, job arrivals

Given that the PH arrival process (see Appendix A.1) is in state j, it will generate an arrival and go to state j' with probability $[\theta\alpha]_{j,j'}$, while with probability $[T]_{j,j'}$ a similar transition occurs without involving an arrival. This means that the evolution of a site in this step can be described by the $(C+1)h \times (C+2)h$ matrix A, defined as

$$A = \begin{bmatrix} T & \theta\alpha & 0 & \dots & 0 \\ 0 & T & \theta\alpha & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & T & \theta\alpha \end{bmatrix}.$$

4.2.3 Step S3, excess redistribution

The transitions for a single site described in the previous steps depend only on the state of the site itself. However, in the redistribution step the state of a site is affected by the state of all the other sites in the network. Therefore, the state transitions at time t due to the redistribution of the excess jobs is influenced by the occupancy vector $M^N(t) = 1/N\,[a_0(t), a_1(t), \dots, a_C(t), a_{C+1}(t)]$, where

- $a_i(t)$, for $i = 0, \dots, C$, is the number of sites with i busy servers after step S2 that do not have an excess job, while
- $a_{C+1}(t)$ indicates the number of sites with an excess job (i.e., the total number of excess jobs at time t). Clearly, these sites have all their C servers occupied.

Thus, the i-th entry $M_i^N(t)$ of the vector $M^N(t)$ equals the fraction of sites holding i jobs (including excess jobs) after step S2. As stated in Section 4.1, we consider two different scheduling algorithms to redistribute the excess job. As the algorithm affects the state transitions, we consider them separately, starting with the *mostfree* algorithm.

Mostfree scheduling

Let $q_{i,i'}(M^N(t))$ be the probability that a site receives $i' - i \geq 0$ excess jobs, given that it held $i \leq C$ jobs after step S2. The mostfree strategy works as follows:

- the first $a_0(t)$ excess jobs will be assigned to the sites with all their servers available,
- the next $a_0(t) + a_1(t)$ excess jobs are forwarded to the sites that had either 0 or 1 busy servers (after this step all the sites with 0 busy servers received two excess jobs, while those with 1 busy server received 1 excess job),
- this continues until all $a_{C+1}(t)$ jobs have been distributed among the free servers or until all servers are busy.

For ease of notation define $b_i(t) = \sum_{k=0}^{i} a_k(t)$ as the number of sites with at most i busy servers after step S2. Provided that we have enough free servers to support the excess jobs, we can find an integer c, with $0 \leq c < C$, such that

$$\sum_{k=0}^{c-1} b_k(t) < a_{C+1}(t) \leq \sum_{k=0}^{c} b_k(t), \tag{4.1}$$

which we denote as $c(M^N(t))$ (for $a_{C+1}(t) = 0$, we set $c = 0$). In other words, all sites with $i \leq c(M^N(t))$ busy servers after step S2 ($b_{c(M^N(t))}(t)$ in total) will end up with at least $c(M^N(t))$ jobs and some of them with $c(M^N(t))+1$ jobs, after step S3. The fraction of these sites that end up with $c(M^N(t))$ jobs equals

$$\beta_{c(M^N(t))} = \frac{\sum_{k=0}^{c(M^N(t))} b_k(t) - a_{C+1}(t)}{b_{c(M^N(t))}(t)}. \tag{4.2}$$

Thus, a site with i busy servers receives $i' - i \geq 0$ jobs with probability $\beta_{c(M^N(t))}$ if $i' = c(M^N(t))$, and with probability $1 - \beta_{c(M^N(t))}$ if $i' = c(M^N(t)) + 1$. If the number of free servers $\sum_{k=0}^{C-1} b_k(t)$ is insufficient to support the $a_{C+1}(t)$ jobs, we let $c(M^N(t))$ equal to C. In this case all the servers become occupied. This yields,

$$q_{i,i'}(M^N(t)) = \begin{cases} 1 - \beta_{c(M^N(t))}, & i < i' = c(M^N(t)) + 1 \leq C, \\ \beta_{c(M^N(t))}, & i \leq i' = c(M^N(t)) < C, \\ 1, & i = i' > c(M^N(t)) \\ 1, & i' = C = c(M^N(t)), \end{cases}$$

for $0 \leq i, i' \leq C$. The third case indicates that no jobs are received when the site has $i > c(M^N(t))$ busy servers.

Now we are able to write down the $(C + 2)h \times (C + 1)h$ transition matrix $Q(\cdot)$ that describes the evolution of a single site during the reallocation step, under the mostfree strategy. It is given by

$$Q(M^N(t)) = \begin{bmatrix} q_{0,0}(M^N(t)) & q_{0,1}(M^N(t)) & \dots & q_{0,C}(M^N(t)) \\ 0 & q_{1,1}(M^N(t)) & \dots & q_{1,C}(M^N(t)) \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & 0 & q_{C,C}(M^N(t)) \\ 0 & \dots & 0 & 1 \end{bmatrix} \otimes I_h, \tag{4.3}$$

where the 1 in the lower right corner indicates that a site with $C+1$ jobs after step S2 will end up with C jobs after step S3 (either due to a redistributed or a dropped job). Notice that the entries of this matrix are independent of N, as can be confirmed by taking the definition of $\beta_{c(M^N(t))}$ and dividing the numerator and the denominator by N.

Random scheduling

We now consider the random redistribution strategy for the excess jobs, and define $\bar{Q}^N(\cdot)$ analogously to mostfree's $Q(\cdot)$. In this case however the matrix $\bar{Q}^N(\cdot)$ depends on N (see below) and this is made explicit. To define the evolution of a single site assume that a site s has i busy servers. In total there are $f(M^N(t)) = \sum_{k=1}^{C} a_{C-k}(t)k$ servers to choose from and $C-i$ of them belong to site s. Therefore the probability that $0 \leq i' \leq C-i$ excess jobs are assigned to site s, equals

$$\bar{q}^N_{i,i+i'}(M^N(t)) = \frac{\binom{C-i}{i'}\binom{f(M^N(t))-(C-i)}{a_{C+1}(t)-i'}}{\binom{f(M^N(t))}{a_{C+1}(t)}}, \tag{4.4}$$

provided that $f(M^N(t)) \geq a_{C+1}(t)$. Otherwise we have that $\bar{q}^N_{i,C}(M^N(t)) = 1$, for all i.

4.2.4 Computation of $M^N(t)$ for large N

To obtain a useful DTMC description of the system, we will observe it at each time epoch immediately after step S2 and before step S3. Given the state $\langle i, j\rangle$ of site s at time t (with i the number of jobs and j the service phase), we can obtain its system state at time $t+1$, which depends on the value of $M^N(t)$, via the transition matrix $R^N(M^N(t))$ defined as

$$R^N(M^N(t)) = Q(M^N(t))SA,$$

for the *mostfree* strategy. For the *random* strategy, we simply replace $Q(\cdot)$ by $\bar{Q}^N(\cdot)$ to obtain $\bar{R}^N(\cdot)$. Since the state evolution of different sites is correlated, the transition matrix of the entire system is hard to express. Luckily, with the mean field framework introduced in [79] (described in Appendix A.3) we can obtain a simple transition matrix for the entire system when it is composed of an infinite number of sites. Therefore we now consider the framework of [79] and discuss how it can be applied for our grid network model.

The main result in [79] states that as the number of objects becomes large, the occupancy measure of the system $M^N(t)$ converges to a deterministic dynamical system (the mean field), whose transition matrix has the same dimension as that of a single object. The state of the mean field at time t is described by the occupancy vector $\mu(t)$, which can then be used to approximate the state of a system with a large but finite number of objects. The convergence result is proved to hold if, for any occupancy vector, there exists a matrix $R(m)$, such that for each entry $[R^N(m)]_{s,s'}$, the set of functions $\{[R^N(m)]_{s,s'}, N \geq 1\}$ converges uniformly to $[R(m)]_{s,s'}$ on the set of all possible occupancy vectors m. For the *mostfree* model this convergence is immediate as the $R^N(m)$

matrices are independent of N. As stated before, this can be seen by simply dividing all the $a_k(t)$ and $b_k(t)$ appearing in $Q(\cdot)$ by N. In fact, this was already made explicit when the matrix $Q(\cdot)$ was defined as being independent of N. On the other hand, for the *random* strategy the matrix $\bar{Q}^N(\cdot)$, and therefore $\bar{R}^N(\cdot)$, depends on N. For any $1 \times (C+2)$ occupancy vector $m = [m_0, m_1, \ldots, m_C, m_{C+1}]$, let us to define $f(m) = \sum_{j=1}^{C} j m_{C-j}$. Then, if $m_{C+1} \leq f(m)$, let $\bar{q}_{i,i+i'}(m)$ be given by

$$\bar{q}_{i,i+i'}(m) = \binom{C-i}{i'} \left(\frac{m_{C+1}}{f(m)}\right)^{i'} \left(1 - \frac{m_{C+1}}{f(m)}\right)^{C-i-i'}. \tag{4.5}$$

Otherwise, let $\bar{q}_{i,C}(m) = 1$, for all i. Finally, define $\bar{Q}(m)$ analogously to Equation (4.3). It is not hard to show that the set of functions $\{[\bar{R}^N(m)]_{s,s'}, N \geq 1\}$, for any s, s' converges uniformly to $[\bar{R}(m)]_{s,s'} = [\bar{Q}(m)SA]_{s,s'}$ on the set of all occupancy vectors m. The main argument to show this is the convergence [108] of the hyper-geometric probabilities in Equation (4.4) to the binomial probabilities in Equation (4.5).

As explained in Appendix A.3, the result in [79] also requires $R(m)$ to be continuous in m, which is clearly the case for both *mostfree* and *random*. A final condition is that $M^N(0)$ converges uniformly to some $\mu(0)$ when $N \to \infty$. This can be achieved by starting with an empty system, i.e., $M^N(0) = \mu(0) = [\alpha, 0, \ldots, 0]$ for every N, where α is the initial vector of the PH arrival process. Now we can define the $1 \times (C+2)h$ vectors $\mu(t)$ iteratively as

$$\mu(t+1) = \mu(t) R\left(\mu(t)\left(I_{C+2} \otimes e_h\right)\right),$$

for $t \geq 0$. Thus, due to [79, Theorem 4.1], for any $t \geq 0$, almost surely,

$$\lim_{N \to \infty} M^N(t) = \mu(t)\left(I_{C+2} \otimes e_h\right).$$

Therefore, to compute the mean field at time t it suffices to perform t matrix multiplications with matrices of size $(C+2)h$ only. As $h = 2$ often suffices to match up to three moments of the IAT distribution (see Appendix A.1), $(C+2)h$ will be fairly small, resulting in a fast computation of $\mu(t)$. Since our interest lies mainly in the steady state behavior (if it exists), we will iteratively compute $\mu(t)$ until $||\mu(t) - \mu(t-1)|| < \epsilon$, for small ϵ. The computation time can be reduced by selecting a different initial vector $\mu(0)$, that is closer to $\mu(t)$ for t large; e.g., while investigating excess traffic rates for various system loads, we could use the steady state of the previous load as an initial vector for the next case. Typical times to compute $\mu(t)$ for t large will be illustrated in Section 4.4.

With the model introduced in this section we are able to compute the vector $\mu(t)$ for any $t \geq 0$, which can be used to approximate $M^N(t)$ for N large. From this vector we can calculate the traffic matrix (inter-site rates) of the grid network. Given the rate r_{kj} of excess jobs of a particular class-k site processed by any of the $\gamma_j N$ class-j sites, the traffic rate from a single class-k site to a single class-j site equals $r_{kj}/(\gamma_j N)$. For the single class model, r_{11} is given by the last component of the occupancy measure $M^N(t)$, for t large enough. Recall that $M^N_{C+1}(t)$ represents the proportion of the sites that hold an excess job just prior to the redistribution step. As all the sites are identical, $M^N_{C+1}(t)$, for large t, is also the percentage of time in which a site has an excess job, i.e., r_{11}. This number can therefore be approximated by $\mu_{C+1}(t)$ for N large, which can be efficiently computed.

This concludes the analysis of the single class case, and the extension to multiple classes is the topic of the next section.

4.3 Mean field solution for multi-class Grids

When the sites differ in the number of servers or the arrival process, the model is a multiple class grid network. The single class grid network case can relatively easily be extended to the multi-class setting, because the Markov chain associated with each object in [79] is allowed to be reducible as explained below. The framework [79] applies to any system consisting of N objects, with N large, that are each characterized by a transition matrix $R^N(m)$, with m being the occupancy vector. This remains true if the state space of this transition matrix can be partitioned into K classes such that $R^N(m)$ can be written as a block diagonal matrix:

$$R^N(m) = \begin{bmatrix} R_1^N(m) & 0 & \dots & 0 \\ 0 & R_2^N(m) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & R_K^N(m) \end{bmatrix}.$$

Because no transitions are possible between states belonging to different classes, the state of an object which belongs to class k at time $t = 0$ will always remain in that class. Hence, we let $R_k^N(\cdot)$ characterize the transitions of a class-k site and define $M^N(0)$ for every N, the system state at $t = 0$, such that $\gamma_k N$ of the N sites start in a class-k state. Let

$$M^{N,(k)}(t) = \frac{1}{\gamma_k N}\left[a_0^{(k)}(t), \dots, a_C^{(k)}(t), a_{C+1}^{(k)}(t)\right]$$

be the occupancy measure of the class-k sites: $a_i^{(k)}(t)$ represents the proportion of class-k sites holding i jobs after step S2 ($0 \le i \le C^{(k)} + 1$). Then the overall occupancy $M^N(t)$ equals

$$M^N(t) = \left[\gamma_1 M^{N,(1)}(t), \dots, \gamma_K M^{N,(K)}(t)\right].$$

The mean field occupancy vector $\mu(t)$, for $t \ge 0$, is built analogously.

4.3.1 Computing $Q_k(m)$ and $\bar{Q}_k^N(m)$

To compute the mean field, we first need an expression for $R_k^N(m)$, the transition matrix of the class-k sites, given that the overall occupancy measure is $M^N(t) = m$. Since the arrivals and service completions are not affected by the presence of multiple classes we still have $R_k^N(m) = Q_k(m)S_kA_k$ and $\bar{R}_k^N(m) = \bar{Q}_k^N(m)S_kA_k$. The $Q_k(\cdot)$ and $\bar{Q}_k^N(\cdot)$ matrices have the same form as in (4.3), except that the expressions for $q_{i,i'}^{(k)}(\cdot)$ and $\bar{q}_{i,i'}^{(N,k)}(\cdot)$ require some modifications as all the sites in the network influence a class-k site and not just the other class-k sites. We consider these modifications in the following sections.

Random scheduling

For the *random* strategy case one finds

$$\bar{q}^{(N,k)}_{i,i+i'}(M^N(t)) = \frac{\binom{C^{(k)}-i}{i'}\binom{f(M^N(t))-(C^{(k)}-i)}{\sum_{k=1}^{K} a^{(k)}_{C^{(k)}+1}(t)-i'}}{\binom{f(M^N(t))}{\sum_{k=1}^{K} a^{(k)}_{C^{(k)}+1}(t)}},$$

with $f(M^N(t)) = \sum_{k=1}^{K}\sum_{s=1}^{C^{(k)}} s a^{(k)}_{C^{(k)}-s}(t)$. The difference for the multi-class case is that the available servers from all the site classes must be considered and the number of servers in each site is a class-dependent attribute. Apart from that, the expression is very similar to Equation (4.4).

Mostfree scheduling

To extend the mostfree strategy to the multi-class case we need a few definitions. Similar to Step S3 in Section 4.2, we start by defining $b^{(k)}_i(t)$ as the number of class-k sites with at most i busy servers after step S2 at time t; then $b^{(k)}_{C^{(k)}-i}(t)$ denotes the number of class-k sites with at least i free servers. Let $a^T(t) = \sum_{k=1}^{K} a^{(k)}_{C^{(k)}+1}(t)$ denote the total number of excess jobs after step S2. Finally, without loss of generality, label the K classes such that $C^{(1)} \geq C^{(2)} \geq \ldots \geq C^{(K)}$.

Provided that there are enough free servers at time t to support the excess jobs, we have $a^T(t) \leq \sum_{i=1}^{C^{(1)}}\sum_{k=1}^{K} b^{(k)}_{C^{(k)}-i}(t)$, where $b^{(k)}_i = 0$ for $i < 0$. Hence, for $a^T(t) > 0$, there exists a $0 < d \leq C^{(1)}$ such that

$$\sum_{i=d+1}^{C^{(1)}}\sum_{k=1}^{K} b^{(k)}_{C^{(k)}-i}(t) < a^T(t) \leq \sum_{i=d}^{C^{(1)}}\sum_{k=1}^{K} b^{(k)}_{C^{(k)}-i}(t),$$

which we denote as $d(M^N(t))$ (for $a^T = 0$ we set $d = C^{(1)}$). If we let $K = 1$ (single class), then $c(M^N(t))$ as defined in (4.1) equals $C^{(1)} - d(M^N(t))$. The value of $d(M^N(t))$ corresponds to the highest number of free servers found in any site after step S3. Thus, any class-k site has at least $C^{(k)} - d(M^N(t))$ busy servers after step S3. Hence, sites that had more than $d(M^N(t))$ free servers after step S2, receive one or more excess jobs such that exactly $d(M^N(t))$ or $d(M^N(t)) - 1$ free servers remain. Similar to (4.2), the fraction of these sites with $d(M^N(t))$ free servers after S3 is

$$\gamma_{d(M^N(t))} = \frac{\sum_{i=d(M^N(t))}^{C^{(1)}}\sum_{k=1}^{K} b^{(k)}_{C^{(k)}-i}(t) - a^T(t)}{\sum_{k=1}^{K} b^{(k)}_{C^{(k)}-d(M^N(t))}(t)}.$$

Notice, for the single-class case we have $\gamma_{d(M^N(t))} = \beta_{c(M^N(t))}$. If the number of free servers $\sum_{i=1}^{C^{(1)}}\sum_{k=1}^{K} b^{(k)}_{C^{(k)}-i}(t)$ is insufficient to support the $a^T(t)$ jobs, we let $d(M^N(t))$ be equal to zero, meaning that all the servers become occupied. This yields, for the class-k

sites, for $k = 1, \ldots, K$

$$q_{i,i'}^{(k)}(M^N(t)) = \begin{cases} 1 - \gamma_{d(M^N(t))}, & i < i' = C^{(k)} - d(M^N(t)) + 1 \leq C^{(k)}, \\ \gamma_{d(M^N(t))}, & i \leq i' = C^{(k)} - d(M^N(t)) < C^{(k)}, \\ 1, & i = i' > C^{(k)} - d(M^N(t)) \\ 1, & i' = C^{(k)} = C^{(k)} - d(M^N(t)), \end{cases}$$

for $0 \leq i, i' \leq C^{(k)}$. The third case indicates that no jobs are received when $i > C^{(k)} - d(M^N(t))$.

4.3.2 Computing the mean field

Given an occupancy vector m, for the *mostfree* case $R_k^N(m) = R_k(m)$, for all N, whereas for the *random* setting, the uniform limit $\bar{R}_k(m)$ is obtained in exactly the same manner as in the single class model (i.e., the hypergeometric probabilities converge to binomial probabilities). Due to [79, Theorem 4.1], we are able to compute the mean field as follows:

$$\mu^{(k)}(t+1) = \mu^{(k)}(t) R^{(k)}\left(\mu(t)\right),$$

for all k, where, as stated before, the vector $\mu(t)$ is defined as

$$\mu(t) = \left[\gamma_1 \mu^{(1)}(t)\big(I_{C^{(1)}+2} \otimes e_{h^{(1)}}\big), \ldots, \gamma_K \mu^{(K)}(t)\big(I_{C^{(K)}+2} \otimes e_{h^{(K)}}\big)\right].$$

We begin with an empty system and therefore $\mu^{(k)}(0) = (\alpha^{(k)}, 0, \ldots, 0)$, where the tuple $(\alpha^{(k)}, T^{(k)})$ characterizes the PH arrival process of a class-k site. And the class-k occupancy measure of a grid network with a large but finite number of sites can be approximated by means of the mean field due to the convergence result

$$\lim_{N \to \infty} M^{N,(k)}(t) = \mu^{(k)}(t)(I_{C^{(k)}+2} \otimes e_{h^{(k)}}).$$

4.3.3 Calculating the demand matrix D

In the previous sections, the mean field approach for the single class and multi-class cases was explained, allowing to calculate a mean field approximation of the occupancy measure $M^N(t)$. Recall that $M_i^{N,(k)}(t)$ represents the proportion of class-k sites holding i jobs after step S2 ($0 \leq i \leq C^{(k)} + 1$). Thus, the proportion of class-k sites with excess jobs equals $M_{C^{(k)}+1}^{N,(k)}(t)$, for t large. As all class-k sites are identical, $M_{C^{(k)}+1}^{N,(k)}(t)$ is also the percentage of time in which a class-k site has an excess job. Therefore it equals the excess rate of a class-k site. With $\lambda^{(k)}$ the mean job arrival rate at a class-k site, the rate of excess jobs processed by a class-k site, denoted as $\lambda_{exc}^{(k)}$, is found as the rate at which a class-k site completes jobs minus the rate of completed jobs that originated in this site; hence, for t large,

$$\lambda_{exc}^{(k)} = \mu^{(k)}(t) Q^{(k)}(M^{(k)}(t)) \begin{bmatrix} 0 \\ p^{(k)} \\ 2p^{(k)} \\ \vdots \\ C^{(k)}p^{(k)} \end{bmatrix} \otimes e_{h^{(k)}} - (\lambda^{(k)} - M_{C^{(k)}+1}^{(k)}(t)).$$

As the probability that an excess job receives service in a class-j site is independent of its type under the *mostfree* and *random* strategy, the rate $r_{k,j}$ of excess jobs of a class-k site served by any class-j site can be computed as

$$r_{k,j} = M^{(k)}_{C^{(k)}+1}(t)\frac{\lambda^{(j)}_{exc}}{\sum_{s=1}^{K}\lambda^{(s)}_{exc}} = \lambda^{(j)}_{exc}\frac{M^{(k)}_{C^{(k)}+1}(t)}{\sum_{s=1}^{K}M^{(s)}_{C^{(s)}+1}(t)},$$

for t large. From these inter-class rates, the demand matrix D can be easily calculated: the rate from a site s of class k to a site d of class j is $D_{s,d} = r_{k,j}/(\gamma_j N)$, where $\gamma_j N$ is the number of class-j sites.

4.4 Numerical results

In this section we first consider two simple Grids in order to analyze the effect of the scheduling algorithm on the performance of the Grid, in terms of the spill rates (recall that the spill rate $r_{k,j}$ is the rate at which excess jobs of a class-k site are sent to any class-j site). Next, we test the mean field model by considering a realistic European network scenario.

4.4.1 The effect of the scheduling algorithm

We first consider a Grid consisting of many sites partitioned in two classes. All the sites have 20 servers and the same arrival process, a Bernoulli process with mean IAT equal to 30 seconds. Class-2 sites represent only 1% of the total number of sites and their load, given by $\rho^{(2)} = \frac{\lambda^{(2)}}{\mu^{(2)} \cdot C^{(2)}}$, is equal to 0.95, i.e., they are heavily loaded. The remaining 99% of the sites are of class 1 and a load between 0.1 and 0.95 will be considered. When their load is equal to 0.95, all the sites in the Grid are identical. Figure 4.1 shows the total spill rate at class-2 sites, and the rate at which these spilled jobs are sent and processed at class-1 and class-2 sites. We observe that when the load of the class-1 sites is low, the *mostfree* algorithm allocates almost every excess job from a class-2 site to a class-1 site. This is the case for loads up to 0.7 in this scenario. On the other hand, the *random* policy assigns a significant fraction of excess jobs to the heavily-loaded class-2 sites. Although this has little influence on the total spill rate of the class-2 sites for low and mid loads, for loads above 0.75 the *mostfree* policy offers a reduction in the spill rate. In fact, the total spill rate under this policy can be up to 20% smaller than under the *random* scheduling. As expected, when both class-1 and class-2 sites have the same load, i.e., $\rho^{(1)} = 0.95$, the spill rates from class-2 sites toward sites of both classes are equal, and the *mostfree* policy still causes a significantly smaller spill rate than the *random* allocation.

We now consider a single-class Grid and compute the spill rate for different values of C, the number of servers per site. The results are included in Fig. 4.2, where the difference between these two policies becomes apparent at high loads. In this case we present the spill probability, which is the probability that a job is sent to a remote site. We find that the maximum reduction in spill probability caused by using the *mostfree* policy is around 15% for $C = 5$, near to 20% for $C = 20$ and above 22% for $C = 100$. Therefore we see an

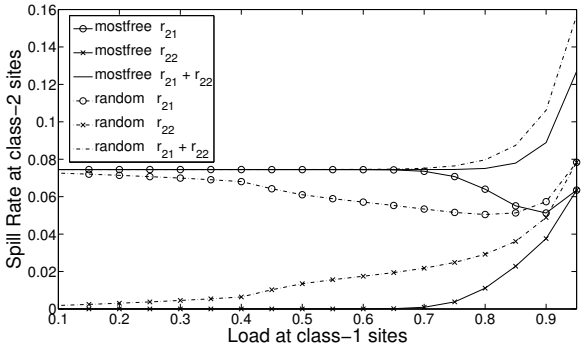


Figure 4.1: Mean field results for a two-class Grid, with variable load for class-1 sites.

increment in the maximum relative difference in spill rate as the number of servers per site increases. However, from Figure 4.2, we also observe that the load range for which the *mostfree* policy outperforms the *random* allocation decreases with the number of servers.

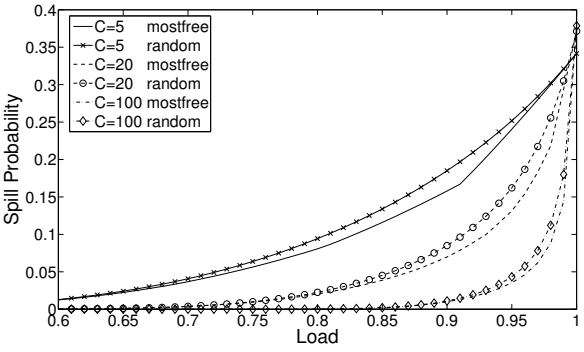


Figure 4.2: Mean field results for a single-class Grid, with variable number of servers

4.4.2 European Grid scenario

We now consider a realistic European Grid scenario, for which we need to extract the information required by the mean field model from real data. The preconditions to allow our mean field methodology are: (i) the job inter-arrival time (IAT) distribution should be modeled as a DPH distribution, (ii) the grid sites should be partitioned in a limited number of classes, and (iii) the number of Grid sites should be large enough. Conditions (ii)–(iii) are required because the mean field assumes an infinite number of sites per class. Hence, mean field results are expected to be closer to those of the finite system when the sites are partitioned into a few classes each with a significant number of sites. Condition (i) is not really limiting, since many real-world traces can be matched with a limited number of phases (keeping the analytical model compact) using moment-matching

procedures (see Appendix A.1).

With respect to (iii), realistic use cases for Grid dimensioning would comprise some tens to a couple of hundreds of sites. These numbers are still acceptable for the methodology to be practical as will be clear from the subsequent case study with $N = 100$ sites and $K = 5$ classes. With respect to the computation times, we found that the arrival process variability affects the number of iterations required for convergence of $\mu(t)$, while the overall load seems to have little effect. For this case study and with $\epsilon = 10^{-10}$, the computation times varied from one to ten minutes. These times can be further reduced, especially for the cases requiring more iterations, by initializing the system in the following manner: let $\pi_j^{(k)}$ be the stationary probability of having j busy servers in an $\mathrm{M}/\mathrm{M}/C^{(k)}/C^{(k)}$ queue, and let $\tau^{(k)}$ be the stationary distribution of the PH arrival process at station k, i.e., $\tau^{(k)} = \tau^{(k)}\left(T^{(k)} + \theta^{(k)}\alpha^{(k)}\right)$ and $\tau^{(k)}e_{h^{(k)}} = 1$. Then, by setting $\mu^{(k)}(0) = (\pi_0^{(k)}, \pi_1^{(k)}, \ldots, \pi_{C^{(k)}}^{(k)}) \otimes \tau^{(k)}$ we obtained a reduction of up to 80% in the number of iterations while the computation times decreased to less than four minutes. Since the $\pi_j^{(k)}$ probabilities can be found with closed-form expressions, their computation require very little time. Note that simulation running times in our case study differ by an order of magnitude, amounting to several hours (e.g. for the case study of Figure 4.3, simulating 10^7 time units took close to two hours).

The major limitation at first sight seems to be constraint (ii). However, looking at real world traces, many sites show similar behavior, which allows clustering the various sites into a limited number of classes. This can be achieved by the *K-means* clustering method [64], where each site is described by a set of V variables called descriptors. The main steps of the algorithm are:

(C1) Select K points in the V-dimensional space as centroids c_k $(k = 1, \ldots, K)$;

(C2) Form clusters C_k by assigning each site s to the closest centroid c_k;

(C3) Recalculate c_k as the V-dimensional mean over C_k;

(C4) If any c_k changed in C3, go back to C2.

We aim at characterizing each of the clusters with a DPH distribution matching the first three moments of the IAT distribution. Hence, we choose as site descriptors: the first non-central moment, the squared coefficient of variation (SCV) and the third normalized moment (n_3) of the IAT distribution. Let m_i be the i^{th} non-central moment, then define $\mathrm{SCV} = \frac{m_2}{m_1^2} - 1$ and $n_3 = \frac{m_3}{m_2 m_1}$. The reason to prefer SCV and n_3 rather than m_2 and m_3 is that they are not affected by the units in which the variables are measured. As the IAT distribution is based on real traces, we rely on the sample moments given by $\bar{m}_i = \frac{1}{S}\sum_{j=1}^{S} x_j^i$, where each x_j corresponds to one of the S samples.

For our case study, we used traces from a real-world EGEE/LCG Grid, deployed in Europe in the frame of the Large Hadron Collider (LHC) experiments at CERN in Geneva and the Enabling Grids for E-sciencE (EGEE) project [1]. We collected Grid-wide job arrival logs, recording the job arrival rate at 58 sites over a period of one month. After screening, we left out 8 sites because of lack of data to allow reliable statistical analysis. We used the clustering approach above, and partitioned the sites into $K = 5$ classes. To

characterize each site class, we used the average moments over the cluster's sites. For each class we used the method in [110] to match the first 3 moments of the job IATs with a DPH model with $h = 2$ phases (except for class 2, whose very small SCV causes matching for $h = 2$ to be restricted to the first 2 moments [110]). Table 4.1 summarizes the class descriptors. It is important to note that these characteristics greatly vary, ranging from low to high arrival rates and from small to large variability. To challenge the mean field method, we considered a case study with $N = 100$ Grid sites, respecting the proportion of each server class as observed in the EGEE/LCG trace.

Class	Mean IAT (s)	SCV	n_3	% Sites	C
1	29.75	136.45	2207.08	10%	150
2	77.24	83.40	488.69	46%	100
3	3696.46	0.46	5.73	6%	5
4	458.08	10.35	60.83	28%	10
5	1870.45	2.95	10.05	10%	10

Table 4.1: Characteristics of the 5 site clusters

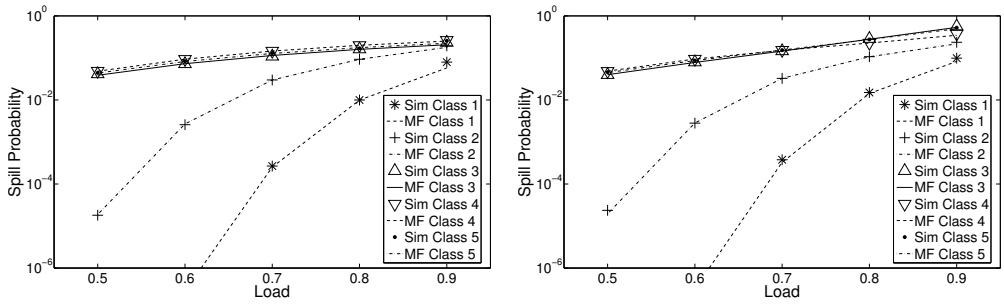
Varying the network load

Given the relevance that the load (ρ) of the network has on its performance we start by analyzing different values for this parameter. We set each class-k site's load to $\rho^{(k)} = \rho$, with $\rho^{(k)} = \frac{\lambda^{(k)}}{\mu^{(k)} \cdot C^{(k)}}$. Recall that, for a class-k site, $\lambda^{(k)}$ is the average job arrival rate, $\mu^{(k)}$ is the average job processing rate and $C^{(k)}$ is the number of servers. Given typical load values in network design, we studied the range $\rho \in [0.5, 0.9]$. We assumed $N = 100$ sites in total, comprising the $K = 5$ classes as outlined in Table 4.1. The number of servers and the IAT distribution at each site are also set as in this table. The average service time $1/p^{(k)}$ for each site of class k is set to obtain the target ρ, as $1/p^{(k)} = \rho C^{(k)} \, \mathrm{E}[\mathrm{IAT}^{(k)}]$, with $\mathrm{E}[\mathrm{IAT}^{(k)}] = 1/\lambda^{(k)}$ the average job IAT for a class-k site.

To evaluate the mean field methodology, we compared the results with the outcome of simulations. For this we implemented a discrete-event simulator and calculated the inter-site rates $D_{s,d}$ for each source site s and destination site d. To comprehensively present the results, the graphs will show for each class k the proportion of jobs sent to remote sites, i.e., the spill probability

$$P_{\mathrm{spill},k} = \left(\sum_{s \in \mathrm{class}k} \sum_{d \neq s} D_{s,d} \right) / \left(\sum_{s \in \mathrm{class}k} \sum_{d} D_{s,d} \right).$$

We compared the analytical results with simulations for both *random* and *mostfree* scheduling strategies. The graphs of Figure 4.3 show that in both cases the analytical and simulation results match very well. For the whole load range, the analytically calculated spill rates fall well within the 95% confidence interval (not shown on the graphs for the sake of clarity) of the simulations' spill rates (even though discrepancy increases for $\rho = 0.9$). Looking at the numerical values, we note that the discrepancy between analytical and



(a) *mostfree* scheduling (b) *random* scheduling

Figure 4.3: Simulation results match well with analytical mean field, for variable Grid resource load. Note that the curves for classes 3–5 overlap to great extent.

simulation results is largest for classes 1 and 2, but it is still less than the standard error of the simulation results (the standard error for a particular spill probability for class-k is given by $\text{stderr}^{(k)} = \sigma^{(k)}/(\gamma_k N)$ with $\sigma^{(k)}$ the variance of the spill probabilities for the $\gamma_k N$ class-k sites). This can be explained by the large SCV of the job IAT in these site classes (see Table 4.1). Note that, as expected, the *mostfree* strategy achieves lower spill probabilities than *random*, especially for high loads ($\rho > 0.7$).

Varying the job IAT variability

Having established the close match between analytical mean field and simulation results over a broad load range, we investigated the impact of the variability of the job inter-arrival times. Hence, we fix the load $\rho = 0.8$, but changed the SCV. For increasing variability on the job IATs, we expect higher $P_{\text{spill},k}$. Figure 4.4 shows that even for large SCV, the simulation results match the analytical results very well. As noted before, in terms of spill probability, *mostfree* outperforms *random* scheduling, but the amount seems dependent on the job IAT variability. As expected, the overall spill probability (over all jobs, regardless of the site class) increases with growing SCV.

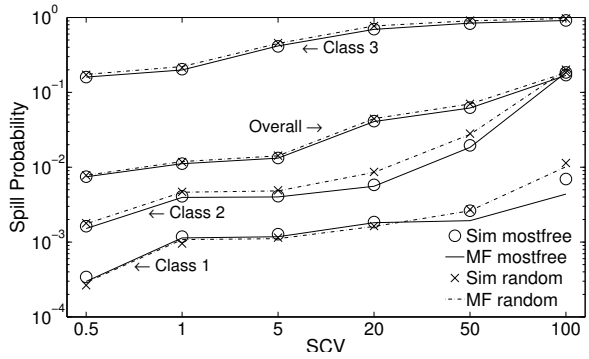


Figure 4.4: Comparison of *mostfree* and *random* scheduling for different SCV of the inter-arrival distribution. Results for classes 4 and 5 overlap with those of class 3.

Part III

Contention Resolution in Optical Switching

Contention Resolution in Optical Switching

In the backbone network, optical fibers have been able to provide the huge bandwidths required by the increasing Internet traffic. This ability has been partly propelled by the development of Wavelength Division Multiplexing (WDM), a technology that allows a single fiber to carry many signals simultaneously by using different wavelengths. Similarly, the switches in the network must also be able to handle the increasing amount of traffic, while avoiding information losses and delays. However, the use of electromagnetic switches to connect optical fibers introduces additional delays caused by the required opto-electronic translations. In contrast, with Optical Packet Switching (OPS) the switches can, in principle, fully process the packets in the optical domain, completely avoiding the mentioned translations. However, due to the lack of fast bit-level optical processing [105], a packet arriving at an OPS switch must be split into header and payload, and the header must be translated to and processed in the electromagnetic domain, while the payload waits in an optical buffer (see below). An alternative solution is called Optical Burst Switching (OBS) [96, 114], where the header travels in a separate channel and is sent to the network some time before the payload. The time frame between the header and the payload is called the offset time and its function is to allow the switching matrix to be set up before the actual arrival of the payload. Therefore, OBS reduces the need to buffer the payload at the entrance of the switch while it waits for the header to be processed and the switching matrix to be set up. This implies a reduction in the delay experienced by the payload at each switch in its path along the network. Therefore, OBS has become an attractive technology that has received much attention during the last years. A detailed description can be found in [94] and the references therein. Although there are many salient features to be analyzed in OBS, what is central for us is that this technology allows the payload to be transmitted in the optical domain and, therefore, contention arises in this domain when two or more bursts try to use the same wavelength in the same output port at the same time. Moreover, the available strategies for contention resolution in the optical domain truly differ from those in the electromagnetic domain. Our focus in this thesis is on the modeling and analysis of these contention resolution schemes, which we now describe in more detail.

One of the main differences between contention resolution in traditional and optical

switches results from the lack of optical random access memory. In fact, to delay a packet[1] in the optical domain one must rely on Fiber Delay Lines (FDLs). These are nothing more than a set of optical fibers that can be used to delay a packet for a time proportional to the length of each fiber. Specifically, we assume that an FDL buffer is made of N fibers, each holding W wavelengths. We also assume that time is slotted and each of these fibers offers a specific delay that is a multiple of the basic time slot. In addition, the set of possible delays provided by the FDLs increases linearly with granularity D, i.e., the first FDL provides a delay equal to D time slots, the second one offers a delay equal to $2D$, and so on, until the last FDL that delivers a delay of ND time slots. As can be seen, the optical buffer is not able to provide every required delay, but only multiples of the granularity D, a parameter that becomes relevant for buffer design. As a result, an FDL buffer creates gaps in the channel whenever a packet faces a delay that is not a multiple of D (in this work we do not consider gap filling strategies). In addition to the number N of FDLs and their granularity D, one should also take into account their location within the switch. There are two main alternatives: there could be a pool of FDLs per port; or the FDLs could be placed in a centralized location, such that they can be used to buffer packets from any port. While sharing the FDLs among all the ports may reduce the total number of fibers required to provide a predefined quality of service, it also implies a more complex switching matrix.

In addition to the FDLs, there is another contention resolution scheme that will be of interest for us: wavelength conversion. A wavelength converter allows a packet to be forwarded via a different wavelength than the one it used to enter the switch. There are two main characteristics to consider when modeling a switch with a pool of converters. The first is the number of converters compared to the number of wavelengths. One could opt for having as many converters as input wavelengths, i.e., if there are K input ports and W wavelengths per port then one would need KW converters. This setup is known as *full* wavelength conversion since in this case there will always be a converter available if an incoming packet needs to be translated. Let C be the number of converters and define the conversion ratio σ as the ratio between the number of converters and the total number of input wavelengths, i.e., $\sigma = \frac{C}{KW}$. Then full wavelength conversion corresponds to $\sigma = 1$. In the opposite case, called null wavelength conversion, there are simply no converters ($\sigma = 0$). In this case each wavelength works on its own and the analysis of the system is reduced to that of a single wavelength. The intermediate case ($0 < \sigma < 1$) is referred to as *partial* conversion and is the most interesting one since it displays the benefits of wavelength conversion without requiring the costly installation of KW converters. The second characteristic to consider is the conversion *range* provided by the converters. A converter is said to provide *full-range* conversion if it is able to translate a signal from any incoming wavelength to any other. In contrast, a *limited-range* converter is able to translate a signal only from or to a subset of the W wavelengths [127]. Although a full-range converter is certainly more flexible, it is more expensive than a limited-range converter and performs slower when trying to convert over a wide range of wavelengths [107]. Additionally, as with the FDLs, the converters could be located in a centralized pool or split among the ports in a non-centralized arrangement. We must

[1]The terms packet and burst will be used interchangeably from here onward.

mention that there is another contention resolution strategy known as deflection routing, which makes use of the whole network to resolve contention, but it is known to have a poor performance at high loads [125, 126]. Moreover, our focus is on the analysis of a single switch, and therefore we will not consider this strategy any further.

In the last few years many efforts have been made toward the analysis of these contention resolution schemes. The effect of full-range wavelength conversion on a *bufferless* switch was studied in [2, 42, 124]. As the number of wavelengths that can be carried by a single fiber has increased significantly, the case of switches with a large number of wavelengths per port has also received attention. This case is treated in [123], where a queueing network model is proposed for a bufferless switch with non-centralized full wavelength conversion. On the other hand, the analysis of a switch equipped with FDLs, but without converters (or equivalently, a single-wavelength switch) was considered in [26, 71–73, 103, 116]. Introducing FDLs in a multi-wavelength switch drastically increases the complexity of the system and its analysis. This results from combining the multidimensional nature of a multi-wavelength switch with the special queuing behavior of the optical buffer. Hence, simulation models have been used to analyze the interaction of both solutions [27, 28, 45]. Also, in [104] an approximation based on an analytical model with a Round-Robin discipline, that relies on the solution of a single wavelength system, is introduced to analyze a multi-wavelength switch. It is shown to work reasonably well only for a fixed packet size when the minimum horizon allocation policy (see Chapter 6) is used. On the other hand, limited-range wavelength conversion implies a more intricate interaction between adjacent wavelengths. This topic has been considered in [3, 4, 41, 95, 128] for the bufferless case relying on (approximate) analytical models. For instance, for exponential packet sizes, the lack of buffers allows an exact solution for the non-centralized full-range conversion case [2]. This result is used in [3] to approximate the performance of a limited-range system.

Clearly, both the switch's design parameters, such as the conversion ratio or the number of FDLs, and the traffic characteristics, such as the load or the burstiness, are likely to have an important effect on the performance of the switch. In the next chapters we propose analytical models that allow us to evaluate these effects for three different switch architectures, including features that have not been analyzed previously. Chapter 5 considers a bufferless switch with a *centralized* pool of full-range converters, where the model allows a general packet-size distribution and a general arrival process. Moreover, the model is able to represent heterogeneous traffic patterns, that is, the traffic characteristics may differ for each output port. This model belongs to the class of mean field models (see Appendix A.3), which provides exact results when the number of wavelengths tends to infinite. We will show that this model provides a good approximation to the performance of a switch with a large (finite) number of wavelengths, which is a relevant case given the advances in WDM technology. Next, Chapter 6 focuses on a switch endowed with a pool of both full-range converters and FDLs *per output port*. In this case we also introduce a mean field model that becomes exact when the number of wavelengths tends to infinite, and provides good approximations for a switch with a large number of these. Moreover, this model is able to consider the effect of combining the conversion and buffering solutions to resolve contention. We must highlight that in both cases, the mean

field models provide an efficient way to analyze the effect of the traffic characteristics and the design parameters on the switch performance. This is in contrast with simulations that would require very long computation times to estimate a very small packet loss probability, especially when the number of wavelengths is large. Finally, Chapter 7 analyzes a switch where *each port* has a pool of *limited-range* converters and FDLs. To this end we introduce a finite MC to model the interaction of two wavelengths under two different allocation policies. This model allows the packet-size and the IATs to follow general distributions, even though we make use of the numerical tractability gained by assuming PH-distributed IATs. Although limited, this model allows us to measure the impact of the wavelength allocation policies when a packet in an input wavelength can only be converted to one of the neighboring wavelengths. Moreover, based on the results of this model, we propose an approximation method for the case where a packet can be converted to any of the two adjacent wavelengths. We show that this method works well for a wide range of parameter values, and is able to provide results in a fraction of the time required by simulations. All in all, the models to be introduced in the next chapters provide means to efficiently evaluate the effect of the contention resolution strategies under general traffic conditions.

Chapter 5

Centralized Partial Conversion

In this chapter we introduce a mean field model for an optical switch with full-range centralized partial wavelength conversion. The main feature of the mean field model (see Appendix A.3) is that it is exact when the number of wavelengths is infinite, and it can be used to approximate the performance of a switch with a large number of wavelengths. Modeling a switch with multiple ports and many wavelengths per port presents several difficulties, particularly with regard to the multidimensional nature of the model, as it requires keeping track of the state of the wavelengths in each port and, additionally, the converters in the centralized pool. These dimensionality problems caused by the number of wavelengths are avoided by using the mean field model, and in fact the model becomes more accurate as the number of wavelengths increases. The switch is assumed to work in a synchronous manner, where the time is divided in equally-spaced slots. At each wavelength packets arrive according to a Markovian arrival process, and their size follows a general distribution with finite support. These traffic characteristics may be different for each output port, a scenario that we refer to as heterogeneous traffic. Moreover, the non-centralized architecture can be analyzed as a special case within our model. The assumption of a general packet-size distribution, instead of fixed size, has the advantage of reducing header processing and being more suitable for IP traffic [27]. The model also provides insight into the effect of the traffic parameters on the packet loss probability, which is considered the main performance measure. In particular, we have found that, if the arrival process is Bernoulli, the loss probability is affected by the packet-size distribution only through its mean. This is no longer the case if the arrivals follow a more general Markovian process, although we have found experimentally that even in this case the loss probability is hardly sensitive to the packet-size distribution. In contrast, the burstiness of the arrival process appears to have an important effect on the loss probability, specially for mid loads. Also, under Bernoulli arrivals we provide a closed expression for the minimum conversion ratio required to attain zero losses when the number of wavelengths tends to infinity, denoted as σ^*. This minimum conversion ratio is shown to depend on the (squared) load and the mean packet size. For Markovian arrivals we are able to compute this ratio with a single run of the mean field model. Also, we have found that when the number of wavelengths is large and the traffic among the ports

is homogeneous, there is no difference between the performance of the centralized and non-centralized architectures. However, under heterogeneous traffic conditions (packet-size distribution and burstiness), important gains in conversion resources can be obtained by using a centralized architecture, even for a large number of wavelengths.

This chapter is organized as follows. Section 5.1 describes in detail the architecture of the switch, while Section 5.2 introduces the mean field model under some simplifying assumptions. The general version of the model is discussed in Section 5.3. The final section is concerned with numerical results that illustrate the behavior of the mean field model and analyze the effect of various traffic parameters on the switch performance.

5.1 The switch architecture

This section describes the architecture and operation of the optical switch under analysis. The switch, shown in Figure 5.1, consists of K input/output ports, each connecting to a fiber carrying W wavelengths, and a centralized pool of C converters. An incoming packet will attempt transmission through an specific output port using the same wavelength in which it entered the switch. If that wavelength is busy the packet will be converted to an available wavelength in the same output port. This conversion is performed by an idle converter in the shared pool. If all the converters or all the other wavelengths in the same output port are busy, the packet must be dropped. The probability that a packet is dropped is considered the main performance measure of the switch. With regard to the converters, we assume that they provide full-range conversion and their number is defined as a proportion of the total number of output wavelengths, i.e., $C = \sigma KW$, where σ is the conversion ratio. When $\sigma = 1$ the system is said to provide *full* conversion. As stated before, an economically-feasible solution for optical switching including converters should limit the use of these devices while guaranteeing a minimal packet loss probability. Therefore our focus here is on the *partial* conversion case, where $0 < \sigma < 1$.

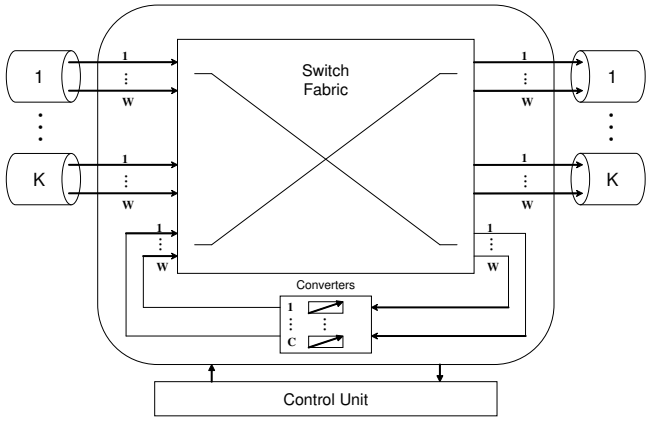


Figure 5.1: Optical switch with K input/output ports, W wavelengths and shared C converters

The switch has a synchronous operation with equally-spaced time slots and the state of the switch is observed at slot boundaries. This type of operation implies a simpler switching matrix compared to the asynchronous case, but it requires packet synchronization and alignment [2, 28]. Each wavelength in output port k has its own arrival process, modeled as a Discrete Markovian Arrival Process (DMAP) [76] characterized by the $m_k \times m_k$ matrices $D_0^{(k)}$ and $D_1^{(k)}$, for $k = 1, \ldots, K$. The entries of the matrices $D_0^{(k)}$ and $D_1^{(k)}$ hold the transition probabilities of the underlying chain associated with zero and one arrivals, respectively (see Appendix A.1). This kind of arrival process is computationally tractable as well as versatile, including widely used arrival processes, such as the Bernoulli process and the discrete-time versions of the interrupted Poisson process (IPP) and the Markov modulated Poisson process (MMPP), as special cases. In particular, it has been previously used to model the behavior of a bufferless asynchronous optical switch with non-centralized converters [2,95,124]. The size of the packets directed to output port k follow a general distribution with finite support Ξ_k. Let $r_i^{(k)}$ be the probability that a packet for output port k is of size i, for $i \in \Xi_k$, and let $L_{\max}^{(k)}$ be the maximum packet size. Therefore, both the arrival process and the packet-size distribution may differ among output ports.

To describe the state of a wavelength or a converter we employ the scheduling horizon. This is defined as the time (number of slots) required by the wavelength/converter to transmit/translate the packet it is currently busy with. Therefore, if a wavelength has a horizon equal to 0 (i.e., it is available) and a new packet of size j arrives, the wavelength will accept the packet and update its horizon to j. Then, the horizon will be reduced by one at every slot until it reaches zero again. During the time the wavelength has a horizon greater than zero, new packets may attempt transmission using the same wavelength. If this occurs, those packets must be converted to another wavelength using an available converter, i.e., a converter with horizon equal to zero. If there is both an idle wavelength in the same output port and an available converter, these two resources are seized by the packet and their horizons are both updated from zero to the value of the packet size. The model to be introduced in the next section relies on this description to represent the state of the switch.

5.2 The Basic Model

In this section we introduce a simplified version of the model for the optical switch. We consider the special case where all the output ports have the same inter-arrival time (IAT) and packet-size distributions (the subscript k will be removed from the parameters of these distributions). Also, we assume that the IATs follow a geometric distribution with parameter p, and therefore the arrival process at each wavelength is a Bernoulli process. We start by describing the model for a finite number of wavelengths and then consider the limit when this number tends to infinity. Finally, we describe some results concerning a fixed point of the model.

In this model, each of the KW output wavelengths in the switch has its own Bernoulli arrival process with parameter p given by $p = \frac{\rho}{E[L]}$, where ρ is the load of the wavelength and L is the random variable describing the packet-size, with expected value $E[L]$. Each

wavelength and converter will be treated as a separate object in a system of $N = KW + C$ interacting objects. To describe their evolution we observe the system at slot boundaries and consider three main steps within a slot: transmission, arrival and reallocation. During packet transmission each wavelength (resp. converter) holding a packet transmits (resp. translates) a part of it proportional to the slot length. After transmission, each wavelength may receive a new arrival with probability p. Those packets that arrive at a busy wavelength form the set of extra-packets that must be reallocated among the idle wavelengths in the same port. While packet arrival and transmission are independent operations for each wavelength in each port, the reallocation depends on the availability of wavelengths in the same output port and converters in the shared converter pool.

To describe the evolution of the wavelengths and the converters during a time slot we consider the three steps separately and associate a transition matrix with each of them: transmission (S_l), arrivals (A_l) and reallocation (Q_l). The subscript l is equal to w or c if the matrix is associated to the evolution of a wavelength or a converter, respectively. Recall that all the output ports have the same traffic pattern (IAT and packet-size distribution) and therefore the matrices S_w and A_w describe the evolution of a wavelength in *any* of the ports. On the other hand, the evolution of a wavelength in port k during the reallocation step depends on the state of all the wavelengths in this port. Therefore, for this step we add a superscript k to the transition matrix (Q_w^k) of a wavelength in output port k. We now define these matrices explicitly and show how they are combined to describe the whole system.

S1 - Transmission: Before transmission each wavelength/converter has a horizon between 0 and $L_{\max}$. During the transmission step (S1) the horizon of each wavelength and each converter is reduced by one. Therefore the transition matrices for wavelengths and converters are given by $S_w = S_c = T_{L_{\max}}$, where T_n is the $(n+1) \times n$ matrix with entries

$$[T_n]_{ij} = \begin{cases} 1, & i = j = 0, \\ 1, & j = i - 1, i = 1, \ldots, n, \\ 0, & \text{otherwise.} \end{cases}$$

Notice that we label the rows and columns of an $m \times n$ matrix from 0 to $m-1$ and from 0 to $n-1$, respectively.

S2 - Arrivals: When a packet arrives at an idle wavelength (horizon equal to 0) it is accepted for transmission in the same wavelength. Therefore, the first row of the matrix A_w is given by

$$[A_w]_{0j} = \begin{cases} 1 - p, & j = 0, \\ pr_j, & j = 1, \ldots, L_{\max}. \end{cases}$$

If the wavelength is busy (horizon equal to $i > 0$) and there is no arrival, the horizon stays unaltered. Otherwise, the new state is $L_{\max} + i$, showing that the wavelength has scheduling horizon equal to i and holds an extra-packet for reallocation. Hence, for $i = 1, \ldots, L_{\max} - 1$,

$$[A_w]_{ij} = \begin{cases} 1 - p, & j = i, \\ p, & j = L_{\max} + i. \end{cases}$$

Therefore, the size of the matrix A_w is $L_{\max} \times 2L_{\max}$. As the arrivals do not affect the state of the converters, the matrix A_c is equal to the identity matrix $I_{L_{\max}}$.

S3 - Reallocation: The last step consists of the reallocation of the extra-packets in the available wavelengths. As mentioned before, to determine the evolution of a single wavelength or converter during this step we need to know the state of the whole system. That is, the transition matrices Q_w^k and Q_c at time t depend on the state of the system at that time. The state of the system with N objects at time t (after S2) is described by the occupancy vector $M^N(t)$, which will be specified below. Let $q_i^{(w,k)}(M^N(t))$ be the probability that an idle wavelength belonging to output port k receives an extra-packet of size i during reallocation at time t, for $i = 1, \ldots, L_{\max}$, and $q_0^{(w,k)}(M^N(t))$ the probability that it remains idle. Since any extra-packet is either converted to an available wavelength or dropped, it is enough to use the scheduling horizon to describe the state of a wavelength *after* this step. Therefore, the $2L_{\max} \times L_{\max}$ matrix $Q_w^k(M^N(t))$ is given by

$$Q_w^k(M^N(t)) = \tag{5.1}$$

$$\left[\begin{array}{c:c:c:c:c:c}
q_0^{(w,k)}(M^N(t)) & q_1^{(w,k)}(M^N(t)) & q_2^{(w,k)}(M^N(t)) & \ldots & q_{L_{\max}-1}^{(w,k)}(M^N(t)) & q_{L_{\max}}^{(w,k)}(M^N(t)) \\
\hdashline
0 & 1 & 0 & \ldots & 0 & 0 \\
0 & 0 & 1 & \ldots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \ldots & 1 & 0 \\
0 & 0 & 0 & \ldots & 0 & 1 \\
\hdashline
0 & 1 & 0 & \ldots & 0 & 0 \\
0 & 0 & 1 & \ldots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \ldots & 1 & 0
\end{array}\right].$$

Likewise, let $q_j^c(M^N(t))$ be the probability that an idle converter receives an extra-packet of size j for conversion during reallocation at time t, for $j = 1, \ldots, L_{\max}$. Then the $L_{\max} \times (L_{\max} + 1)$ matrix $Q_c(M^N(t))$ has a first row given by

$$[Q_c(M^N(t))]_{0j} = q_j^c(M^N(t)), \quad j = 0, 1, \ldots, L_{\max},$$

where $q_0^c(t)$ is the probability that an idle converter remains idle after S3. The rest of the converters keep the same horizon, hence $[Q_c(M^N(t))]_{ii} = 1$, for $i = 1, \ldots, L_{\max} - 1$. Recall that before this step the maximum horizon a converter can have is $L_{\max} - 1$.

To define the value of $q_j^{(w,k)}(M^N(t))$ and $q_j^c(M^N(t))$ we need to specify $M^N(t)$, the occupancy vector that describes the state of the whole system with N objects before S3. The entries of this vector hold the fraction of the $N = KW + C$ objects that are in each possible state. For port k, let $W_k^N(t)$ be the $1 \times 2L_{\max}$ vector with entries $w_i^k(t)$. For $i = 0, \ldots, L_{\max}$, $w_i^k(t)$ holds the number of wavelengths in port k with horizon equal to i after S2, holding *zero* extra-packets. For $i = L_{\max} + j$ and $j = 1, \ldots, L_{\max} - 1$, $w_i^k(t)$ holds the number of wavelengths in port k with horizon equal to j after S1 that received an extra-packet during S2. Similarly, let $C^N(t)$ be the $1 \times L_{\max}$ vector with

entries $c_i(t)$, which hold the number of converters with horizon equal to i after S2, for $i = 0, 1, \ldots, L_{\max} - 1$. Before using the vectors $W_k^N(t)$ and $C^N(t)$ to construct $M^N(t)$ we need to normalize them by the total number of objects $KW + C$. The fraction of objects that are wavelengths in port k is $\frac{W}{KW+C} = \frac{1}{K(1+\sigma)}$, for $k = 1, \ldots, K$. Similarly, the fraction of objects that are converters is $\frac{C}{KW+C} = \frac{\sigma}{(1+\sigma)}$. We can now define the state vector $M^N(t)$ as

$$M^N(t) = \frac{1}{1+\sigma}\left[\frac{1}{K}W_1^N(t), \ldots, \frac{1}{K}W_K^N(t), \sigma C^N(t)\right],$$

which holds the proportion of objects (wavelengths and converters) in each of the $(2K + 1)L_{\max}$ possible states.

To determine the evolution of the idle wavelengths and converters, three quantities are relevant: the number of available converters $c_0(t)$; the number of available wavelengths $w_0^k(t)$ in output port k; and the number of extra-packets $d_k(M^N(t))$ in output port k, which is given by

$$d_k(M^N(t)) = \sum_{j=1}^{L_{\max}-1} w_{L_{\max}+j}^k(t).$$

Since at most $w_0^k(t)$ extra-packets can be received at output port k after conversion, the number of extra-packets from this port that are sent to the centralized pool is $f_k(M^N(t)) = \min\{w_0^k(t), d_k(M^N(t))\}$, and the total number of extra-packets sent for conversion is

$$f(M^N(t)) = \sum_{k=1}^{K} f_k(M^N(t)).$$

Therefore, the total number of extra-packets that is actually converted is $g(M^N(t)) = \min\{c_0(t), f(M^N(t))\}$. Since there are no priorities among the ports, the probability that a converter that will be used by an extra-packet is assigned to an extra-packet from output port k is $f_k(M^N(t))/f(M^N(t))$. Therefore the average number of extra-packets of output port k that are converted is

$$g_k(M^N(t)) = \frac{f_k(M^N(t))}{f(M^N(t))} g(M^N(t)).$$

Now we can determine the time-dependent transition probabilities for an idle wavelength in output port k during S3,

$$q_j^{(w,k)}(M^N(t)) = \begin{cases} 1 - \frac{g_k(M^N(t))}{w_0^k(t)}, & j = 0, \\ \frac{g_k(M^N(t))}{w_0^k(t)} r_j & j = 1, \ldots, L_{\max}. \end{cases} \tag{5.2}$$

In a similar manner we can define the transition probabilities for an idle converter in this step,

$$q_j^c(M^N(t)) = \begin{cases} 1 - \frac{g(M^N(t))}{c_0(t)}, & j = 0, \\ \frac{g(M^N(t))}{c_0(t)} r_j, & j = 1, \ldots, L_{\max}. \end{cases} \tag{5.3}$$

The factor r_j comes from the fact that the $g(M^N(t))$ converted extra-packets are selected randomly among the set of extra-packets, therefore keeping the same packet-size distribution. Notice that $q_j^{(w,k)}(\cdot)$ and $q_j^c(\cdot)$ are actually independent of N as can be checked by dividing the quantities involved in their computation by N. This concludes the definition of the transition matrices related to the reallocation step. We now turn to the combination of the three steps to describe the evolution of the whole system.

5.2.1 Combining S1, S2 and S3 - Mean Field

By observing the system after S2, we define the transition matrix at time t for a single wavelength in output port k as $R^N_{(w,k)}(M^N(t)) = Q^k_w(M^N(t))S_wA_w$, and for a single converter as $R^N_c(M^N(t)) = Q_c(M^N(t))S_cA_c$. These $K+1$ matrices can be combined into a single matrix describing the evolution of a single object at time t,

$$R^N(M^N(t)) = \begin{bmatrix} R^N_{(w,1)}(M^N(t)) & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & R^N_{(w,K)}(M^N(t)) & 0 \\ 0 & \cdots & 0 & R^N_c(M^N(t)) \end{bmatrix}.$$

This system, composed of wavelengths and converters, can be analyzed under the framework introduced in [79] for a general system of interacting objects. When the number of objects tends to infinity and under some mild conditions, such a system is shown to converge to its mean field [79], which is a time-dependent deterministic model (see Appendix A.3). In our case, the objects are of $K+1$ different classes, their status at time t is contained in the vector $M^N(t)$ and the evolution of the system is described by the matrix $R^N(\vec{m})$, where $\vec{m}$ is a $1 \times (2K+1)L_{\max}$ occupancy vector. There are two conditions for the system to tend to its mean field when the number of objects tends to infinity: first, the entries of the transition matrix $[R^N(\vec{m})]_{ij}$ must converge uniformly to some $[R(\vec{m})]_{ij}$ on the set of all occupancy vectors $\vec{m}$ when $N \to \infty$. Second, these entries must be continuous in $\vec{m}$. In our model both conditions hold as can be seen from the definition of the matrices $Q^k_w(\cdot)$ and $Q_c(\cdot)$. In fact, and as already mentioned, these matrices are independent of the number of objects as becomes clear if the quantities involved in the computation of $q_j^{(w,k)}(\cdot)$ and $q_j^c(\cdot)$ are divided by N.

To comply with the conditions in [79] the initial state of the system $M^N(0)$ must also converge uniformly to some $M(0)$. Let $e_1^{L_{\max}}$ be the $1 \times L_{\max}$ vector with 1 in the first position and zero everywhere else. Let the initial state of the system be given by

$$C^N(0) = e_1^{L_{\max}}A_c \text{ and } W_k^N(0) = e_1^{L_{\max}}A_w,\ k = 1, \ldots, K,$$

which corresponds to an empty system and is independent of N. Let the state of the mean field at time $t \geq 0$ be described by the vector

$$\mu(t) = \frac{1}{1+\sigma}\left[\frac{1}{K}\mu^{w,1}(t), \ldots, \frac{1}{K}\mu^{w,K}(t),\ \sigma\mu^c(t)\right],$$

with initial state $\mu(0) = M^N(0)$, and let its evolution be given by $\mu(t+1) = \mu(t)R(\mu(t))$. Then, by [79, Theorem 4.1], for any fixed time t, almost surely,

$$\lim_{N\to\infty} M^N(t) = \mu(t), \quad t \geq 0.$$

Hence, we can approximate the behavior of a switch with a large number of wavelengths (objects) by means of the mean field. However, this result is time-dependent and says nothing about the behavior of the system when t tends to infinity. This is the topic of the next section.

5.2.2 Combining S1, S2 and S3 - Fixed Point

We are now left with a deterministic system with initial state $\mu(0)$ and time-dependent transition matrix $K(\mu(t))$. To study the behavior of this system as a function of time we change the observation time-points. We now observe the system after S1, i.e., after transmission and just before arrivals. Let $\alpha(t)$ be the occupancy vector describing the state of all the wavelengths and converters at time t just after S1 and before S2. Similar to $\mu(t)$, the vector $\alpha(t)$ can be partitioned as

$$\alpha(t) = \frac{1}{1+\sigma}\left[\frac{1}{K}\alpha^{w,1}(t), \ldots, \frac{1}{K}\alpha^{w,K}(t),\, \sigma\alpha^{c}(t)\right],$$

where $\alpha^{w,k}(t)$ (resp. $\alpha^{c}(t)$) is an occupancy vector describing the state of the wavelengths in output port k (resp. converters in the centralized pool). Since $\alpha(t)$ and $\mu(t)$ describe the state of the switch before and after S2, respectively, $\mu(t)$ can be obtained from $\alpha(t)$ as

$$\mu(t) = \alpha(t)\begin{bmatrix} I_K \otimes A_w & 0 \\ 0 & A_c \end{bmatrix},$$

where $\otimes$ is the Kronecker product. Now let the matrix $P_w^k(\alpha(t))$ describe the transition probabilities of the wavelengths in output port k at time t, observing the system after S1. This is given by $P_w^k(\alpha(t)) = A_w Q_w^k(\mu(t))S_w$, where, as stated above, $\alpha(t)$ completely determines $\mu(t)$. A similar matrix $P_c(\alpha(t))$ can be specified for the evolution of the converters observed just after S1.

The main advantage of defining the $L_{\max} \times L_{\max}$ matrices $P_w^k(\alpha(t))$ and $P_c(\alpha(t))$ is that they can be expressed as

$$P_l^k(\alpha(t)) = \begin{bmatrix} p_0^{(l,k)}(\alpha(t)) & p_1^{(l,k)}(\alpha(t)) & \ldots & p_{L_{\max}-2}^{(l,k)}(\alpha(t)) & p_{L_{\max}-1}^{(l,k)}(\alpha(t)) \\ 1 & 0 & \ldots & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & 0 \end{bmatrix}, \tag{5.4}$$

for $l \in \{w, c\}$ and $k = 1, \ldots, K$ (if $l = c$ the sub/superscript k is obviously removed). The specific values of the first-row entries, in the case of the matrices associated to the evolution of the wavelengths, are given by

$$p_j^{(w,k)}(\alpha(t)) = \begin{cases} (1-p)(q_0^{(w,k)}(\mu(t)) + q_1^{(w,k)}(\mu(t))) + pr_1, & j = 0, \\ (1-p)q_{j+1}^{(w,k)}(\mu(t)) + pr_{j+1}, & j = 1, \ldots, L_{\max}-1. \end{cases} \tag{5.5}$$

For the converters those values are

$$p_j^c(\alpha(t)) = \begin{cases} q_0^c(\mu(t)) + q_1^c(\mu(t)), & j = 0, \\ q_{j+1}^c(\mu(t)), & j = 1, \ldots, L_{\max} - 1. \end{cases} \tag{5.6}$$

These $K+1$ matrices can be recombined into a single matrix $P(\alpha(t))$ with $K+1$ irreducible classes,

$$P(\alpha(t)) = \begin{bmatrix} P_w^1(\alpha(t)) & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & P_w^K(\alpha(t)) & 0 \\ 0 & \cdots & 0 & P_c(\alpha(t)) \end{bmatrix}.$$

Let $\mathcal{M}_{(K+1)L_{\max}}$ be the set of all occupancy vectors of size $(K+1)L_{\max}$. The matrix $P(\alpha)$ defines a mapping from $\mathcal{M}_{(K+1)L_{\max}}$ into $\mathcal{M}_{(K+1)L_{\max}}$, which is said to have a fixed point α if there exists a vector $\alpha \in \mathcal{M}_{(K+1)L_{\max}}$ such that $\alpha P(\alpha) = \alpha$. To find this fixed point we set-up the system of equations $\alpha P(\alpha) = \alpha$ and solve it for α. To do so, we first need an explicit definition of the (first-row) entries of $P_w^k(\alpha)$ and $P_c(\alpha)$ in terms of α. Let the first-row entries of these matrices, as defined in Equation (5.4), be $p_j^{(w,k)} = p_j^{(w,k)}(\alpha)$ and $p_j^c = p_j^c(\alpha)$, for $j = 0, \ldots, L_{\max} - 1$. These probabilities depend on the value of $q_j^{(w,k)} = q_j^{(w,k)}(\mu)$ and $q_j^c = q_j^c(\mu)$, for $j = 0, \ldots, L_{\max} - 1$, as in equations (5.5) and (5.6). The vector μ is given by

$$\mu = \alpha \begin{bmatrix} I_K \otimes A_w & 0 \\ 0 & A_c \end{bmatrix},$$

and therefore $\mu_0^{(w,k)} = \alpha_0^{(w,k)}(1-p)$ and $\mu_0^c = \alpha_0^c$. This equation relates the proportion of idle wavelengths and converters before and after arrivals in the fixed point. Similarly, the proportion of wavelengths in output port k holding an extra-packet after S2 in the fixed point is given by $\delta_k = (1 - \alpha_0^{(w,k)})p$.

Now let ϕ_k be the number of extra-packets in port k sent for conversion in the fixed point, as a proportion of the total number of objects, which is given by

$$\phi_k = \frac{1}{K(1+\sigma)} \min\left\{\mu_0^{(w,k)}, \delta_k\right\} = \frac{1}{K(1+\sigma)} \min\left\{\alpha_0^{(w,k)}(1-p), (1-\alpha_0^{(w,k)})p\right\},$$

and thus only depends on the vector α. Therefore, the number of converted extra-packets in the fixed point, as a proportion of the total number of objects, is given by

$$\gamma = \min\left\{\sum_{k=1}^{K} \phi_k, \frac{\sigma}{1+\sigma}\mu_0^c\right\} = \min\left\{\sum_{k=1}^{K} \phi_k, \frac{\sigma}{1+\sigma}\alpha_0^c\right\}, \tag{5.7}$$

which also depends on α alone. Additionally, let γ_k be the number of converted extra-packets from output port k as a fraction of the total number of objects, given by

$$\gamma_k = \frac{\phi_k}{\sum_{j=1}^{K} \phi_j}\,\gamma, \tag{5.8}$$

where the first term on the right-hand side is the probability that an extra-packet sent for conversion is actually an extra-packet from output port k.

Since γ and γ_k depend only on $\alpha_0^{(w,k)}$ and α_0^c, we can use them in Equations (5.2) and (5.3) to determine the transition probabilities for the idle wavelengths in the fixed point in terms of α as

$$q_j^{(w,k)} = \begin{cases} 1 - \frac{K(1+\sigma)\gamma_k}{\mu_0^{(w,k)}} & = 1 - \frac{K(1+\sigma)\gamma_k}{\alpha_0^{(w,k)}(1-p)}, \quad j = 0, \\ \frac{K(1+\sigma)\gamma_k}{\mu_0^{(w,k)}} r_j & = \frac{K(1+\sigma)\gamma_k}{\alpha_0^{(w,k)}(1-p)} r_j, \quad j = 1, \ldots, L_{\max}, \end{cases}$$

and for the idle converters as

$$q_j^c = \begin{cases} 1 - \frac{(1+\sigma)\gamma}{\sigma\mu_0^c} & = 1 - \frac{(1+\sigma)\gamma}{\sigma\alpha_0^c}, \quad j = 0, \\ \frac{(1+\sigma)\gamma}{\sigma\mu_0^c} r_j & = \frac{(1+\sigma)\gamma}{\sigma\alpha_0^c} r_j, \quad j = 1, \ldots, L_{\max}. \end{cases}$$

Using these expressions and equations (5.5) and (5.6) we find that the first-row entries of the matrix $P_w^k(\alpha)$ are given by

$$p_j^{(w,k)} = \begin{cases} (1-p) - \frac{K(1+\sigma)\gamma_k}{\alpha_0^{(w,k)}}(1 - r_1) + pr_1, & j = 0, \\ \left(\frac{K(1+\sigma)\gamma_k}{\alpha_0^{(w,k)}} + p \right) r_{j+1}, & j = 1, \ldots, L_{\max} - 1, \end{cases}$$

and the first-row entries of the matrix $P_c(\alpha)$ are

$$p_j^c = \begin{cases} 1 - \frac{(1+\sigma)\gamma}{\sigma\alpha_0^c}(1 - r_1), & j = 0, \\ \frac{(1+\sigma)\gamma}{\sigma\alpha_0^c} r_{j+1}, & j = 1, \ldots, L_{\max} - 1. \end{cases}$$

Now that the entries of the matrix $P(\alpha)$ are explicitly given in terms of α, γ and γ_k, we solve the system $\alpha = \alpha P(\alpha)$. Due to the structure in Equation (5.4) we find that

$$\alpha_0^l = \frac{1}{\sum_{i=0}^{L_{\max}-1} \sum_{j=i}^{L_{\max}-1} p_j^l} \text{ and } \alpha_i^l = \alpha_0^l \sum_{j=i}^{L_{\max}-1} p_j^l,$$

for $i = 1, \ldots, L_{\max} - 1$ and $l \in \{(w,k), c\}$. To find $\alpha_0^{(w,k)}$ we first evaluate the sum in the denominator to find

$$\begin{aligned} \sum_{i=0}^{L_{\max}-1} \sum_{j=i}^{L_{\max}-1} p_j^{(w,k)} &= (1-p) - \frac{K(1+\sigma)\gamma_k}{\alpha_0^{(w,k)}} + \left(\frac{K(1+\sigma)\gamma_k}{\alpha_0^{(w,k)}} + p \right) \sum_{i=1}^{L_{\max}} \sum_{j=i}^{L_{\max}} r_j, \\ &= (1-p) + \frac{K(1+\sigma)\gamma_k}{\alpha_0^{(w,k)}} \left(E[L] - 1 \right) + \rho. \end{aligned}$$

Therefore we can solve for $\alpha_0^{(w,k)}$ to find

$$\alpha_0^{(w,k)} = \frac{1 - \gamma_k (E[L] - 1) K(1+\sigma)}{1 - p + \rho},$$

and the remaining entries of $\alpha^{(w,k)}$ are

$$\alpha_i^{(w,k)} = \left(\gamma_k K(1+\sigma) + p\alpha_0^{(w,k)} \right) P[L \geq i + 1],$$

for $i = 1, \ldots, L_{\max} - 1$, where $P[L \geq i+1]$ is the probability that the packet size is greater than or equal to $i+1$. In a similar manner we find that the entries of the vector α^c are

$$\alpha_0^c = 1 - \frac{1+\sigma}{\sigma}\gamma(E[L]-1), \text{ and } \alpha_i^c = \frac{1+\sigma}{\sigma}\gamma P[L \geq i+1], \quad i = 1, \ldots, L_{\max} - 1.$$

We have found expressions for the fixed-point vector α, but these are still in terms of γ and γ_k. We now consider each of the three possible values that γ and γ_k can take to find explicit formulas for the fixed point vector in each case.

Case 1: The first case corresponds to a switch with an infinite number of wavelengths that has enough converters to convert every packet and is not overloaded. From equations (5.7) and (5.8) we find that

$$\gamma = \frac{p}{K(1+\sigma)} \sum_{k=1}^{K} \left(1 - \alpha_0^{(w,k)}\right), \text{ and } \gamma_k = \frac{(1-\alpha_0^{(w,k)})p}{K(1+\sigma)}.$$

Using these values we find that

$$\alpha_i^{(w,k)} = \begin{cases} 1 - \rho + p, & i = 0, \\ pP[L \geq i+1], & i = 1, \ldots, L_{\max} - 1. \end{cases}$$

And for the converters we find that

$$\alpha_i^c = \begin{cases} 1 - \frac{1}{\sigma}\left(p(E[L]-1)\right)^2, & i = 0, \\ \frac{p^2}{\sigma}(E[L]-1)P[L \geq i+1], & i = 1, \ldots, L_{\max} - 1. \end{cases}$$

Case 2: The second case corresponds to a system that does not have enough converters to convert every extra-packet. In this case we find that $\gamma = \frac{\sigma\alpha_0^c}{1+\sigma}$, since the converters become the bottleneck of the system. The fixed point for the converters is given by

$$\alpha_0^c = \frac{1}{E[L]} \text{ and } \alpha_i^c = \frac{P[L \geq i+1]}{E[L]}, \quad i = 1, \ldots, L_{\max} - 1.$$

As in this case γ_k is given by

$$\gamma_k = \frac{1 - \alpha_0^{(w,k)}}{\sum_{j=1}^{K}\left(1 - \alpha_0^{(w,j)}\right)} \frac{\sigma}{1+\sigma}\alpha_0^c,$$

we can use the value of α_0^c to find

$$[\alpha_0^{(w,k)} = \frac{E[L] - \sigma(E[L]-1)}{E[L](1-p+\rho)}, \text{ and } \alpha_i^{(w,k)} = \frac{\sigma+\rho}{E[L](1-p+\rho)}P[L \geq i+1],$$

for $i = 1, \ldots, L_{\max} - 1$.

Case 3: The last case considers a heavily loaded switch where the wavelengths are not enough to handle the incoming packets, meaning an extra-packet might find a converter but not an idle wavelength in the output port. In this case we find that γ and γ_k are

$$\gamma = \frac{1-p}{K(1+\sigma)} \sum_{k=1}^{K} \alpha_0^{(w,k)} \text{ and } \gamma_k = \frac{\alpha_0^{(w,k)}(1-p)}{K(1+\sigma)}.$$

Therefore, the vector $\alpha_0^{(w,k)}$ is given by

$$\alpha_0^{(w,k)} = \frac{1}{E[L]}, \text{ and } \alpha_i^{(w,k)} = \frac{P[L \geq i+1]}{E[L]}, \quad i = 1, \ldots, L_{\max} - 1.$$

And the vector α^c is given by

$$\alpha_0^c = 1 - \frac{(1-p)(E[L]-1)}{\sigma E[L]}, \text{ and } \alpha_i^c = \frac{1-p}{\sigma E[L]} P[L \geq i+1], \quad i = 1, \ldots, L_{\max} - 1.$$

The loss probability and the optimal conversion ratio

An interesting observation is that the value of α_0^w and α_0^c does not depend on the distribution of the packet size, but only on its expected value $E[L]$. This becomes relevant when looking at the loss probability of the system p_{loss}, which is the main measure of performance. The loss probability is the ratio between the average number of packets that must be dropped per time slot and the average number of packets that enter the system in each time slot. Therefore, the loss probability is given by

$$p_{\text{loss}} = \frac{\frac{\delta}{1+\sigma} - \gamma}{\frac{p}{1+\sigma}} = \frac{\delta - \gamma(1+\sigma)}{p},$$

where $\delta = \frac{1}{K}\sum_{k=1}^{K} \delta_k$. As shown above, δ and γ depend on the value of α_0^w and α_0^c alone, and therefore p_{loss} does not depend on the packet-size distribution, but only on its expected value. This confirms previous observations [2, 123] related to an apparent insensitivity of the switch performance to the packet-size distribution when the number of wavelengths is large. However, this result relies on the assumption of geometrically-distributed IATs. We have observed that this result no longer holds if the IATs are described by a general DMAP, although even in this case the packet-size distribution appears to have little influence on the loss probability. This will be illustrated in the numerical results in Section 5.4.

Based on the previous results we can consider the question of how many converters are necessary to attain a loss probability equal to zero. Or, in others words, what is the minimum conversion ratio σ^* such that $p_{\text{loss}} = 0$. For p_{loss} to be equal to zero, γ must be equal to $\frac{\delta}{1+\sigma}$, as in case 1. If the conversion ratio is just enough to prevent any losses, then γ must also be equal to $\frac{\sigma\alpha_0^c}{1+\sigma}$, as in case 2. Therefore, we can compute the value of σ^* by equating the value of γ in these two cases. Solving this equation for σ we find that

$$\sigma^* = \rho^2 \left(1 - \frac{1}{E[L]}\right). \tag{5.9}$$

The value of σ^* is proportional to the square of the load and if $E[L]$ tends to infinite (the slot length tends to zero) σ^* tends to ρ^2. A decrease in the mean packet size (increase in the slot length) implies a decrease in the number of converters required to attain zero loss probability.

With a similar analysis we can consider the situation when both cases 1 and 3 occur. This is the point where the wavelengths become the bottleneck of the system. By solving a similar equation as before we find that $\rho = 1$, i.e., the system will only present losses caused by the lack of available wavelengths if it is overloaded ($\rho > 1$). This result is expected since the number of wavelengths is assumed to be infinite and, if no losses are caused by the lack of converters, the only way to observe packet losses is by having a load greater than one.

The previous discussion shows that there exists a fixed point for the mapping defined by $P(\cdot)$ and that this point can be expressed explicitly in terms of the system parameters. However, we have not shown that, starting from any vector $\alpha(0)$, the system always converges to that fixed point. In our experiments we start with an arbitrary initial state $\alpha(0)$ and let the system evolve, finding two possible behaviors. On the one hand, when the system has enough converters to prevent losses the system state converges toward a single state, which coincides with the fixed point described in this section. On the other hand, if the system presents losses due to the lack of converters, it will converge toward a set of points, which are visited periodically. Moreover, we have observed that in the latter case the period is equal to the greatest common divisor of the possible packet sizes (d), and the average of these states is equal to the fixed point described above. Therefore, in the experiments we let the system evolve until a time t such that $||\alpha(t) - \alpha(t-d)|| < \epsilon$, with $\epsilon = 10^{-10}$, which always results in the convergence of the system state. Another important issue is to show that a sequence of finite systems in steady state with increasing number of wavelengths tends toward a limit equal to the fixed point of the mean field. We have observed through simulations that this is the case, but we did not find a formal proof.

5.3 Generalizations

In this section we consider two main generalizations for the model of the optical switch with a centralized pool of converters: first, we assume that the traffic is heterogeneous among the output ports, i.e., the arrival process and the packet-size distribution may be different for each output port; second, we assume that the arrival process is a DMAP instead of the simpler Bernoulli process considered in the previous section. To describe the model we follow a similar approach as in the previous section, considering three steps in each slot (transmission, arrivals and reallocation) and defining transition matrices for each of them. From here onward, when we refer to an arbitrary output port k, we assume $k = 1, \ldots, K$.

The matrices associated with the evolution of the wavelengths in port k are $\bar{S}_w^k$, $\bar{A}_w^k$ and $\bar{Q}_w^k$ for transmission, arrivals and reallocation, respectively. For the converters these are $\bar{S}_c$, $\bar{A}_c$ and $\bar{Q}_c$, respectively. As described in Section 5.1, the arrival process at each wavelength in output port k is a DMAP characterized by the $m_k \times m_k$ matrices $D_0^{(k)}$

and $D_1^{(k)}$. Also, in output port k, the packet size L_k follows a general distribution with finite support Ξ_k and $r_i^k = P[L_k = i]$, for $i \in \Xi_k$. Let $L_{\max}^{(k)} = \max\{i : i \in \Xi_k\}$, i.e., the maximum packet size of the traffic for output port k, and $L_{\max} = \max_{k=1}^{K}\{L_{\max}^{(k)}\}$ the maximum packet size among all the traffic in the switch.

S1 - Transmission: The description of the state of a wavelength in output port k before transmission is made of the scheduling horizon and the state of the arrival process. Therefore the state space is the set $\{(i,j) : 0 \leq i \leq L_{\max}^{(k)}, 1 \leq j \leq m_k\}$. This set is ordered lexicographically and we refer to the subset $\{(i,j) : 1 \leq j \leq m_k\}$ as level i, for $i = 0, \ldots, L_{\max}^{(k)}$. The state space of a converter before transmission is the set $\{i : 0 \leq i \leq L_{\max}\}$ since the only descriptor is the scheduling horizon and the maximum value it can take is $L_{\max}$. Therefore, the transition matrix for a wavelength in output port k during S1 is $\bar{S}_w^k = T_{L_{\max}^{(k)}} \otimes I_{m_k}$, where the Kronecker product $\otimes$ reflects the fact that the packet transmission has no influence on the state of the arrival process. For the converters the transition matrix in this step is simply $\bar{S}_c = T_{L_{\max}}$.

S2 - Arrivals: After transmission, each wavelength may receive a new packet depending on the phase of its arrival process. If the wavelength is idle the packet is scheduled for transmission, otherwise it becomes part of the set of extra-packets to be reallocated. *Before* S2 the state space for a wavelength in port k is $\{(i,j) : 0 \leq i \leq L_{\max}^{(k)} - 1, 1 \leq j \leq m_k\}$. *After* S2 the state space also includes a description of those wavelengths holding an extra-packet, therefore it is the set $\{(i,j) : 0 \leq i \leq 2L_{\max}^{(k)} - 1, 1 \leq j \leq m_k\}$. The first $L_{\max}^{(k)} + 1$ levels describe the case where the wavelength has no extra-packets and a horizon equal to $i = 0, \ldots, L_{\max}^{(k)}$. The remaining levels represent the case where the wavelength has a scheduling horizon equal to $i = 1, \ldots, L_{\max}^{(k)} - 1$ and a packet arrives in this step, making the level equal to $L_{\max}^{(k)} + i$.

The matrix $\bar{A}_w^k$ is therefore an $L_{\max}^{(k)} \times 2L_{\max}^{(k)}$ block-matrix with block-size m_k. The first block-row is given by

$$[\bar{A}_w^k]_{0j} = \begin{cases} D_0^{(k)}, & j = 0, \\ D_1^{(k)} r_j^k, & j = 1, \ldots, L_{\max}^{(k)}. \end{cases}$$

And the remaining block-rows are

$$[\bar{A}_w^k]_{ij} = \begin{cases} D_0^{(k)}, & j = i, \\ D_1^{(k)}, & j = L_{\max}^{(k)} + i, \end{cases}$$

for $i = 1, \ldots, L_{\max}^{(k)} - 1$. Since the arrivals have no effect on the state of the converters, the transition matrix for them is simply $\bar{A}_c = I_{L_{\max}}$.

S3 - Reallocation: For the reallocation step we need the state of the complete system just after arrivals, which is contained in the occupancy vector $\bar{M}^N(t)$. This vector is of size $b = \sum_{i=1}^{K} 2L_{\max}^{(k)} m_k + L_{\max}$, and can be partitioned in $K+1$ vectors as

$$\bar{M}^N(t) = \frac{1}{1+\sigma}\left[\frac{1}{K}\bar{W}_1^N(t), \ldots, \frac{1}{K}\bar{W}_K^N(t), \sigma\bar{C}^N(t)\right],$$

where the $1 \times 2L_{\max}^{(k)} m_k$ vector $\bar{W}_k^N(t)$ describes the state of the wavelengths in output port k after arrivals. The vector $\bar{W}_k^N(t)$ can be partitioned in $2L_{\max}^{(k)}$ vectors $\bar{w}_i^k(t)$ of size m_k. For $i = 0, \ldots, L_{\max}^{(k)}$, the j-th entry of vector $\bar{w}_i^k(t)$ holds the number of wavelengths in output port k with horizon equal to i, phase of the arrival process equal to j and no extra packets. For $i = L_{\max}^{(k)} + l$ and $l = 1, \ldots, L_{\max}^{(k)} - 1$, the j-th entry of $\bar{w}_i^k(t)$ holds the number of wavelengths in output port k with horizon equal to l, holding one extra packet, and with the arrival process in phase j, with $j = 1, \ldots, m_k$. Correspondingly, $\bar{C}^N(t)$ has entries $\bar{c}_i(t)$, which hold the number of converters with horizon equal to i, for $i = 0, \ldots, L_{\max}$.

Therefore, there are $\bar{c}_0(t)$ available converters in the shared pool and $\bar{w}_0^k(t) e_{m_k}$ available wavelengths in output port k, where e_{m_k} is a column vector of size m_k with all its entries equal to one. The number of extra packets in output port k is

$$\bar{d}_k(\bar{M}^N(t)) = \sum_{j=1}^{L_{\max}^{(k)}-1} \bar{w}_{L_{\max}+j}^k(t) e_{m_k}.$$

Thus, the number of extra-packets sent to the converter pool from output port k is given by $\bar{f}_k(\bar{M}^N(t)) = \min\{\bar{w}_0^k(t) e_{m_k}, \bar{d}_k(\bar{M}^N(t))\}$, while the total number of packets sent for conversion is $\bar{f}(\bar{M}^N(t)) = \sum_{k=1}^K \bar{f}_k(\bar{M}^N(t))$. The number of extra-packets that is actually converted, which is limited by the number of available converters, is $\bar{g}(\bar{M}^N(t)) = \min\{\bar{c}_0(t), \bar{f}(\bar{M}^N(t))\}$. Using these quantities we can determine the number of extra-packets that are converted and returned to output port k for transmission in a different wavelength. Since the traffic for all the output ports has the same priority, the probability that an extra-packet sent to the converter pool, from any output port, is actually converted is $\bar{g}(\bar{M}^N(t))/\bar{f}(\bar{M}^N(t))$. Thus, the average number of extra-packets of output port k that are actually converted is

$$\bar{g}_k(\bar{M}^N(t)) = \bar{f}_k(\bar{M}^N(t)) \frac{\bar{g}(\bar{M}^N(t))}{\bar{f}(\bar{M}^N(t))} = \bar{g}(\bar{M}^N(t)) \frac{\bar{f}_k(\bar{M}^N(t))}{\bar{f}(\bar{M}^N(t))}.$$

We can now determine, for $j = 0, \ldots, L_{\max}^{(k)}$, the probability $\bar{q}_j^{(w,k)}(\bar{M}^N(t))$ that an idle wavelength in output port k ends up with a horizon equal to j after the reallocation step. These are given by

$$\bar{q}_j^{(w,k)}(\bar{M}^N(t)) = \begin{cases} 1 - \frac{\bar{g}_k(\bar{M}^N(t))}{\bar{w}_0^k(t) e_{m_k}}, & j = 0, \\ \frac{\bar{g}_k(\bar{M}^N(t))}{\bar{w}_0^k(t) e_{m_k}} r_j^k & j = 1, \ldots, L_{\max}^{(k)}. \end{cases}$$

Similarly, let $\bar{q}_j^c(\bar{M}^N(t))$ be the probability that an idle converter ends up with a horizon equal to j after reallocation, for $j = 0, \ldots, L_{\max}$. These are given by

$$\bar{q}_j^c(\bar{M}^N(t)) = \begin{cases} 1 - \frac{\bar{g}(\bar{M}^N(t))}{\bar{c}_0(t)}, & j = 0, \\ \frac{\bar{g}(\bar{M}^N(t))}{\bar{c}_0(t)} \bar{r}_j, & j = 1, \ldots, L_{\max}. \end{cases}$$

Here $\bar{r}_j$ is the probability that a packet, among the traffic for all the output ports, has a size equal to j, for $j = 1, \ldots, L_{\max}$. It is given by $\bar{r}_j = \frac{1}{\lambda} \sum_{k=1}^K \lambda_k r_j^k$, where λ_k is the

mean arrival rate of the traffic directed to output port k, and $\lambda = \sum_{k=1}^{K} \lambda_k$. The arrival rate at output port k is given by $\lambda_k = \pi_k D_1^{(k)} e_{m_k}$, where π_k is the invariant vector of the transition matrix $D^{(k)} = D_0^{(k)} + D_1^{(k)}$.

Since only the idle wavelengths are affected during S3, the wavelengths with horizon greater than zero before this step, with or without extra-packets, keep the same horizon. Let the matrix $\hat{Q}_w^k(\bar{M}^N(t))$ be built in the same manner as the matrix $Q_w(M^N(t))$ in Equation (5.1), replacing the $q_j^{(w,k)}(\bar{M}^N(t))$ with $\bar{q}_j^{(w,k)}(\bar{M}^N(t))$. Then this is the transition matrix for the scheduling horizon of a wavelength in output port k. Since the reallocation has no effect on the phase of the arrival process, the transition matrix for the state of a wavelength in output port k during S3 is $\bar{Q}_w^k(\bar{M}^N(t)) = \hat{Q}_w^k(\bar{M}^N(t)) \otimes I_{m_k}$. Similarly, let $\bar{Q}_c(\bar{M}^N(t))$ be the $L_{\max} \times L_{\max}$ transition matrix for the converters during this step. The first row of this matrix describes the evolution of an idle converter, therefore $[\bar{Q}_c(\bar{M}^N(t))]_{0j} = \bar{q}_j^c(\bar{M}^N(t))$, for $j = 0, 1, \ldots, L_{\max}$. On the other hand, any converter with a horizon greater than zero keeps the same horizon after this step, hence $[\bar{Q}_c(\bar{M}^N(t))]_{ii} = 1$, for $i = 1, \ldots, L_{\max} - 1$.

5.3.1 Combining S1, S2 and S3 - Mean Field

To describe the evolution of the system we observe it after S2. Therefore, the transition matrix for a wavelength in output port k at time t is $\bar{R}_k^N(\bar{M}^N(t)) = \bar{Q}_w^k(\bar{M}^N(t))\bar{S}_w^k\bar{A}_w^k$, for $k = 1, \ldots, K$. Similarly, the transition matrix for a converter is $\bar{R}_c^N(\bar{M}^N(t)) = \bar{Q}_c(\bar{M}^N(t))\bar{S}_c\bar{A}_c$. Therefore, the transition matrix for an arbitrary object is

$$\bar{R}^N(\bar{M}^N(t)) = \begin{bmatrix} \bar{R}_1^N(\bar{M}^N(t)) & \ldots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \ldots & \bar{R}_K^N(\bar{M}^N(t)) & 0 \\ 0 & \ldots & 0 & \bar{R}_c^N(\bar{M}^N(t)) \end{bmatrix}.$$

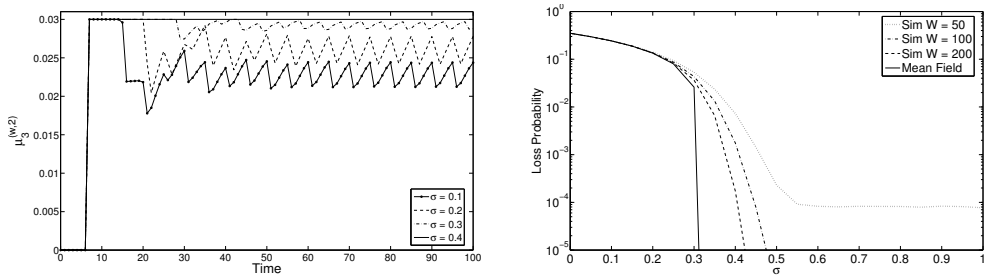
As before, this system can be analyzed within the framework introduced in [79]. For any occupancy vector $\vec{m}$ of size b, the entries of the matrix $\bar{R}^N(\vec{m})$ are continuous in $\bar{m}$ and are independent of the number of objects N. Therefore, $\bar{R}^N(\vec{m})$ trivially converges to $\bar{R}(\vec{m})$ entry-wise, when N tends to infinity. Hence, the evolution of the system with a large number of objects (wavelengths) can be approximated by its mean field, described by the vector

$$\bar{\mu}(t) = \frac{1}{1+\sigma}\left[\frac{1}{K}\bar{\mu}^1(t), \ldots, \frac{1}{K}\bar{\mu}^K(t), \sigma\bar{\mu}^c(t)\right],$$

for $t \geq 0$. Initially the system is assumed to be empty and therefore

$$\bar{C}^N(0) = e_1^{L_{\max}}\bar{A}_c \text{ and } \bar{W}_k^N(0) = \begin{bmatrix} \pi_k & 0_{(L_{\max}^{(k)}-1)m_k} \end{bmatrix} \bar{A}_w,\ k = 1, \ldots, K,$$

where 0_m is a $1 \times m$ zero vector. As this initial state is independent of N, it also complies with the condition of uniform convergence of $\bar{M}^N(0)$ to $\bar{M}(0)$ when N tends to infinity. Then, the initial state of the mean field is $\bar{\mu}(0) = \bar{M}^N(0)$ and its evolution is given by $\bar{\mu}(t+1) = \bar{\mu}(t)\bar{R}(\bar{\mu}(t))$. Then, by [79, Theorem 4.1], for any fixed time t, almost surely, $\lim_{N\to\infty} \bar{M}^N(t) = \bar{\mu}(t),\ t \geq 0$. Hence, the mean field can be used to approximate



(a) $K = 2$, $L \in \{5, 15\}$, Geo IATs, $\rho = 0.6$ (b) $K = 4$, $L \sim \text{Unif}(5, 15)$, Geo IATs, $\rho = 0.6$

Figure 5.2: Mean field and simulation results

the behavior of a switch with a large number of wavelengths, a general arrival process per wavelength, general packet-size distribution, and heterogeneous traffic (the traffic parameters can be different for each output port).

As described in the previous section, the mean field model is time dependent and we have not provided a convergence result when the time t tends to infinity. However, we have performed a large number of experiments and found that the general model behaves similarly to the basic model introduced in the previous section, when the time tends to infinity. Namely, when the model presents no losses, the system state always converges to a unique fixed point, without any influence of the initial state. Also, when the lack of converters causes the system to present losses, the system converges to a set of states that are visited periodically, where the period d is equal to the greatest common divisor of the possible packet sizes of all the traffic entering the switch. Therefore, to obtain long-run performance measures, we start with an empty system and let it evolve until $||\bar{\mu}(t) - \bar{\mu}(t-d)|| < \epsilon$. Once convergence is reached, the performance measures are computed for each of the d states, and these are averaged to obtain the long-run performance measures. This periodic behavior will be illustrated in the next section, together with the effect of the traffic characteristics on the long-run performance of the switch.

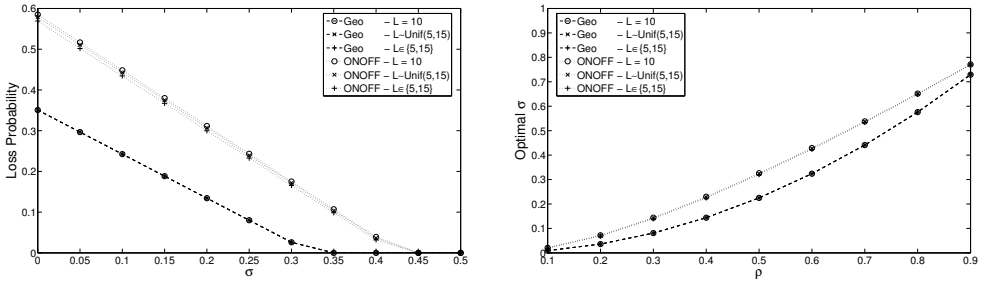
5.4 Results

In this section we start by illustrating the time-dependent behavior of the mean field model. Then we compare the stationary performance of the model against simulations of a switch with a (large) finite number of wavelengths. Afterward we make use of the mean field model to analyze the effect of various traffic parameters on the switch performance. As mentioned before we consider the loss probability and σ^*, the minimum conversion ratio to attain a zero loss probability, as the main performance measures. We examine the effect of the packet-size distribution, the load and the arrival process' burstiness, for both the homogeneous and heterogeneous traffic cases.

5.4.1 Validation

We start by looking at the time-dependent behavior of the mean field model, as shown in Figure 5.2(a). In this scenario the switch has two output ports, homogeneous traffic, geometric IATs, load equal to 0.6, and the packet size is either 5 or 15 with equal probability. In Figure 5.2(a) we depict, as a function of time, the fraction of wavelengths in output port 2 with scheduling horizon equal to 3 ($\mu_3^{(w,2)}$). This selection is arbitrary as any other entry in the state vector shows a similar behavior. Since the mean packet size $E[L]$ is 10 and the load ρ is 0.6, we know from Equation (5.9) that the optimal conversion ratio σ^* is 0.324. Hence, if the conversion ratio is less than this value the system will present losses due to the lack of converters. In this figure we observe that when σ is below 0.324 the system tends to a periodic state, and the period is equal to 5, which is the greatest common divisor of the packet sizes (5 and 15). Also, when the conversion ratio is closer to 0.324 we see that the fluctuations are smaller, and when σ surpasses σ^* the system converges to a single fixed point. We have found the same behavior in a large number of experiments, from which we have concluded that the system may converge either to a single state or to a set of states visited periodically. The former case occurs when the system has enough converters to prevents losses, while the latter is the result of an under-dimensioned conversion ratio. Additionally, when the latter case occurs, the number of different states that the system visits in the long run is equal to the greatest common divisor of the possible packet sizes.

We now compare the loss probability in a finite simulated system with the one computed with the mean field model. In Figure 5.2(b) we show this comparison for a switch with 4 ports, the packet size is uniformly distributed between 5 and 15, the IATs are geometrically distributed and the load per wavelength is 0.6. In this figure the loss probability is depicted against the conversion ratio σ. We observe how the performance of the finite systems tends to that of the mean field when the number of wavelengths per port increases, in this case from 50 to 200. If the conversion ratio is equal to one, and the traffic is uniform with geometric IATs, a finite switch behaves as a Geo/Geo/KW/KW loss queue, and therefore, as the conversion ratio increases, the loss probability of a finite switch converges toward that of the loss queue. For instance, for $W = 50$ we observe that this minimum loss probability is around 10^{-4}, and for a conversion ratio of 0.55 the loss probability of the finite system has already reached a value very close to the minimum. A similar behavior occurs for $W = 100$ and $W = 200$, but in this case the loss probability is so small that simulations become computationally prohibitive. The main difference between the mean field and a finite system is that the mean field model can be dimensioned to attain zero loss probability (σ^*), while the conversion ratio in any real finite system will be dimensioned to attain a very small loss probability ($\hat{\sigma}$). Although σ^* will typically be an optimistic value for $\hat{\sigma}$, it is a very close approximation, especially if the number of wavelengths is large. Therefore, the mean field model can be used to provide a fast-to-compute value to start the search for the actual $\hat{\sigma}$, thus restricting the search for the value of $\hat{\sigma}$ to a small neighborhood above σ^*. Another very relevant feature of the mean field model is that it let us analyze the effect of various traffic parameters on the performance of the switch, without performing any time-consuming simulations,



(a) Loss probability. $\rho = 0.6$, $\gamma = 5$

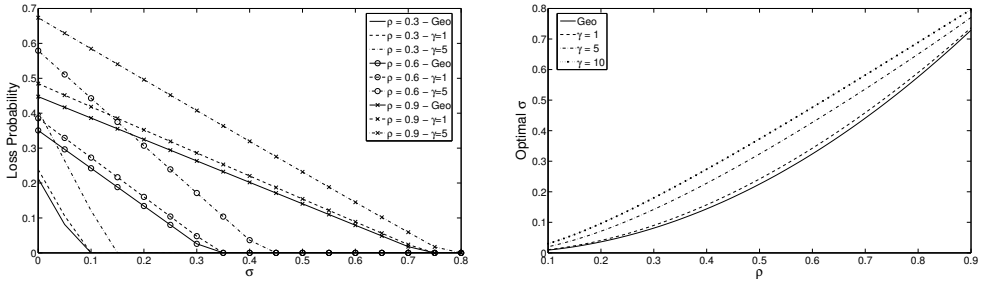
(b) Optimal σ. $\gamma = 5$

Figure 5.3: Effect of the packet-size distribution

which become more expensive as the number of wavelengths increases, particularly if the measure of interest (the loss probability) is very small. The remainder of this section is devoted to the analysis of the effect of the traffic parameters on both the loss probability and σ^*.

5.4.2 Homogeneous traffic

In this section we consider the case where the traffic is homogeneous, i.e., the traffic to all the output ports has the same characteristics. Figure 5.3 illustrates the effect of the packet-size distribution on the switch performance. As it was established in Section 5.2, when the IATs follow a geometric distribution, the only influence of the packet-size distribution is through its mean. In this case we change the distribution without altering the mean, and Figure 5.3(a) shows that this has no effect on the loss probability when the IATs are geometrically distributed. We also consider a more general arrival process, called ON-OFF, which is a particular case of a DMAP. An ON-OFF process has an underlying chain with two states: in the so-called ON state the process generates arrivals with geometric IATs, while in the OFF state no arrivals are generated. This kind of arrival process has been previously used to model the arrival process at an optical switch [95,116,123]. The duration of the ON and OFF periods is geometrically distributed, with the mean duration of the OFF periods being γ times that of the ON periods. In Figure 5.3 we also observe that for ON-OFF arrivals with $\gamma = 5$ the effect of the packet-size distribution is rather small, even though the distributions we consider are significantly different. The three packet-size distributions are: deterministic with packet size $B = 10$; uniformly distributed between 5 and 15; and a distribution with two equally likely values, 5 and 15. Figure 5.3(b) shows that the effect of the packet-size distribution on σ^* is also rather small, a result that holds for a broad range of load values. At this point we must recall that we are considering the homogeneous-traffic case, where all the wavelengths have the same arrival process and packet-size distribution. In this case the results are obtained for $K = 2$, but this parameter has no effect on the results since the number of converters is proportional to the total number of wavelengths, which is infinite for any value of K. In other words, if the traffic is homogeneous all the wavelengths have the same characteristics and the number of ports becomes irrelevant because there is an



(a) Loss probability. $L \sim \text{Unif}(5, 15)$

(b) Optimal σ. $L \sim \text{Unif}(5, 15)$

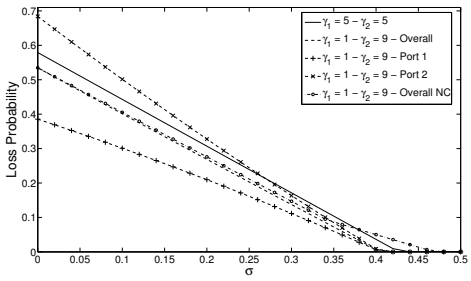
Figure 5.4: Effect of the burstiness

infinite number of identical wavelengths and the number of converters is defined as a proportion of the total number of wavelengths. Therefore, when presenting the results under the assumption of homogeneous traffic we do not need to specify the value of K, as the results are the same for every $K \geq 1$. This also means that when the number of wavelengths tends to infinity and the traffic among the ports is homogeneous, there is no difference between the performance of the centralized and non-centralized architectures.

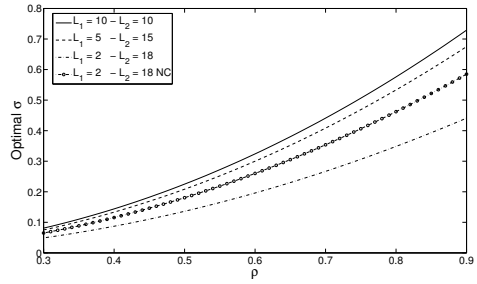
Although the packet-size distribution seems to have little effect on the performance of the switch, Figure 5.3 shows a significant difference between the results for geometric and ON-OFF arrivals. We now consider the effect of the arrival process' burstiness in more detail, by means of the ON-OFF process. A simple measure of the burstiness of an arrival process is the ratio between its peak rate and its mean rate [102]. For geometric IATs these two rates are equal and the ratio is one. For the ON-OFF process the peak rate is q (the rate of the geometric IATs during the ON periods), the mean rate is $\frac{q}{\gamma+1}$ and the ratio is $\gamma + 1$. Therefore, increasing the value of γ while keeping the load fixed increases the burstiness of the process, which is expected since the same number of arrivals will occur in shorter time intervals (ON periods), followed by longer silent (OFF) periods. Figure 5.4(a) depicts the results for geometric and ON-OFF arrivals with various values of γ (one and five) and various loads (0.3, 0.6 and 0.9). We observe that a larger burstiness implies a significantly larger loss probability, and therefore a larger conversion ratio to achieve zero losses. This effect is considerable for any load, but it is particularly relevant for mid loads. The effect of the burstiness on σ^* is shown in Figure 5.4(b), where the larger absolute effect for mid loads is evident. For instance, for $\rho = 0.9$ the value of σ^* for geometric IATs is around 0.73, while for ON-OFF($\gamma = 5$) arrivals is 0.77. For a load of 0.5, σ^* is 0.22 for geometric IATs and 0.32 for ON-OFF($\gamma = 5$) arrivals. It appears that when the load is high, the main cause of losses is the load and the burstiness only comes in second place.

5.4.3 Heterogeneous traffic

We now consider heterogeneous traffic conditions, starting with the case of a difference in burstiness among the traffic directed to the output ports. In Figure 5.5(a) we show the loss probability for a switch with two output ports, where the only difference between the



(a) Heterogeneous burstiness ($\gamma_1 = 1$), $\rho = 0.6$, $L \sim \mathrm{U}(5, 15)$

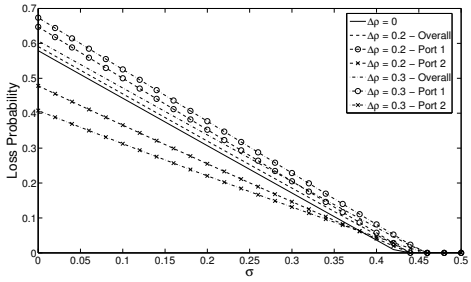


(b) Heterogeneous packet size. Geo IATs

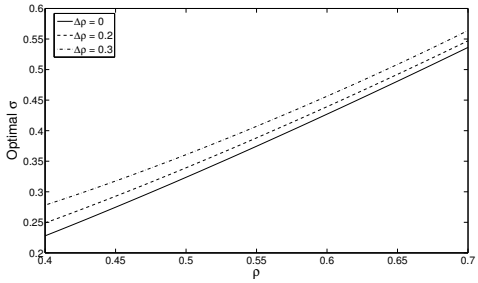
Figure 5.5: Effect of heterogeneity - $K = 2$

traffic directed to these ports is related to the burstiness. We consider two cases: in the first, both output ports have the same ON-OFF arrival process, with parameter $\gamma = 5$; in the second case, the traffic to the first output port is an ON-OFF process with $\gamma = 1$, while for the second port the traffic follows an ON-OFF process with $\gamma = 9$. In the first case, the loss probabilities per port are the same, equal to the overall loss probability. For the second case the ports present a very dissimilar performance, where the larger losses correspond to the port with the more bursty traffic, as expected. When the conversion ratio is highly under-dimensioned the difference between the performance of the ports is very large, as is the overall loss probability. As the conversion ratio increases and gets closer to σ^*, the difference between the loss probabilities of the ports decreases. We also observe that the overall loss probability is smaller for the heterogeneous-traffic than for the homogeneous-traffic case, and there is also a difference in the value of σ^*, favorable to the heterogeneous case. This is particularly relevant since, for the type of arrival processes we are considering, the overall burstiness is the mean of the burstiness among the traffic for all ports. As a result, in both scenarios the overall burstiness is 5, but when the traffic is heterogeneous both the loss probability and σ^* are smaller than in the homogeneous case. In addition, Figure 5.5(a) also depicts the loss probability for the heterogeneous case when there converters are arranged in a non-centralized fashion (labeled NC). We observe that the overall loss probability is very similar for the centralized and non-centralized cases up to a certain value of the conversion ratio (in this instance 0.35). For larger values of σ the difference increases considerably, with the centralized architecture showing a smaller loss probability and a significantly smaller σ^*.

In Figure 5.5(b) we also consider heterogeneous traffic conditions, but this time the difference is in the packet-size distribution. In this case the load offered to the switch is the same, but there are differences in the mean packet size per port. Figure 5.5(b) shows the value of σ^* for three cases: in the first case the packet size of both ports is deterministic and equal to 10; in the second the packets for the first port have size equal to 5 while those for the second have size equal to 15; in the last case the packets for the first and second ports are of size 2 and 18, respectively. In all the cases the load is the same for both output ports. We see that the packet size has an important effect on σ^*, particularly for high loads. When the asymmetry in the packet sizes is larger, the conversion



(a) Loss probability. $\rho = 0.6$, $L \sim \text{Unif}(5, 15)$, $\gamma = 5$



(b) Optimal σ^*. $L \sim \text{Unif}(5, 15)$, $\gamma = 5$

Figure 5.6: Effect of load heterogeneity - $K = 2$

requirements decrease, but this is significant only if the asymmetry is sufficiently large. In particular, we observe little difference between the first and the second cases, but a large difference between these two and the third case. The gain in conversion requirements can be partly attributed to the centralized location of the converters, as opposed to having a pool of converters per port. The non-centralized case is illustrated for the third case in Figure 5.5(b) with the label NC, which shows the conversion requirements to attain a zero loss probability if each port had its own set of converters. As stated above, under the assumption of an infinite number of wavelengths, the number of ports K has no effect on the performance of the switch if the traffic is homogeneous. Therefore, in that case the difference between centralized or non-centralized conversion vanishes as the number of wavelengths becomes large. However, as soon as the heterogeneity in the traffic is considered (either in burstiness or packet-size distribution) we observe an important gain obtained by centralizing the conversion resources. The other reason for having a lower loss probability in the latter two cases is that, as the ports have the same load, the arrival rate for the port with the smaller packet size is larger, and therefore there is a larger proportion of small packets, which means that the mean packet size is also smaller.

As a final scenario we consider heterogeneously loaded ports. We analyze a switch with two ports, where the difference between the load of the first (ρ_1) and second (ρ_2) ports is given by $2\Delta\rho$, with $\Delta\rho = \rho_1 - \rho = \rho - \rho_2$, and ρ is the overall load. Figure 5.6(a) illustrates the effect of load heterogeneity, under ON-OFF($\gamma = 5$) arrivals, for three values of $\Delta\rho$: 0, 0.2 and 0.3. We observe how the overall loss probability is only slightly affected, while the loss probability per port is significantly different, especially if the conversion ratio is too small compared to σ^*. Also, the effect of $\Delta\rho$ on the minimum conversion ratio to attain a zero loss probability, shown in Figure 5.6(b), is rather small, even when the asymmetry in the loads is large. In this case, the value of σ^* is identical under non-centralized and centralized conversion, meaning that no gain in conversion resources is obtained by using centralized conversion when the asymmetry in traffic arises from a difference in the load. However, in the non-centralized case the converters would need to be allocated in each port proportionally to the loads. As the load is a very dynamic parameter, a converter allocation based on the mean load per port would result in a combination of periods with many idle converters (lowly loaded) and periods with many

losses (highly loaded). Also, an allocation based on the peak load would require a large number of converters per port. In a dynamic-load scenario, the centralized architecture will provide an important gain, as it will be able to combine peak periods for some ports with valley periods for others. In addition, it is highly unlikely to find a real scenario where the only difference in the characteristics among the ports' traffic is the load, while the burstiness and the packet-size distribution are the same for every port. Typically, the difference will be in all these characteristics, and therefore the centralized architecture will provide a better performance and require fewer converters than the non-centralized one, even if the number of wavelengths is large.

Chapter 6

Partial Conversion and Fiber Delay Lines

In this chapter we consider an OBS switch equipped with a pool of full-range wavelength converters and a set of FDLs *per output port*. Each of these ports has W wavelengths that can be used to simultaneously transmit the same number of packets. To analyze the performance of this switch we introduce a mean field model (see Appendix A.3) that is exact when the number of wavelengths tends to infinity. Moreover, the performance of a switch with a large but finite number of wavelengths tends to that of the mean field as the number of wavelengths increases. Therefore, the mean field model can be used to approximate the performance of a switch with a large number of wavelengths, as has been confirmed by means of simulations. Furthermore, the time required to evaluate a particular scenario with the mean field model is typically a few seconds on a personal computer, compared to the long-lasting simulations that are needed when the number of wavelengths is large and the performance measures to evaluate are small (e.g. packet loss probability). As a result, the mean field model can be used to efficiently evaluate the effect of various parameters on the performance of the switch. These include, among others, the number of converters, the number and granularity of the FDLs, the packet size distribution and the arrival process' burstiness. The main features of the mean field model can be summarized as follows: (i) contention is resolved by means of both wavelength conversion and optical buffering; (ii) there is a pool of converters and FDLs per output port; (iii) the number of converters may vary between zero and the number of wavelengths, i.e., the switch has *partial conversion*; (iv) two different policies are considered for wavelength allocation: minimum horizon and minimum gap (see Section 6.1); (v) the arrival process is assumed to be a general DMAP (see Appendix A.1), which is able to represent general correlated inter-arrival times; (vi) the burst size is assumed to follow a general distribution with finite support.

The main performance measure for the switch is the loss probability, which is the probability that an incoming burst has to be dropped due to the infeasibility of transmitting, converting or buffering it. Since the number of wavelengths is large, it is expected that the loss probability will decrease to a near-zero value if the number of converters is

large enough, as in an Erlang loss system. Therefore, it becomes relevant to determine the minimum number of converters required to attain a near-zero loss probability. We can reformulate this in terms of the conversion ratio $\sigma = C/W$, where C and W are the number of converters and wavelengths per output port, respectively (notice that here C is the number of converters per output port and not its total number in the switch as in Chapter 5). The goal is therefore to find the minimum value of σ such that the loss probability is almost zero, which will be denoted as σ^*. As will be shown later, with the mean field model we are able to compute the value of σ^* in a single run, which allows us to consider the effect that other parameters have on σ^*, specially the number and granularity of the FDLs. In this direction we have found that the effect of the number of FDLs on the loss rate and σ^* is highly dependent on the load. While for high loads it may have little or no effect, for mid loads the addition of a few FDLs may reduce both the loss rate and σ^*. Also, if the number of WCs is insufficient ($\sigma < \sigma^*$), increasing the number of FDLs may not improve the loss rate. However, under bursty traffic the effect of adding FDLs is more substantial, helping to reduce the conversion requirements. The effect of the granularity on σ^* also depends on the load: while for low and mid loads the set of granularity values with the best performance depends on the packet size distribution, for high loads the performance is inversely proportional to the granularity. As a result, among the best possible values for the granularity under mid loads, the results favor the selection of a small granularity since this requires fewer converters to attain a near-zero loss probability at high loads. In addition, the minimum horizon policy shows a consistently worse performance than its minimum gap counterpart. And when $\sigma < \sigma^*$ the addition of converters may even worsen the loss rate under the minimum horizon policy.

This chapter is organized as follows. Section 6.1 introduces the architecture and operation of the switch under analysis. Section 6.2 describes the mean field model in detail. Section 6.3 compares the results of the mean field model with results from the simulation of a finite system. This section also analyzes the effect of various parameters on the performance of the switch, with special emphasis on the effect of the allocation policies, the number and granularity of the FDLs and the burstiness of the arrival process.

6.1 Switch Arquitecture

In this section we describe the operation and main features of the optical switch, the wavelength allocation policies and some modeling issues relevant for the description of the switch. The optical switch under analysis, shown in Figure 6.1, is made of K input/output ports, each one connected to a fiber with W wavelengths. The switch works in a synchronous manner, where the time is divided in equally-spaced slots and the state of the switch is observed at slot boundaries. The arrival process at each wavelength is modeled as a DMAP (see Appendix A.1) characterized by the set of $m \times m$ matrices $\{B_0, B_1, \ldots, B_{L_{\max}}\}$, where $L_{\max}$ is the maximum packet length. As mentioned in the previous chapter, the class of DMAP processes has been successfully used before to model the arrival process at a bufferless optical switch [2, 95, 124]. It includes many well-known processes as special cases, e.g., the discrete-time versions of the Poisson process, interrupted Poisson process (IPP), Markov modulated Poisson process (MMPP), etc. When

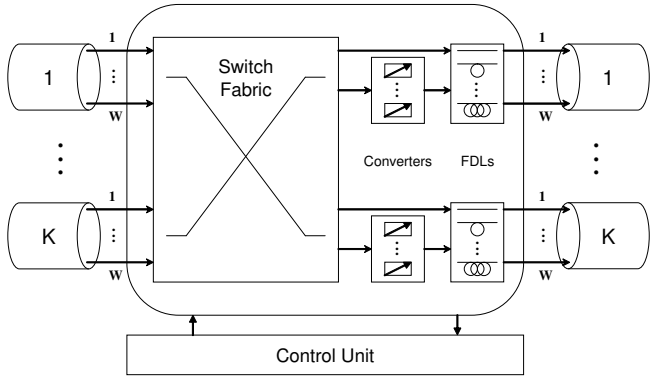


Figure 6.1: Optical switch with K input/output ports, W wavelengths, converters and FDLs

a burst arrives, it is switched to the corresponding output port using its own wavelength, called home wavelength. If the home wavelength is available for transmission in the output port, the burst starts transmission immediately. If the wavelength is already transmitting another burst or has scheduled the transmission of a burst waiting in the FDL, the new packet is buffered using the FDL. In case the FDL has no available buffering capacity in that wavelength, the incoming burst is converted to a different wavelength using one of the available converters. If there are no idle converters or no wavelengths with available buffering capacity, the burst must be dropped. Thus, to resolve contention the switch first tries to buffer the signal and only if this is not feasible it tries to convert it, aiming to minimize the converter usage, as the *minConv* strategy in [45].

To analyze the performance of the switch we assume the incoming traffic to be uniformly distributed among the output ports, so we can focus on a single output port. To describe the state of one of these ports we consider two types of objects: wavelengths and converters. The state of a single wavelength is described by the scheduling horizon, which is the time until all the packets already scheduled for transmission in that wavelength have left the switch. If the horizon is equal to 0 and a packet of size L arrives, it can start transmission immediately and the horizon increases to L. On the other hand, if the incoming burst finds a horizon equal to h, it will experience a delay of at least h units before actual transmission. As the buffering is carried out by a set of N FDLs, the possible delay a packet can experience depends on the length of these delay lines. Here we assume the N FDLs have linearly growing length with granularity D, as explained in the introduction to this part. With this setup an incoming packet that observes a scheduling horizon equal to h has to wait for $D\left\lceil \frac{h}{D} \right\rceil$ slots, if $h \leq ND$. If the packet is of size L the new value of the horizon is $D\left\lceil \frac{h}{D} \right\rceil + L$. Notice, in this particular case the wavelength remains unused during $D\left\lceil \frac{h}{D} \right\rceil - h$ slots just prior to the packet transmission. We refer to this wasted period as a *gap*. If h is greater than ND the packet cannot be buffered in the FDL using the same wavelength and it must be reallocated in another wavelength with horizon less than or equal to ND.

A packet that cannot be buffered in its home wavelength, called an extra-packet, can be reallocated if there is both a wavelength with scheduling horizon no greater than ND and an available converter. Hence, it is necessary to check the state of all the wavelengths and the converters. There are C converters per output port and the state of a single converter is also described by its scheduling horizon. As a converter has no buffering capacity, its horizon reduces to the time required by the packet already in service to be completely converted. Then, if an extra-packet of size L finds an available converter (and there is a wavelength with available buffering capacity) the horizon of the selected converter changes its value from 0 to L. Naturally, when this conversion occurs the horizon of the wavelength that receives the burst increases its value as described previously. As mentioned before, the number of converters C per output port is determined as a fraction of the number of wavelengths W, i.e., $C = \sigma W$, where σ is the conversion ratio. If an extra-packet finds an idle converter it has to choose a wavelength among those with horizon less than or equal to ND. This selection can be made using two different allocation policies: *minimum horizon*, which selects the wavelength with the minimum scheduling horizon; and *minimum gap*, which selects the wavelength with a horizon such that the allocation of a new packet generates a gap of minimum value among all available wavelengths. Recall, the gap is the difference between the horizon observed by an incoming packet and the actual delay that a packet assigned to the wavelength must face. An important assumption is that each wavelength with available buffering capacity can only receive one extra-packet during one slot, even if it has enough free FDLs to receive more than one additional packet. Removing this assumption would complicate both the wavelength allocation policies and their corresponding modeling aspects.

To model the evolution of the switch in a single slot we consider the following order of events: first, the busy wavelengths (resp. converters) transmit (resp. translate) part of the packet in service, reducing their horizons by one. Second, a new packet may arrive at each wavelength with a probability related to the current state $i \in \{1, \ldots, m\}$ of its arrival process; the packet is buffered if there is space available in its home wavelength, otherwise it becomes part of the set of extra-packets. Third, the extra-packets are converted to a different wavelength with available buffering capacity. An extra-packet that does not find an available converter or a wavelength with buffering capacity must be dropped. The probability that a packet is dropped is called the loss probability and is considered the main performance measure.

6.2 The Mean Field model

Our model is based on a general result for a system of interacting objects introduced in [79] (see Appendix A.3). In our case, the system consists of two types of objects: wavelengths and converters. To describe the evolution of the system during a time slot we start with the state of the objects at the beginning of the time slot. Then we determine the transition matrices that describe the state transitions at each of the three steps: transmission, arrivals and reallocation. The matrices associated to these steps are S_k, A_k and Q_k, respectively, where the subscript k may be equal to w or c depending on whether the matrix describes the transition of a wavelength or a converter. These matrices are

then used to build a complete description of the evolution of the switch at each time slot. At the beginning of slot t (before packet transmission) the state of a *single* wavelength can be described by the tuple $\{(H(t), J(t)), t \geq 0\}$, with $H(t)$ the scheduling horizon of the wavelength and $J(t)$ the phase of its arrival process. Its state space is the set $\{(i,j) : 0 \leq i \leq ND + L_{\max}, 1 \leq j \leq m\}$. Similarly, a converter can be described by its scheduling horizon $\{\bar{H}(t), t \geq 0\}$ with state space $\{i : 0 \leq i \leq L_{\max}\}$. We now define the transition matrices for each of the three steps.

6.2.1 Step 1, packet transmission

In the first step (S1) the horizon of each busy wavelength and each busy converter is reduced by one, as they transmit (translate) part of the scheduled packets. Let T_n be the $(n+1) \times n$ matrix with entries

$$[T_n]_{ij} = \begin{cases} 1, & i = j = 0, \\ 1, & j = i - 1, i = 2, \ldots, n, \\ 0, & \text{otherwise.} \end{cases}$$

As in the previous chapter, we label the rows and columns of an $m \times n$ matrix from 0 to $m-1$ and from 0 to $n-1$, respectively. Then the evolution of a single wavelength in S1 is given by the transition matrix $S_w = T_{ND+L_{\max}} \otimes I_m$, where I_n is the identity matrix of size n and $\otimes$ denotes the Kronecker product. This product shows that packet transmissions affect the horizon value, but not the phase of the arrival process. Accordingly, the matrix $S_c = T_{L_{\max}}$ contains the transition probabilities for a converter during S1.

6.2.2 Step 2, packet arrivals

The arrival of packets during the second step (S2) has no influence on the state of a converter; therefore, its transition matrix in this step is given by $A_c = I_{L_{\max}}$. Similarly, the matrix A_w describes the transition of a single wavelength in S2, but its definition is more involved. If after S1 the wavelength has a horizon less than or equal to ND, it can accept any incoming packet. On the other hand, if the scheduling horizon is greater than ND and a packet arrives, it cannot be buffered and becomes part of the extra-packets. To keep track of the size of the (possibly empty) set of extra-packets, the horizon and the phase of the arrival process, we separate the resulting state space after S2 into two sets. The first set is $\{(i,j) : 0 \leq i \leq ND + L_{\max}, 1 \leq j \leq m\}$, which captures two cases: the horizon was less than or equal to ND after S1, and the transition in S2 results in a horizon equal to i and a phase of the arrival process equal to j; or the horizon i was greater than ND, and no packet was received. In this first set the wavelength holds zero extra-packets. The second set is $\{(ND + L_{\max} + i, k, j) : 1 \leq i \leq L_{\max} - 1, 1 \leq k \leq L_{\max}, 1 \leq j \leq m\}$, considering the case where the horizon was equal to $ND+i$ after S1 and the arrival process (during S2) generates a packet of size k and makes a transition to phase j. The two sets of states are put together by imposing a lexicographic order, resulting in a transition matrix A_w of size $m(ND + L_{\max}) \times m(ND + L_{\max}^2 + 1)$ (since the horizon after step S1 is at most $ND + L_{\max} - 1$).

To explicitly describe the matrix A_w we partition the state space before and after S2 in levels. Before S2, level i is the set of states $\{(i,j) : 1 \leq j \leq m\}$, for $0 \leq i \leq ND + L_{\max} - 1$. After S2, there are two subsets of levels, corresponding to the two subsets of the state space described above: in the first subset, level i is the set of states $\{(i,j) : 1 \leq j \leq m\}$, for $0 \leq i \leq ND + L_{\max}$; in the second subset, level (i,k) is the set of states $\{(ND+L_{\max}+i,k,j) : 1 \leq j \leq m\}$, for $1 \leq i \leq L_{\max}-1$ and $1 \leq k \leq L_{\max}$. Let the matrix $A_w^{\{i,i'\}}$ contain the transition probabilities from level i to level i', for $0 \leq i \leq ND + L_{\max} - 1$ and $0 \leq i' \leq ND + L_{\max}$. These matrices are given by

$$A_w^{\{i,i'\}} = \begin{cases} B_0, & 1 \leq i = i' \leq ND + L_{\max} - 1, \\ B_k, & i' = D\left\lceil \frac{i}{D} \right\rceil + k,\ 0 \leq i \leq ND,\ 1 \leq k \leq L_{\max}. \end{cases}$$

To explain this expression we consider the two cases separately: in the first case the wavelength is busy and receives no packet in this slot; in the second case the wavelength receives a new packet that can be immediately transmitted or buffered, increasing the scheduling horizon. Similarly, let the matrix $A_w^{\{i,(i',k')\}}$ contain the transition probabilities from level i to level (i',k'), for $0 \leq i \leq ND + L_{\max} - 1$, $1 \leq i' \leq L_{\max} - 1$ and $1 \leq k' \leq L_{\max}$. These matrices are given by

$$A_w^{\{ND+i,(i',k')\}} = B_{k'},$$

for $1 \leq i = i' \leq L_{\max} - 1$ and $1 \leq k' \leq L_{\max}$. These transitions correspond to the case where the wavelength is busy and receives a packet that cannot be buffered. Since all the possible transitions in S2 have been covered, we now turn to the last step.

6.2.3 Step 3, packet conversion and reallocation

In this step (S3) the extra-packets that arrived in the previous step are reallocated using the available converters. To determine the evolution of a *single* wavelength or converter it is necessary to consider the state of the whole system (W wavelengths and C converters). It is important to stress however that we do not need to determine the joint evolution of multiple wavelengths or converters for the mean field result to apply (for more on this see Appendix A.3). Let $w_i(t)$ be the $1 \times m$ vector whose j-th entry contains the number of wavelengths holding no extra-packets with horizon equal to i and phase of the arrival process equal to j after S2, for $0 \leq i \leq ND + L_{\max}$ and $1 \leq j \leq m$. Additionally, let the j-th entry of the $1 \times m$ vector $w_{(ND+L_{\max}+i,k)}(t)$ be the number of wavelengths at time t with horizon equal to $ND + i$ after S1 that receive a packet of size k in S2, after which the phase of the arrival process is equal to j, for $1 \leq i \leq L_{\max} - 1$, $1 \leq k \leq L_{\max}$ and $1 \leq j \leq m$. The vector

$$M^{W,(w)}(t) = \frac{1}{W}[w_0(t), \ldots, w_{ND+L_{\max}}(t), w_{(ND+L_{\max}+1,1)}(t), \ldots, w_{(ND+2L_{\max}-1,L_{\max})}(t)]$$

describes the state of all the wavelengths at time t before S3 as fractions of the total number of wavelengths W. Analogously, let $c_i(t)$ be the number of converters with horizon equal to i at time t before S3, for $i = 0, \ldots, L_{max}$. The state of the converters at time t, as a fraction of the total number of converters, is therefore contained in the vector $M^{W,(c)}(t) = \frac{1}{C}[c_0(t), \ldots, c_{L_{\max}}(t)]$. The superscript W indicates that the system is composed of W wavelengths and $C = \sigma W$ converters.

The state of the complete system at time t can be described by the vector

$$M^W(t) = \left[\frac{1}{1+\sigma}M^{W,(w)}(t),\ \frac{\sigma}{1+\sigma}M^{W,(c)}(t)\right],$$

which is called the occupancy vector and contains the fraction of objects in each state, including both wavelengths and converters. The weights $\frac{1}{1+\sigma}$ and $\frac{\sigma}{1+\sigma}$ are the proportion of wavelengths and converters, respectively, in relation to the total number of objects. Based on this vector, we can define the matrices $Q_w(M^W(t))$ and $Q_c(M^W(t))$, which contain the transition probabilities in S3 under the *minimum horizon* policy for wavelengths and converters, respectively. The matrices $\bar{Q}_w(M^W(t))$ and $\bar{Q}_c(M^W(t))$ contain similar information for the *minimum gap* policy. However, to specify these matrices it is necessary to first determine the number and size of the extra-packets that can actually be converted, regardless the wavelength allocation policy.

Let $d_i(M^W(t))$ be the number of extra-packets of size i, for $1 \leq i \leq L_{\max}$, which is given by

$$d_i(M^W(t)) = \sum_{k=1}^{L_{\max}-1} w_{(ND+L_{\max}+k,i)}(t)e_m,$$

where e_m is a column vector of size m with all its entries equal to one. Therefore, the total number of extra-packets is $d(M^W(t)) = \sum_{i=1}^{L_{\max}} d_i(M^W(t))$. Also, let $W_{ND}(M^W(t))$ be the number of wavelengths with horizon less than or equal to ND after S2, i.e.,

$$W_{ND}(M^W(t)) = \sum_{i=0}^{ND} w_i(t)e_m.$$

The number of extra-packets that can actually be converted ($\hat{d}(M^W(t))$) is given by the minimum of three quantities: the number of packets to convert, the number of wavelengths with available buffering capacity, and the number of available converters, i.e.,

$$\hat{d}(M^W(t)) = \min\{d(M^W(t)), W_{ND}(M^W(t)), c_0(t)\}.$$

As each wavelength with available buffering capacity receives at most one extra-packet, $\hat{d}(M^W(t))$ is also the number of wavelengths that receive an extra-packet in S3. The selection of these $\hat{d}(M^W(t))$ wavelengths is done using the *minimum horizon* or *minimum gap* policies. Once a wavelength is chosen to receive an extra-packet, the selection of the packet is done randomly among the $d(M^W(t))$ extra-packets. This means that the probability that a selected wavelength receives a packet of size k, for $1 \leq k \leq L_{\max}$, is $p_k(M^W(t)) = \frac{d_k(M^W(t))}{d(M^W(t))}$. Relying on these definitions, the purpose of the following subsections is to determine the transition matrices for both wavelength allocation policies.

Minimum Horizon

To determine the wavelengths that, under the *minimum horizon* (*minH*) policy, will receive the $\hat{d}(M^W(t))$ extra-packets, we define the quantities $\alpha_i(M^W(t))$ as the number of wavelengths with horizon less than or equal to i after S2, i.e., $\alpha_i(M^W(t)) =$

$\sum_{k=0}^{i} w_k(t)e_m$, for $0 \leq i \leq ND$. As the extra-packets are assigned to the wavelengths with the smallest horizons, we need to find an $h(M^W(t))$ such that

$$\alpha_{h(M^W(t))-1} < \hat{d}(M^W(t)) \leq \alpha_{h(M^W(t))}.$$

This means that the wavelengths with horizon strictly less than $h(M^W(t))$ receive one extra packet each, while those with a horizon strictly greater than $h(M^W(t))$ receive no extra-packets. The packets that cannot be accommodated in the wavelengths with horizons up to $h(M^W(t))-1$ are randomly assigned among the wavelengths with horizon equal to $h(M^W(t))$. Let $\theta(M^W(t))$ be the probability that a wavelength receives a packet in S3 if its horizon is equal to $h(M^W(t))$. This is given by

$$\theta(M^W(t)) = \frac{\hat{d}(M^W(t)) - \alpha_{h(M^W(t))-1}}{w_{h(M^W(t))}(t)e_m}.$$

Now we can define $r_i(M^W(t))$, the probability that a wavelength with horizon equal to i receives an extra-packet in S3 under the *minH* policy, as

$$r_i(M^W(t)) = \begin{cases} 1, & 0 \leq i < h(M^W(t)), \\ \theta(M^W(t)), & i = h(M^W(t)), \\ 0, & h(M^W(t)) < i \leq ND. \end{cases}$$

Let $u_{ii'}(M^W(t))$ be the probability that a wavelength with horizon i and holding no extra-packets after S2 ends up with a horizon equal to i' after S3, for $0 \leq i \leq ND + L_{\max}$ and $0 \leq i' \leq ND + L_{\max}$. These probabilities are given by

$$u_{ii'}(M^W(t)) = \begin{cases} 1 - r_i(M^W(t)), & 0 \leq i = i' \leq ND, \\ r_i(M^W(t))p_k(M^W(t)), & 0 \leq i \leq ND, \\ & i' = D\left\lceil \frac{i}{D} \right\rceil + k, \\ 1, & ND < i = i' \leq ND + L_{\max}. \end{cases}$$

Now let $u_{(i,k),i'}(M^W(t))$ be the probability that a wavelength with horizon $ND+i$ after S1, that received a packet of size k in S2, ends up with a horizon equal to i' after S3, for $1 \leq i \leq L_{\max} - 1$, $1 \leq k \leq L_{\max}$ and $0 \leq i' \leq ND + L_{\max}$. Since such a wavelength does not alter its horizon, irrespective of whether the extra-packet is successfully reallocated or not, the transition probabilities are given by

$$u_{(i,k),ND+i'}(M^W(t)) = \begin{cases} 1, & 1 \leq i = i' \leq L_{\max} - 1, \\ & 1 \leq k \leq L_{\max}, \\ 0, & \text{otherwise.} \end{cases}$$

Let $U(M^W(t))$ be the $(ND + L_{\max}^2 + 1) \times (ND + L_{\max} + 1)$ matrix that describes the evolution of the horizon during S3. The first $ND + L_{\max} + 1$ rows of this matrix have entries $u_{ii'}(M^W(t))$, for $0 \leq i \leq ND + L_{\max}$. The remaining $(L_{\max} - 1)L_{\max}$ rows are made by the entries $u_{(i,k),i'}(M^W(t))$ in lexicographic order, for $1 \leq i \leq L_{\max} - 1$ and $1 \leq k \leq L_{\max}$. Therefore, the transition matrix for a single wavelength during S3 is

$$Q_w(M^W(t)) = U(M^W(t)) \otimes I_m,$$

making explicit that the allocation of extra-packets has no effect on the phase of the arrival process.

With regard to the converters, only those with horizon equal to 0 may be affected during S3 since these are used to translate the $\hat{d}(M^W(t))$ extra-packets. Let $b_i(M^W(t))$ be the probability that an idle converter receives a packet of size i in S3, for $1 \leq i \leq L_{\max}$. Also, let $b_0(M^W(t))$ be the probability that the converter remains idle. Clearly,

$$b_i(M^W(t)) = \begin{cases} \frac{c_0(t)-\hat{d}(M^W(t))}{c_0(t)}, & i = 0, \\ \frac{\hat{d}(M^W(t))}{c_0(t)} p_i(M^W(t)), & 1 \leq i \leq L_{\max}. \end{cases}$$

Therefore, the entries of the $L_{\max} \times (L_{\max}+1)$ transition matrix for a single converter in S3 can be defined as

$$[Q_c(M^W(t))]_{ij} = \begin{cases} b_j(M^W(t)), & i = 0,\ 0 \leq j \leq L_{\max}, \\ 1, & 1 \leq i = j \leq L_{\max}-1, \\ 0, & \text{otherwise.} \end{cases}$$

Minimum Gap

In this section we determine the matrices $\bar{Q}_w(M^W(t))$ and $\bar{Q}_c(M^W(t))$ to describe the evolution of the system during S3 under the *minimum gap* (*minG*) policy. Since the wavelength allocation policy has no effect on the state of the converters, $\bar{Q}_c(M^W(t)) = Q_c(M^W(t))$. To specify the transition matrix for a single wavelength we start by defining the gap function $v(h) = D\left\lceil \frac{h}{D} \right\rceil - h$, which is the size of the gap created when assigning a packet to a wavelength with horizon h, for $0 \leq h \leq ND$. Now we can define $g_i(M^W(t))$ as the number of wavelengths with $v(\cdot) = i$, which is given by

$$g_i(M^W(t)) = \sum_{\{j:v(j)=i\}} w_j(t) e_m, \quad 0 \leq i \leq D-1.$$

In a similar manner as in the previous section, we define $\gamma_i(M^W(t))$ as the number of wavelengths with $v(\cdot) \leq i$, i.e., $\gamma_i(M^W(t)) = \sum_{j=0}^{i} g_j(M^W(t))$, for $0 \leq i \leq D-1$. In this case we need to find an $x(M^W(t))$ such that

$$\gamma_{x(M^W(t))-1} < \hat{d}(M^W(t)) \leq \gamma_{x(M^W(t))}.$$

Thus, $\gamma_{x(M^W(t))-1}$ extra-packets can be assigned to the wavelengths with $v(\cdot) < x(M^W(t))$. The packets that cannot be accommodated in these wavelengths are distributed among those with $v(\cdot) = x(M^W(t))$, while the rest of the wavelengths receive zero extra-packets. In this case, however, we use the *minH* policy to allocate the remaining $\hat{d}(M^W(t)) - \gamma_{x(M^W(t))-1}$ extra-packets among the wavelengths with $v(\cdot) = x(M^W(t))$ (as opposed to randomly). Since the horizon h can be expressed as $h = D\left\lceil \frac{h}{D} \right\rceil - v(h)$ and the wavelengths that may receive a packet have $v(\cdot) = x(M^W(t))$, we only need to focus on $l(h) = \left\lceil \frac{h}{D} \right\rceil$, which takes values between 0 and N. Let $f_i(M^W(t))$ be the number of wavelengths with horizon h such that $v(h) = x(M^W(t))$ and $l(h) = i$, for $0 \leq i \leq N$. Also, let $\phi_i(M^W(t)) = \sum_{j=0}^{i} f_j(M^W(t))$ be the number of wavelengths with horizon h such

that $v(h) = x(M^W(t))$ and $l(h) \leq i$, for $0 \leq i \leq N$. We then need to find a $y(M^W(t))$ such that

$$\phi_{y(M^W(t))-1} < \hat{d}(M^W(t)) - \gamma_{x(M^W(t))-1} \leq \phi_{y(M^W(t))}.$$

Then, among the wavelengths with horizon h such that $v(h) = x(M^W(t))$, one extra-packet is assigned to each wavelength with $l(h) < y(M^W(t))$, no extra-packet is assigned to those with $l(h) > y(M^W(t))$, and the remaining extra-packets are randomly assigned among the wavelengths with $l(h) = y(M^W(t))$. Therefore, the probability that a wavelength with horizon h such that $v(h) = x(M^W(t))$ and $l(h) = y(M^W(t))$ receives an extra-packet during S3 is

$$\eta(M^W(t)) = \frac{\hat{d}(M^W(t)) - \gamma_{x(M^W(t))-1} - \phi_{y(M^W(t))-1}}{f_{y(M^W(t))}(M^W(t))}.$$

Now we can define $\bar{r}_i(M^W(t))$ as the probability that a wavelength with horizon equal to i receives an extra-packet in S3 under the *minG* policy, given by

$$\bar{r}_i(M^W(t)) = \begin{cases} 1, & 0 \leq v(i) < x(M^W(t)), \\ 1, & v(i) = x(M^W(t)), \\ & l(i) < y(M^W(t)), \\ \eta(M^W(t)), & v(i) = x(M^W(t)), \\ & l(i) = y(M^W(t)), \\ 0, & \text{otherwise.} \end{cases}$$

Based on these probabilities we can build the matrix $\bar{U}(M^W(t))$ in the same manner as we did with the matrix $U(M^W(t))$ for the *minH* policy, but replacing the $r_i(M^W(t))$ by $\bar{r}_i(M^W(t))$, for $0 \leq i \leq ND$. Thus, the transition matrix of a wavelength in S3 under the *minG* policy is $\bar{Q}_w(M^W(t)) = \bar{U}(M^W(t)) \otimes I_m$.

6.2.4 Computation of $M^W(t)$ for large W

In the previous sections we built the transition matrices related to each of the three main events (steps) in a slot, for wavelengths and converters separately. These matrices can be combined to describe the evolution of a single object as a DTMC. We will observe the system just after S2 and, therefore, the state of the wavelengths (resp. converters) at time t is described by the vector $w^W(t)$ (resp. $c^W(t)$). Since the order of the events is S3, S1 and S2, the transition matrices of a single wavelength or converter under the *minH* policy are

$$R_k^W(M^W(t)) = Q_k(M^W(t))\, S_k\, A_k,\ k \in \{w, c\}.$$

The superscript W refers to the total number of wavelengths in the system. We now combine these two matrices into $R^W(\cdot)$ to describe the evolution of a single object, which can be a wavelength or a converter, as a DTMC with two non-communicating classes

$$R^W(M^W(t)) = \begin{bmatrix} R_w^W(M^W(t)) & 0 \\ 0 & R_c^W(M^W(t)) \end{bmatrix}.$$

A similar construction can be made to determine the matrix $\bar{R}^W(M^W(t))$ for the *minG* policy. We now consider the framework in [79] to compute $M^W(t)$ when W is large.

The discussion is for the *minH* policy, but it applies *mutatis mutandis* for the *minG* policy. In [79] the authors show that, under some mild conditions, a system of interacting objects converges to its mean field when the number of objects is large. The mean field is a time-dependent deterministic system that can be used to approximate the behavior of a system with a large number of objects (a brief description of the mean field result in [79] can be found in Appendix A.3). In our case the objects are of two classes (wavelengths and converters) and their evolution is described by the matrix $R^W(\vec{m})$, where $\vec{m}$ is a $1 \times m(ND + L_{\max}^2 + L_{\max} + 1)$ occupancy vector. The first condition for this result to hold is that the entries of the transition matrix of a single object $[R^W(\vec{m})]_{ij}$ converge uniformly to some $[R(\vec{m})]_{ij}$ on the set of all occupancy vectors when $W(1+\sigma) \to \infty$. In our model the transition matrix $R^W(\vec{m})$ is actually independent of the number of objects $W(1+\sigma)$. This can be seen by dividing all the quantities involved in the computation of the probabilities $u_{ii'}(M^W(t))$ and $b_i(M^W(t))$ by $W(1+\sigma)$. This means that $R(\vec{m}) = R^W(\vec{m})$. The second condition is that $[R(\vec{m})]_{ij}$ must be continuous in $\vec{m}$, which also holds for both allocation policies. Since both conditions are valid for the model described by the matrix $R(\vec{m})$, we can approximate the evolution of the system by means of the mean field, which is described by the vector $\mu(t)$, for $t \geq 0$. Let $\mu(t) = \left[\frac{1}{1+\sigma}\mu^{(w)}(t),\ \frac{\sigma}{1+\sigma}\mu^{(c)}(t)\right]$, for $t \geq 0$. The initial state of the wavelengths is defined as $\mu^{(w)}(0) = [\pi_B,\, 0, \ldots, 0]$, where the $1 \times m$ vector π_B is the stationary probability distribution of the Markovian arrival process. Similarly, the vector $\mu^{(c)}(0) = [1,\, 0, \ldots, 0]$ describes the initial state of the converters. The initial distribution is independent of the number of objects and establishes that all the wavelengths and converters are idle at time 0. Now, let the mean field model evolve as $\mu(t+1) = \mu(t)R(\mu(t))$, then, by [79, Theorem 4.1], for any fixed time t, almost surely,

$$\lim_{W \to \infty} M^W(t) = \mu(t).$$

Using the mean field model we can compute the state of the system at time t by performing t vector-matrix multiplications, where the vector is of size $1 \times m(ND + L_{\max}^2 + 1)$. Additionally, at each time slot the matrix $Q(\vec{m})$ (or $\bar{Q}(\vec{m})$) must be computed since it depends on the value of the occupancy vector. However, if the packet-size distribution is independent of the arrival process the description of the system after S2 can be simplified. In this case the probability distribution of the extra-packets' size is equal to the original packet-size distribution. Therefore it is not necessary to keep track of the size of the extra-packets, but only their number is required, reducing the size of the occupancy vector to $1 \times m(ND + 2L_{\max})$. Also, the structure of the transition matrices in S1 can be exploited to further reduce the computation times.

In addition to approximate the state of a switch with a large number of wavelengths at any finite time t, we are particularly interested in its long-run behavior, but the mean field model is time-dependent and gives no information about this behavior. However, we have numerically observed that when the conversion ratio is large enough to prevent losses caused by the lack of available converters, the state of the system converges to a unique fixed point. When the conversion ratio is not enough to avoid packet losses the system shows a stationary periodic behavior. The period has been observed to be equal to the greatest common divisor of the possible packet sizes. Even though we do not provide a

formal proof of this fact, the results presented in the next section, as well as many others not included here, support this observation. Actually, a formal proof appears to be hard since the evolution of the sequence of occupancy vectors $\{\mu(t),\, t \geq 1\}$ is determined by the matrix $R(\mu(t))$, whose entries are a nonlinear function of the occupancy vector. Moreover, it must be shown that $\{\mu(t),\, t \geq 1\}$ converges either toward a single point (if the number of converters is enough to prevent losses), or toward a set of points that the sequence will visit cyclically (if there are losses because of the lack of converters). Let q be the greatest common divisor of the possible packet sizes. As we do not know in advance if the conversion ratio is enough to prevent losses or not[1], we observe the system every q time slots to check the difference in the entries of the occupancy vector, and we let it evolve until this difference is less than $\epsilon = 10^{-10}$. For each of the q states we compute the performance measures, as shown in the next section, and their average is the value of the long-run performance measures.

6.2.5 Computation of the performance measures

If at time t the mean field has reached one of the q states toward which it converges, then $d(M^W(t))$ is also the number of packets requiring conversion per slot in this state, which we call the *spill rate*. Similarly, $\hat{d}(M^W(t))$ is called the *conversion rate*, while $d(M^W(t)) - \hat{d}(M^W(t))$ is the *loss rate*. In a system with W wavelengths the total arrival rate is $W\lambda$, where λ is the arrival rate at each wavelength, given by

$$\lambda = \pi_B \sum_{k=1}^{L_{\max}} B_k e_m = \pi_B (I_m - B_0) e_m.$$

Therefore the spill probability p_{spill}, i.e., the probability that an incoming packet requires conversion, is given by $p_{\text{spill}} = \frac{d(M^W(t))}{W\lambda}$. Dividing the numerator and denominator by the number of objects $W(1+\sigma)$, we get

$$p_{\text{spill}} = \frac{\delta(M^W(t))}{\frac{\lambda}{1+\sigma}},$$

where $\delta(M^W(t)) = \frac{d(M^W(t))}{W(1+\sigma)}$ is independent of the number of objects. In a similar manner, we define $\hat{\delta}(M^W(t))$ as $\frac{\hat{d}(M^W(t))}{W(1+\sigma)}$, which allows us to define the conversion probability p_{conv} and the loss probability p_{loss} as

$$p_{\text{conv}} = \frac{\hat{\delta}(M^W(t))}{\frac{\lambda}{1+\sigma}},\ p_{\text{loss}} = \frac{\delta(M^W(t)) - \hat{\delta}(M^W(t))}{\frac{\lambda}{1+\sigma}}.$$

6.3 Results

The purpose of this section is two-fold: first, we illustrate the time-dependent behavior of the mean field model as well as its convergence toward a state that matches well

[1] Actually, by running the mean field model once with $\sigma = 1$, we can determine the required σ value at once.

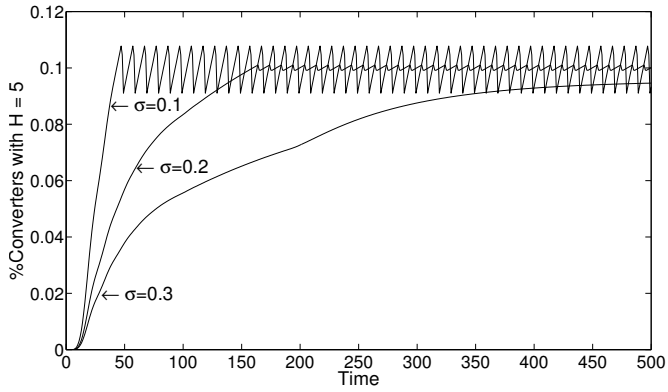
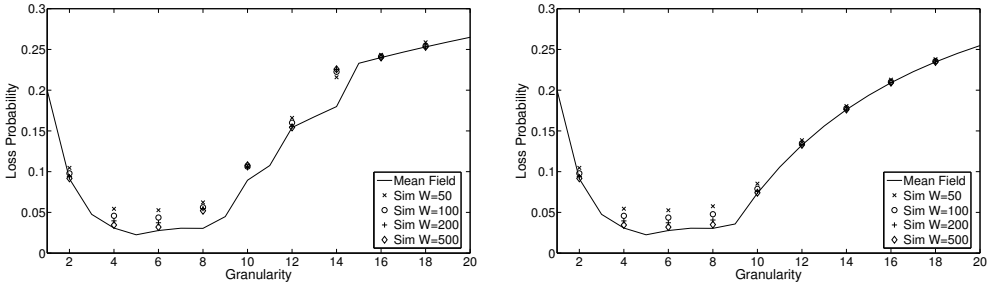


Figure 6.2: Time-dependent behavior of a switch with $N = 3$, $\rho = 0.8$, $D = 10$, geometric IATs and packet size equal to 10

with the results obtained by means of simulation. This will be considered in Section 6.3.1. Second, we make use of the mean field model to analyze the effect of the switch parameters on its performance. We consider the loss probability as the main measure of performance and put special emphasis on the minimum conversion ratio required to attain zero loss probability, referred to as σ^*. In a switch with a finite number of wavelengths, the goal is to determine a conversion ratio such that the loss probability stays below a certain predefined threshold. Since there are no analytical models available to determine the exact loss probability in a switch with a large number of wavelengths, FDLs and partial wavelength conversion, the only alternative is to rely on simulation. However, to estimate a very small loss probability using simulation requires long computation times since the event that must be observed (a packet loss) becomes very unlikely. One of the main advantages of the mean field model is the fast computation of the approximate loss probability and σ^* for any particular scenario with a large number of wavelengths. This allows the analysis of the effect that the various switch parameters have on these performance measures. Sections 6.3.2 and 6.3.3 deal with these issues, where the latter is concerned with the effect of the arrival process' burstiness on σ^*.

6.3.1 Validation

Given the time-dependent nature of the mean field model, there is a natural interest in the behavior of the state vector $\mu(t)$ as a function of time. In Figure 6.2 we illustrate this behavior using the fraction of converters with horizon equal to 5, i.e., $\mu_5^{(c)}(t)$. The selection of this value is arbitrary as all the other entries in the state vector behave in a similar manner. To fix the arrival rate we use the load $\rho = \lambda E[L]$, where $E[L]$ is the expected value of the packet size. In this scenario the switch has $N = 3$ FDLs per output port, the load ρ is 0.8, the granularity is $D = 10$, the burst length equals 10, the inter-arrival times (IATs) follow a geometric distribution (meaning $B_0 = 1 - 0.8/10 = 0.92$ and $B_{10} = 0.8/10 = 0.08$), the policy is *minG* and the conversion ratio is between 0.1 and 0.3. As can be seen in Figure 6.2, when the conversion ratio is equal to 0.1 the



(a) *minimum horizon* policy (b) *minimum gap* policy

Figure 6.3: Mean field model vs. simulation for a switch with $N = 5$, $\rho = 0.8$, $\sigma = 0.1$, packet size equal to 10 and geometric IATs

state of the converters is highly variable and after a short warm-up period it adopts a periodic behavior. When the conversion ratio rises to 0.2 the warm-up period becomes longer and the state of the converters is clearly less variable, but the period is exactly the same and equal to the packet size, in this case 10 slots. Finally, if the conversion ratio is equal to 0.3 no losses are caused by the lack of converters. In this case the warm-up period is even longer, but the system reaches a unique fixed point. A similar behavior has been observed in all the experiments performed, with a periodic long-run behavior and period equal to the greatest common divisor of the possible packet sizes. This periodic behavior arises when the conversion ratio is not enough to prevent packet losses. This is an important observation as it indicates that an under-dimensioned number of WCs leads to a periodic system behavior. If there are plenty of converters to translate any extra-packet, the system converges to a unique fixed point, as in Figure 6.2 for $\sigma = 0.3$.

Since the mean field model tends converges to a set of states in the long run, we can use these states to compute the performance measures of the system as indicated in Section 6.2.5. A first question to address is how the mean field model approximates the behavior of a finite system. In Figure 6.3 we compare the results of the mean field model with results from simulation of a switch with 50, 100, 200 and 500 wavelengths. The estimates from simulations have confidence intervals with half width less than 1% of the mean, obtained with the batch-means method. As can be expected, the simulations require long execution times to obtain a small confidence interval for the loss probability. Figure 6.3 shows how the performance of the finite system tends to that of the mean field model, getting closer as the number of wavelengths increases. In this scenario, as in many others, the convergence for the *minG* policy, shown in Figure 6.3(b), is smoother than for the *minH* policy, shown in Figure 6.3(a). For both policies, the accuracy of the mean field approximation depends on the granularity, being almost exact for granularities well above the packet size. For granularities between 2 and 10, the performance of the finite system smoothly converges to that of the mean field for the *minG* policy. On the other hand, the convergence for the *minH* policy shows different patterns for different granularity values. Notably, when $D = 14$ the performance of the finite system does not appear to converge to the mean field. We have observed that, in this case the loss

probability of a sequence of finite systems with increasing number of wavelengths first increases and then decreases toward the loss probability of the mean field, but this only occurs when the finite system has a few thousand wavelengths. On the other hand, the *minG* policy aims at minimizing the gaps in the FDLs, and an increase in the number of wavelengths directly implies more options to allocate an extra-packet while creating the smallest possible gap. Therefore, increasing the number of wavelengths will reduce the loss probability under this policy and the convergence will be smoother than under the *minH* policy. As a result, when approximating the performance of a finite system with a given number of wavelengths, the mean field model is expected to be less accurate under the *minH* than under the *minG* policy.

6.3.2 The combined effect of FDLs and WCs

One of the main characteristics of the mean field model is its ability to include both partial wavelength conversion and buffering as solutions for contention resolution. We exploit this feature in this section by analyzing the effect of three main parameters: the conversion ratio σ, the number of FDLs N and their granularity D, as well as the wavelength allocation policy. The arrival process is assumed to be geometric as the effect of burstiness in the arrival process will be the topic of the next section. We start by comparing the spill, conversion and loss probabilities for both allocation policies. In Figure 6.4 these three quantities are shown for a switch with $N = 3$ FDLs, granularity $D = 10$, load equal to 0.8 and packet size with equally probable values 8 and 12. For both policies the conversion probability increases linearly with the number of converters up to a point from which it no longer increases. During the interval where this probability increases the converters are the bottleneck of the system, and therefore they are busy all the time. When the switch has enough converters to translate any extra-packet, i.e., when spill and conversion probabilities are equal, the switch no longer experiences losses due to the lack of converters. Notice, we can determine the σ value where the loss rate becomes zero (σ^*) by running the mean field model once with $\sigma = 1$ and noting the percentage of busy converters, solving the dimensioning problem of WCs in a single run. From the figure we observe that the *minG* policy requires a smaller conversion ratio than the *minH* policy to reach the point where spill and conversion probabilities are the same. Furthermore, from this point on the spill probability under *minH* is larger than under *minG*, confirming the well-known result that *minH* is less efficient in managing the buffering resources (FDLs).

An observation that can be made from Figure 6.4, also found in Figure 6.5(a) as well as in many other experiments, is the existence of jumps in the spill and loss probabilities as a function of the conversion ratio, for the *minH* policy. These jumps are closely related to the discrete nature of the FDLs and the way in which the *minH* policy reallocates the extra-packets. As this policy selects the wavelengths with minimum horizon, the reallocated packets go first to the wavelengths with horizon 0 and, if the number of converted packets is larger than the number of wavelengths with horizon 0, the packets are sent to the wavelengths with horizon equal to 1. However, this allocation creates large gaps (of size $D - 1$) in the wavelengths that receive the converted packets. This

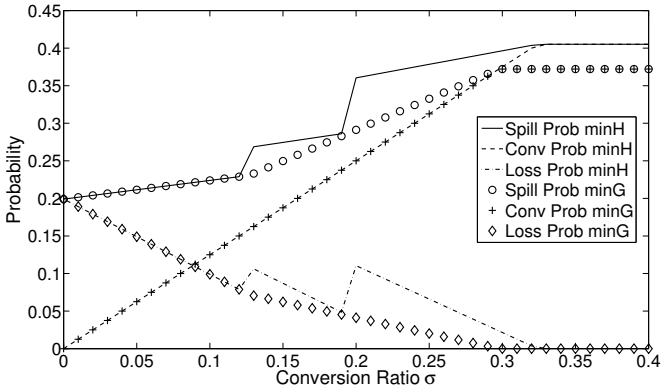
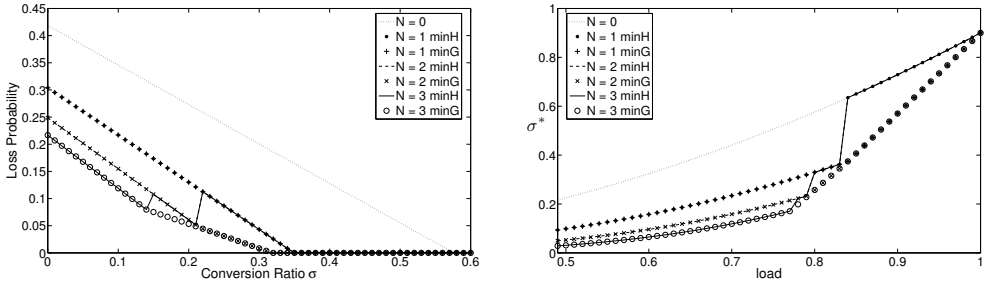


Figure 6.4: Comparison of policies for a switch with $N = 3$, $\rho = 0.8$, $D = 10$, geometric arrivals and packet size equal to $\{8, 12\}$

implies that the gap size distribution is affected in a bad manner, reducing the capacity of the wavelengths and causing the spill probability to increase. Hence, the jump in the spill probability, and therefore in the loss probability, is caused by an increase in the conversion ratio that makes the system able to convert more packets than the wavelengths with horizon equal to 0 are able to admit. This jump can be seen in Figure 6.4 when σ goes from 0.12 to 0.13. The other jumps occur similarly when the conversion ratio goes from a value in which the reallocated packets can be handled by the wavelengths with horizon less than or equal to iD to a value in which they cannot, for $1 \leq i \leq N$. Notice that the number of jumps is at most equal to N, but might be less than this value.

A relevant issue in the design of an optical switch is the influence of the number of FDLs on the loss probability. Figure 6.5(a) shows the loss probability as a function of the conversion ratio, for a variable number of FDLs and both allocation policies. The packet size can be 5 or 15 with equal probability, the load is 0.8 and the granularity is 10. The effect of adding FDLs on the loss probability depends on the conversion ratio. If the conversion ratio is large enough, then adding more FDLs has *no effect*. However, the conversion ratio σ where the loss rate drops to zero does depend on N. For instance, in Figure 6.5(a), having $N = 1$ FDLs allows us to use significantly fewer WCs compared to having zero FDLs, while increasing N to 2 has a smaller effect, and an additional FDL has no effect (as a buffer capacity of $N = 2$ suffices with $C = 0.3W$ WCs). If σ is such that the switch has losses due to the lack of converters, then the addition of buffering capacity might reduce the losses substantially. However, adding an extra FDL might also have no effect at all, even if the switch presents losses. This is clear in Figure 6.5(a) for $\sigma = 0.25$, where the loss with two FDLs is lower than with one, but the addition of a third makes no difference.

As stated before, we can determine the value of σ at which the loss probability drops to zero (σ^*) in a single run of the mean field model. In Figure 6.5(b) we illustrate how the load affects the value of σ^* under both policies. In this case the IATs follow a geometric distribution, the packet size is uniformly distributed between 5 and 15, and the granularity is 9. As expected, a higher load implies a larger σ^*. Also, for high loads the *minG* policy



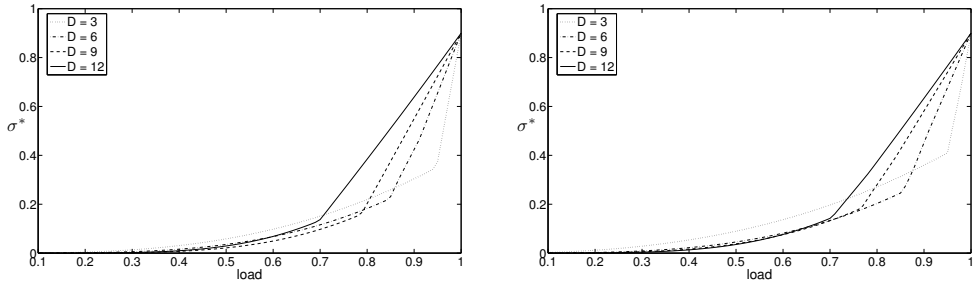
(a) $\rho = 0.8$, $D = 10$, geometric arrivals and packet size equal to $\{5, 15\}$

(b) $D = 9$, geometric arrivals and packet size uniformly distributed between 5 and 15

Figure 6.5: Comparison of policies

requires a smaller conversion ratio to achieve zero losses than the *minH* policy. In relation to the number of FDLs, it is clear that the addition of one FDL reduces the value of σ^* for the *minG* policy, but the effect of additional FDLs depends on the load. For high loads, there is no difference in having one or more FDLs, while for mid and low loads the addition of FDLs may reduce the value of σ^*. This behavior can be explained as follows. If the switch has enough converters to prevent losses and the load is one, the probability that a wavelength has a horizon less than ND after S1 is almost zero in the fixed point. When the load diminishes, the probability that the horizon is between $(N-1)D$ and $ND-1$ smoothly increases, but for values less than $(N-1)D$ it remains close to zero. To obtain a positive probability of having a wavelength with horizon less than $(N-1)D$ it is necessary for the load to go below a certain threshold, which in Figure 6.5(b) corresponds to 0.83. This behavior is independent of the value of N, explaining why the addition of more than one FDL has no effect on the conversion ratio required to achieve zero losses for loads over 0.83 in this scenario. Similar thresholds can be found for the values of the load required to have a positive probability that a wavelength has a horizon between $(i-1)D$ and $iD-1$, for $1 \leq i \leq N$. Hence, for loads above these thresholds having more than $N - i + 1$ FDLs has no effect on σ^*. These thresholds coincide with the location of the jumps for the *minH* policy, but under this policy the probability of having a horizon less than ND is zero if $\sigma \geq \sigma^*$ and the load is greater than 0.83. If the load goes below this value, the probability of a horizon between $(N-1)D$ and $ND-1$ suddenly becomes positive and takes similar values to those of the *minG* policy. Therefore, both policies reach a similar σ^* at $\rho = 0.83$, but the *minG* policy does it in a smooth manner while the *minH* policy shows a big reduction in σ^* when the load goes from from 0.84 to 0.83. We may conclude that incorporating one or two FDLs may result in a significant cost reduction, as fewer WCs are needed. However, the results suggest that additional FDLs have little use as they affect the required number of WCs in a less profound manner, especially for higher loads. The effect of the number of FDLs on σ^* will be discussed again in the next section when looking at the effect of burstiness in the arrival process.

The granularity of the FDL is a parameter with a significant influence on the loss probability for the single-wavelength buffer, as shown in [26, 71, 116]. In Figure 6.6 we illustrate the effect of the granularity on σ^* for two different packet-size distributions. For



(a) B = 10 (b) $B = \{5, 15\}$

Figure 6.6: Effect of the granularity on σ^* for a switch under *minG* policy, geometric arrivals and 3 FDLs

clarity reasons the results are only shown for the *minG* policy. The corresponding results for *minH* have a similar behavior as a function of D, but include the jumps already shown in previous figures, redounding in a worse performance. In Figure 6.6(a) we consider the case where the packet size is fixed and equal to 10 slots. Here we observe two different behaviors for the value of σ^* depending on the load. For low to mid loads, the optimal granularity is $D = 9$, as has been suggested before for fixed-length packet sizes. However, in the second region (high loads) this is no longer the case. At a load around 0.7 there is a pronounced change in the slope of the curve corresponding to $D = 12$. This change rapidly puts this curve above the others, making it the one with the highest requirements in terms of converters. A similar change in slope is suffered by the other curves, in an order that is inversely proportional to the granularity. Therefore, for high loads (in this case above 0.85) a lower granularity means a smaller σ^*. A similar behavior is observed in Figure 6.6(b), where the packet size can be 5 or 15 with equal probability. In this case however there is almost no difference among the granularities between 6 and 12 along the first region of the load range. This result agrees with previous observations related to the larger set of optimal granularities when the packet size is variable. Therefore, among the best possible values for the granularity in the first region, the results just described favor the selection of a small granularity since this requires fewer converters to attain a near-zero loss probability at high loads.

6.3.3 The effect of the arrival process' burstiness

In this last section we analyze the effect of the burstiness in the arrival process on the minimum conversion ratio to attain zero losses σ^*. This is possible due to the versatility of the arrival process assumed by the model (DMAP). In particular we consider an ON-OFF process with two states, where in one state the process generates arrivals with geometric IATs while in the other state no arrivals are generated. This kind of arrival process has been previously used to model the arrival process in an optical switch [95, 116, 124], and it was also used in the previous chapter. The duration of the ON and OFF periods (the sojourn time of the chain in each state) is geometrically distributed with the mean duration of the OFF periods being κ times that of the ON periods. In Chapter 5 we

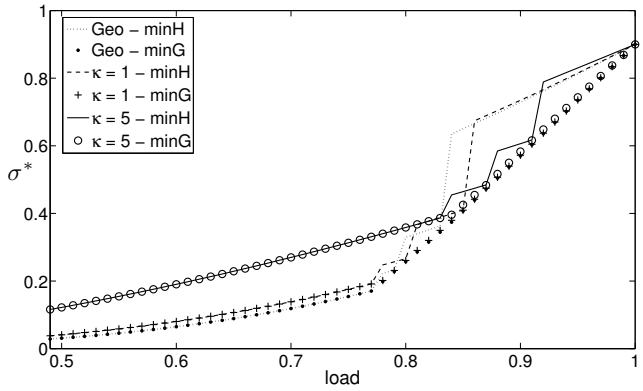
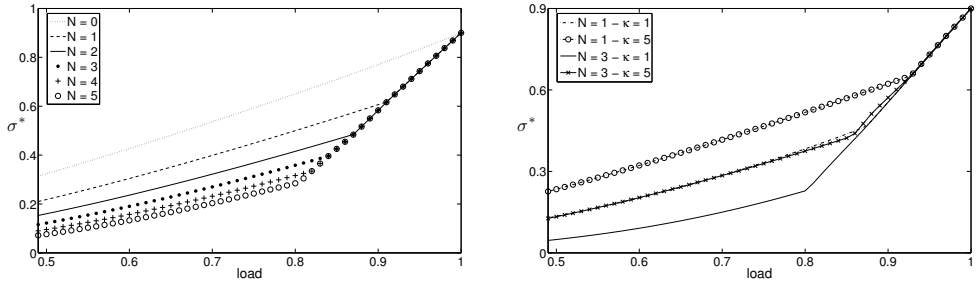


Figure 6.7: Comparison of policies for a switch with an ON-OFF arrival process, 3 FDLs, $D = 8$ and packet size equal to $\{5, 15\}$

referred to κ as γ. Here we prefer κ to avoid confusion with the γ_x defined in the previous section. Recall that a simple measure of the burstiness of an arrival process is the ratio between its peak rate and its mean rate [85, 102]. For geometric IATs these two rates are equal and the ratio is one. For the ON-OFF process the peak rate is q (the rate of the geometric IATs during the ON periods), the mean rate is $\frac{q}{\kappa+1}$ and the ratio is $\kappa + 1$. Therefore, increasing the value of κ while keeping the load fixed increases the burstiness of the process. Figure 6.7 shows the effect of the burstiness on σ^*. As expected, the increase in the burstiness implies a higher conversion ratio to attain zero losses, with a large difference for the range of mid loads. Here again the *minH* policy shows large jumps when increasing the load, compared to the smooth behavior of the *minG* policy. In this case however we can compare the effect of the allocation policy versus that of the burstiness. From Figure 6.7 we see that when the load goes above 0.85 the conversion requirements for the *minH* policy under geometric arrivals becomes up to 50% higher than those for the *minG* policy under an ON-OFF arrival process with $\kappa = 5$. This is an important difference in conversion requirements and reveals a significantly worse performance of the *minH* policy even under non-bursty traffic.

We have seen that the *minG* policy requires significantly less conversion resources than its *minH* counterpart. Therefore, we now focus on the *minG* policy and analyze the effect of the number of FDLs under bursty traffic, as illustrated in Figures 6.8(a) and 6.8(b). Figure 6.8(a) shows this effect for a switch where the packet size is uniformly distributed between 5 and 15, and the granularity is 9. This is the same scenario as in Figure 6.5(b), the only difference being that now the arrival process is ON-OFF with $\kappa = 5$. A first comparison between the two figures yields the expected result of higher σ^* when the arrival process is ON-OFF instead of geometric. A more relevant observation is that the addition of FDLs under bursty traffic has a more significant effect than it had under geometric arrivals. For instance, adding a second or a third FDL produces a larger reduction on σ^* under bursty traffic. As well as in the geometric case, this difference vanishes when the load becomes sufficiently high, giving no advantage in placing additional FDLs. However, the range of loads for which placing additional FDLs makes a difference is larger in the



(a) σ^* for a switch under *minG* policy with an ON-OFF arrival process, $D = 9$ and packet size uniformly distributed between 5 and 15

(b) σ^* for a switch under *minG* policy with an ON-OFF arrival process, $D = 8$ and packet size equal to $\{5, 15\}$

Figure 6.8: Effect of the burstiness on σ^*

bursty traffic case. An example of this is the placement of the second FDL. While placing a second FDL makes no difference for loads above 0.83 under geometric IATs, it is worth doing so for loads up to 0.92 under bursty traffic. Therefore, the addition of FDLs under bursty traffic not only reduces the conversion requirements in a more significant manner than under non-bursty traffic, but this reduction is valid for a larger range of loads, increasing the value of additional FDLs.

We conclude by taking a final look at the effect of the FDLs on the minimum conversion ratio to attain zero losses. From Figure 6.8(b) it is evident that increasing the number of FDLs, in this case from one to three, has a significant effect on reducing σ^*. Furthermore, this reduction is as strong as to make the conversion requirements for the case with $N = 1$ and $\kappa = 1$ similar to those of the case with $N = 3$ and $\kappa = 5$. Consequently, it is possible, at least in part, to compensate the effect of the burstiness on σ^* by including additional FDLs, supporting a switching solution that combines both conversion and buffering resources to resolve contention, especially under bursty traffic.

Chapter 7

Limited-Range Conversion and Fiber Delay Lines

This chapter presents an approach to evaluate the performance of an optical switch equipped with a pool of both *limited-range* wavelength converters and Fiber Delay Lines *per port* to resolve contention. In order to study the performance of this system, we propose an analytical model that allows a general behavior for the packet-size distribution, while the inter-arrival times are assumed to be of Phase-Type (PH), a class of distributions that allows a wide variety of behaviors (see Appendix A.1). As indicated in Section 7.2.5, this assumption can be relaxed to allow for general inter-arrival times. In general, limited-range conversion is difficult to analyze because the interaction among the wavelengths depends on their specific location. Namely, the state of a particular wavelength is affected by and affects the behavior of its closest neighbors: if a packet arrives through a wavelength where it cannot be transmitted immediately nor buffered, it can be converted to a wavelength *in the range* allowed by the converter. This range is typically made of the wavelengths closest to the one in which the packet arrived. Therefore, to fully model a port with W wavelengths, it would be necessary to keep track of the state of each wavelength explicitly. With the addition of FDLs, keeping track of the state of each wavelength becomes infeasible even for moderate values of W.

In this chapter we start by proposing a model where the whole set of wavelengths is partitioned in smaller subsets that can be analyzed separately. In this manner it is not necessary to keep track of the state of the total number of wavelengths, but only of those in the subset. Although this subset is only of size two, it allows us to study the effect of different wavelength allocation policies on the packet loss rate. Moreover, we identify a linear association between the loss rate in this simple configuration and in the more complex case where a wavelength can make use of its two closest neighbors to convert a packet that it cannot buffer. This approach works well for different configurations and is particularly useful for the mid load case, when simulations become computationally expensive. The results show that, given the restriction in the wavelength range in which an incoming packet can be converted, even a few FDLs help to significantly reduce packet losses by exploiting the time domain.

This chapter starts by presenting the switch architecture and the different wavelength allocation policies in Section 7.1. Next, Section 7.2 describes the analytical model for the simpler case, while Section 7.3 reports several results about the effect of the policies and other parameters in the performance of the switch. Section 7.4 presents the approximation for the more complex case as well as results related to its behavior.

7.1 Switch architecture

The optical switch analyzed in this chapter has a set of K input and output ports, as shown in Figure 7.1. Packets arrive and leave the switch through any of the W wavelengths in each port. A set of converters is attached to each input port, while each output port has its own set of FDLs, in a similar fashion as the space switch described in [94, Chapter 10]. Similarly to the architectures considered in the previous chapters, we assume a synchronous operation, making the switching matrix design simpler [28], and variable packet sizes, reducing the amount of header processing [27]. We also assume that the traffic is uniformly distributed among the output ports. Thus we can evaluate the performance of a single port since all of them behave in a similar fashion. When a packet is destined to a specific output port, it comes through one of the W wavelengths (in an input port), referred to as its home wavelength. As mentioned before, we assume that the converters provide limited-range wavelength conversion, namely they are able to convert to a specific subset of the output wavelengths only [127]. This set is usually made of the home and some adjacent wavelengths [106]. Thus if a packet cannot be transmitted nor buffered using its home wavelength it must be allocated in another wavelength among those that are reachable from its home wavelength.

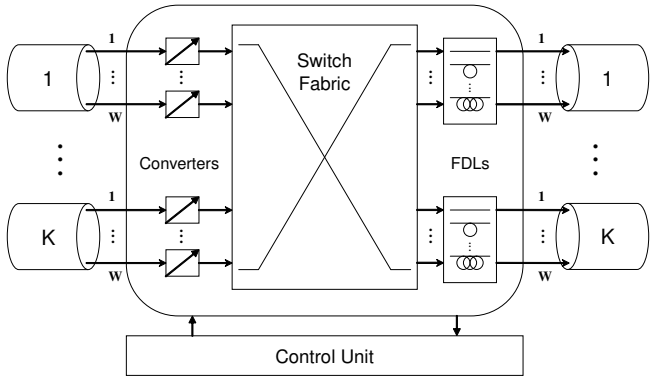


Figure 7.1: Switch architecture with K input/output fibers, W wavelengths, converters and FDLs

The selection of a wavelength for transmission depends on the state of each wavelength in the reachable set. The state of a wavelength can be described by the scheduling horizon, i.e., the time required to transmit all the packets already allocated (being transmitted or buffered) in this wavelength. Recall that an FDL buffer creates gaps in the channel and

these gaps have a relevant influence on the switch performance since the channel will be idle for several intervals even when some packets are waiting for transmission in the FDL. Therefore when a packet is allocated to a particular wavelength, this choice implies both the delay that the packet must face and the gap that it creates on the channel [28].

For the case of limited-range wavelength conversion, the wavelength allocation decision is made over the restricted set of reachable wavelengths, called the output set. We consider two different alternatives to compose the output set:

- Symmetric set: the output set includes the home wavelength and d wavelengths on either side of it [127]. This gives a set of variable size, depending on the position of the home wavelength. For those wavelengths that are at least d positions separated from the first and the last wavelength, the set is of size $2d+1$. For the wavelengths on the borders the output set is of size $d+1$ only. In general, if the wavelengths are numbered from 1 to W, where the first and the last are the borders, the output set of wavelength i is

  $$\{\max\{i-d,1\},\max\{i-d+1,1\},\ldots,i,\ldots min\{i+d-1,W\},min\{i+d,W\}\},$$

  for $i=1,\ldots,W$. The parameter d is called degree of conversion [95].

- Fixed set: in this case the set of wavelengths is partitioned, such that a packet can only be forwarded through the wavelengths that belong to the same partition as its home wavelength. The partition is assumed to be built by adjacent wavelengths since the converters perform faster for a small range of wavelengths.

Once the output set is determined, the selection of the wavelength used for transmission can be made in several ways:

- Random: the wavelength used to transmit the packet is selected randomly among those in the output set where the required delay is not greater than ND.

- Minimum Horizon (*MinH*): the packet is sent through the wavelength that offers the minimum scheduling horizon, i.e., the one where the delay before transmission is minimum.

- Minimum Gap (*MinG*): the wavelength selected for transmission is such that the gap created by the scheduling of the packet is minimum among the set of output wavelengths.

The two latter scheduling policies have been studied in [27,28] for the case of full-range wavelength conversion using simulation models. It must be noted that in those works, as well as here, we assume that whenever a conversion is required, a converter will be available to perform it (full wavelength conversion). In the next section we present a model for the fixed-set case where the wavelengths are allocated according to the *MinH* or *MinG* policies.

7.2 Analytical model for two wavelengths

In this section we assume that the W wavelengths are partitioned in $\frac{W}{2}$ subsets each of size 2. In principle, the model can be generalized to a partitioning with $\frac{W}{k}$ subsets each of size k, with k being a divisor of W. Nevertheless we will restrict ourselves to the two-wavelength case to limit the computation times. Indeed, the size of the model for realistic parameter values is a key issue since the state space grows exponentially if k is chosen larger than two. Nevertheless, the two-wavelength case captures to some extent the benefits of having multiple wavelengths since it includes the effect of the wavelength allocation policy. A first approach to model this system is to generalize the horizon model presented in [73, 116] for the single-wavelength buffer. In this case, each wavelength is represented by the scheduling horizon seen by the incoming packet, that is, the time required until all packets scheduled on this wavelength have left the system. If this horizon is greater than ND, the maximum achievable delay, on all wavelengths, the packet is dropped. Even for the case with two wavelengths this approach becomes problematic in terms of the required computation times for realistic parameter values. Another approach is the one proposed in [103], where the state variable is the waiting time of the last accepted packet, using a modified version of the Lindley equation. Although this method provides a much smaller state space, it is only directly applicable for the single-wavelength system.

In order to keep the model size numerically tractable, we propose a model that mixes the two approaches mentioned above by observing the system only when an incoming packet is accepted. Whenever a packet arrives to the FDL it is accepted for transmission if at least one of the wavelengths is able to delay the packet until this wavelength becomes idle. Thus, the system can be represented with two state variables: W_n, the waiting time of the n-th accepted packet, for $n \geq 1$; and H_n, the value of the scheduling horizon of the wavelength that did not admit the n-th accepted packet at its arrival, for $n \geq 1$. As the FDL only provides delays that are multiples of D, the waiting time W_n only takes values in the set $\{0, D, 2D, \ldots, ND\}$. The cardinality of this set is much smaller than that of the scheduling horizon H_n, since this variable can adopt values in the set $\{0, 1, 2, \ldots, ND + L_{\max} - 1\}$, where $L_{\max}$ is the maximum packet size.

For the policies studied later, the sequence $\{\{W_n, H_n\}, n \geq 1\}$ is a Markov chain as will be clear from the evolution equations in each case. With the combination of the waiting time and the scheduling horizon as state variables, the state space is much smaller than keeping track of both horizons as state variables. Moreover the state space of the combined variables is not equal to the cartesian product of the sets described above since many of the possible combinations are not reachable. In fact the reachable state space highly depends on the wavelength allocation policy. It is important to remark that the same modeling approach can be used for $k > 2$. This representation would require one delay variable and $k - 1$ horizon variables, resulting in a huge state space even for $k = 3$.

The packet arrival process to a fiber is characterized through two sequences of i.i.d. random variables: $\{T_n, n \geq 1\}$ is the time between the arrival of the n-th packet and the next one; and $\{L_n, n \geq 1\}$ is the size of the n-th packet. In this chapter we assume that $\{L_n, n \geq 1\}$ follows a general discrete distribution with finite support and $\{T_n, n \geq 1\}$ follows a DPH distribution with parameters (β, S) (see Appendix A.1), an assumption

that can be easily relaxed to admit any general discrete distribution as well (see Section 7.2.5). We also assume that the arrivals destined to a tagged output fiber are uniformly distributed among all the wavelengths. Thus the arrival process to a fixed set of two wavelengths is a thinned DPH arrival process, since with probability $\frac{2}{W}$ a packet comes through one of the wavelengths in the set. With this assumption the set of two wavelengths can be analyzed in isolation and the inter-arrival times to it can be described as a sequence of i.i.d. variables $\{I_n, n \geq 1\}$ that follow a DPH distribution with parameters (α, T), with $\alpha = \beta$ and

$$T = S + \left(1 - \frac{2}{W}\right) s\beta.$$

Even though we exploit the flexibility of DPH distributions for the inter-arrival times, it is important to note that the state space size is independent of the number of phases of the DPH variable. This becomes clear in the next subsections where we present the evolution equations of the system, depending on the rule used for wavelength allocation.

7.2.1 Minimum Horizon policy

When the next packet arrives after the n-th accepted packet, it will find that the horizons of the wavelengths are equal to $[H_n - I_n]^+$ and $[W_n + L_n - I_n]^+$, where $[x]^+$ stands for $\max(x, 0)$. If at least one of these horizons is less than or equal to ND, the packet will be accepted in the wavelength with the minimum horizon value. Thus the waiting time of the arriving packet will be equal to the smallest multiple of D larger than or equal to the horizon of the selected wavelength. The horizon value of the other wavelength will remain identical. The evolution equations that describe this process are

$$H_{n+1} = \max\{[H_n - I_n]^+, [W_n + L_n - I_n]^+\}, \tag{7.1}$$

$$W_{n+1} = \left\lceil \frac{min\{[H_n - I_n]^+, [W_n + L_n - I_n]^+\}}{D} \right\rceil D. \tag{7.2}$$

If the arriving packet sees that both horizons are above ND, the maximum delay the FDL offers, the packet must be dropped. In that case the evolution of the system is given by

$$H_{n+1} = \max\{[H_n - \tilde{I}_n]^+, [W_n + L_n - \tilde{I}_n]^+\},$$

$$W_{n+1} = \left\lceil \frac{min\{[H_n - \tilde{I}_n]^+, [W_n + L_n - \tilde{I}_n]^+\}}{D} \right\rceil D,$$

where $\tilde{I}_n$ is the time until the next accepted packet, which has a different distribution than I_n, as will be explained in subsection 7.2.3.

7.2.2 Minimum Gap policy

Under the *MinG* policy the incoming packet is assigned to the wavelength in which the gap generated by accepting the packet is minimum, in case both wavelengths are able to

accept the incoming packet. Let G_n^1 be the gap generated if the packet is accepted by the wavelength that did *not* accept the last packet, which is given by

$$G_n^1 = \left\lceil \frac{[H_n - I_n]^+}{D} \right\rceil D - [H_n - I_n]^+.$$

Equivalently, let G_n^2 be the gap generated if the packet is accepted by the wavelength used by the previous accepted packet, that is

$$G_n^2 = \left\lceil \frac{[W_n + L_n - I_n]^+}{D} \right\rceil D - [W_n + L_n - I_n]^+.$$

If $G_n^1 < G_n^2$ the packet will be sent to the wavelength that did not receive the last packet, causing the new values of the state variables to be

$$H_{n+1} = [W_n + L_n - I_n]^+ \quad \text{and} \quad W_{n+1} = \left\lceil \frac{[H_n - I_n]^+}{D} \right\rceil D.$$

If $G_n^1 > G_n^2$ the packet will be sent to the wavelength that was also used by the previous accepted packet, making the variables evolve as

$$H_{n+1} = [H_n - I_n]^+ \quad \text{and} \quad W_{n+1} = \left\lceil \frac{[W_n + L_n - I_n]^+}{D} \right\rceil D.$$

In case both potential gaps have the same value or if only one of the wavelengths is able to receive the packet, the evolution follows Equations (7.1) and (7.2). If the next arriving packet has to be dropped due to the value of the horizons (both above ND) the evolution will follow the same equations already shown in this section, but using variable $\tilde{I}_n$ instead of I_n to describe the time until the next accepted packet. The distribution of this variable is addressed in the next subsection.

7.2.3 Distribution of the time until the next accepted packet

As defined above, the inter-arrival times I_n can be described by i.i.d. DPH variables with parameters (α, T). This implies that the arrival process is a DPH renewal process [76] with renewal density $r(\cdot)$ given by

$$r(k) = \alpha(T + t\alpha)^{k-1}t, \quad k \geq 1. \tag{7.3}$$

The density $r(k)$ gives the probability of having an arrival at slot $n + k$ given an arrival in slot n (either with or without arrivals in between). When the system is in state (W_n, H_n) it may not be able to accept a packet arriving in the next time slot, i.e., $\min\{H_n, W_n + L_n\} - 1 > ND$, in which case we use a different inter-arrival distribution to take into account the period of time in which the channel is unavailable. For this we first define $K_n = \min\{H_n, W_n + L_n\} - ND - 1$ as the number of slots required by the system to have a horizon equal to ND after the arrival of the n-th accepted packet. Using this quantity and Equation (7.3) we can define the probability distribution of the time until the arrival of the next accepted packet $\tilde{I}_n$ as

$$P(\tilde{I}_n = k) = \alpha(T + t\alpha)^{K_n - 1}T^{k - K_n}t, \quad k \geq K_n.$$

In this expression the arrival process restarts after the arrival of the last admitted packet. A new phase is selected with probability mass α. Then the system enters an unavailability period of length $K_n - 1$ where every arriving packet is dropped. This period is followed by $k - K_n$ slots without arrivals, after which the arrival process goes to absorption generating an arrival in the next time slot.

7.2.4 Loss rate

As the model only keeps track of the accepted packets, the loss rate (LR) can be computed from the expected number of losses generated when a new packet is accepted. The expected losses generated by the last admitted packet is computed as a weighted sum of the expected loss in each state with the stationary probability distribution as the weights. In a state where $W_n = W$, $H_n = H$ and the last accepted packet was of size B, the expected number of losses is equal to the expected number of arrivals in a time interval of length $[\min\{W + L, H\} - ND - 1]^+$. This value represents the number of time slots required by the FDL before it is able to accept a new packet.

The expected number of arrivals (A) in a time interval of length M, denoted by $E[A|M]$, depends on the inter-arrival distribution. For the case of geometric inter-arrival times with parameter p, the expected number of arrivals is the expected value of a binomial distribution with parameters p and M, that is pM. For the case of DPH IATs with parameters (α, T), we make use of the renewal density $r(k)$, defined in Equation (7.3). As $r(k)$ is the probability of an arrival at slot k for $k \geq 1$, the expected number of arrivals in an interval of length M can be computed as

$$E[A|M] = \sum_{l=1}^{M} r(l) = \sum_{l=1}^{M} \alpha(T + t\alpha)^{l-1}t = \alpha(I - (T + t\alpha)^M)(I - (T + t\alpha))^{-1}t. \quad (7.4)$$

Let π_{ij} be the stationary probability distribution that an arbitrary accepted packet had to wait i time slots and the scheduling horizon of the wavelength that did not admit that packet is equal to j, for all possible states (i, j) in the state space Ω. Also let Ξ be the support of the sequence $\{L_n, n \geq 1\}$, and b_k be the probability mass at point k, for $k \in \Xi$. Then the expected number of losses generated by the last accepted packet is given by

$$\begin{aligned} E[Loss] &= \sum_{(i,j)\in\Omega} \pi_{ij} \sum_{k\in\Xi} b_k E[A|[\min\{i + k, j\} - ND - 1]^+] \\ &= \sum_{(i,j)\in\Omega} \pi_{ij} \sum_{k\in\Xi} b_k \sum_{l=1}^{[\min\{i+k,j\}-ND-1]^+} \alpha(T + t\alpha)^{l-1}t. \end{aligned}$$

Finally, as every accepted packet generates on average a number of losses equal to $E[Loss]$, the loss rate of the system is given by

$$LR = \frac{E[Loss]}{E[Loss] + 1}.$$

7.2.5 General inter-arrival times

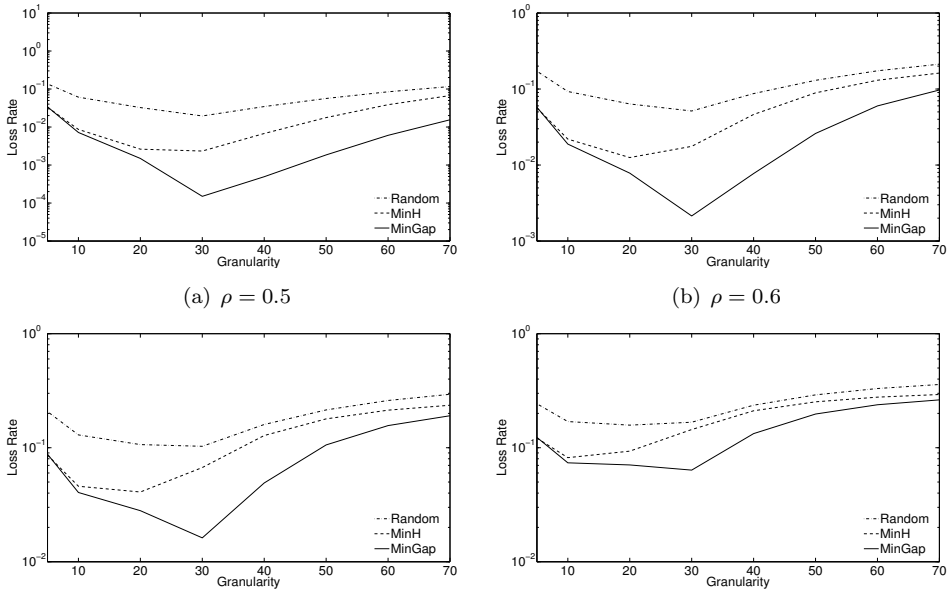
The case of general inter-arrival times can be dealt with in the same way as described above for the DPH case. Let f be the probability mass function of the inter-arrival times. By conditioning on the last arrival before slot k, the renewal density can be defined as

$$r(k) = f(k) + \sum_{j=1}^{k-1} f(j)r(k-j), \quad k \geq 1. \tag{7.5}$$

This function can be recursively evaluated for any finite k starting with $r(1) = f(1)$. Using the renewal density, the distribution of the time until the arrival of the next accepted packet is given by

$$P(\tilde{I}_n = k) = f(k) + \sum_{j=1}^{K_n-1} f(k-j)r(j), \quad k \geq K_n.$$

This expression accounts for an unavailability period of length $K_n - 1$, after which the next packet arrives at time k. Finally the computation of the LR only requires the determination of the expected value of arrivals in an interval of arbitrary length M, defined above as $E[A|M]$. This can be done in a similar way as in Equation (7.4), by simply summing the values of the density, that is $E[A|M] = \sum_{l=1}^{M} r(l)$. Using this setting it is possible to deal with general inter-arrival times. However, from a practical point of view, the general process offers little additional value, as the DPH class includes any general distribution with finite support.



(a) $\rho = 0.5$ (b) $\rho = 0.6$

(c) $\rho = 0.7$ (d) $\rho = 0.8$

Figure 7.2: Loss Rate for Fixed Output Set with $B = 30$, $N = 5$ and geometric IATs

7.3 Comparison among policies

In this section we present several results about the performance of the switch with a fixed output set of size two. Here we compare the three policies introduced in Section 7.1: *Random*, *MinH* and *MinG*. The comparison is always done in terms of the LR as a function of the granularity, since the former is the main measure of performance and the latter is the most critical design parameter for the FDL buffer. Regarding the other parameters, all the results are for a switch with 32 wavelengths with a load ranging from 50% to 80%. The number of FDLs varies from 1 to 7, since the length of the longest fiber must be kept short enough to be implementable. For the packet size we use three different distributions: the first one assumes a fixed packet size equal to 30 slots; the second has two equiprobable values: 10 and 50 slots; and the third one is a uniform distribution between 20 and 40 slots. We use two different inter-arrival distributions for the comparisons: one is the simple geometric distribution, while the other is a mixture of geometric distributions. The probability mass function of a geometric random variable X with parameter p is given by

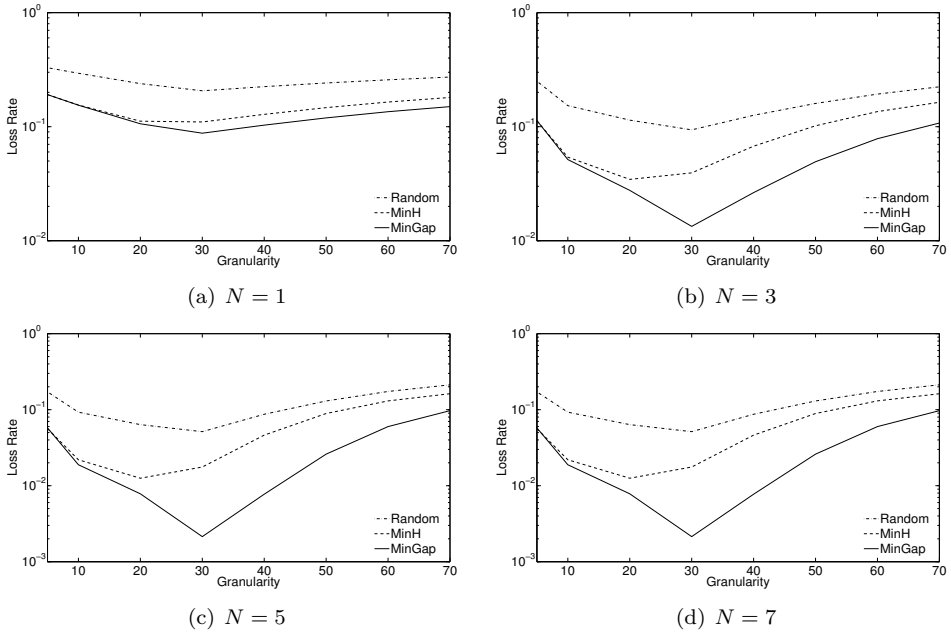
$$P(X = k) = (1 - p)^{k-1}p, \quad k \in \{1, 2, \dots\},$$

while the one of a random variable Y representing the mixture of two geometric variables with parameters p_1 and p_2 is given by

$$P(Y = k) = \alpha_1(1 - p_1)^{k-1}p_1 + \alpha_2(1 - p_2)^{k-1}p_2, \quad k \in \{1, 2, \dots\},$$

where α_1 and α_2 are the mixing probabilities. We use this distribution to analyze the case of highly variable inter-arrival times by setting the squared coefficient of variation (SCV) equal to 5.

It should be clear that the set of possible combinations of parameter values is too large to be presented exhaustively. Therefore we concentrate separately on the effect of each of these parameters on the performance of the switch. In Figure 7.2 we show the switch LR under the three policies, fixing the number of FDLs equal to 5 and varying the load from 50% to 80%. In Figure 7.3 the load is equal to 60%, while the number of FDLs increases from 1 to 7. In both cases the IATs follow a geometric distribution. In these figures we clearly observe the performance gain obtained by using the *MinH* or the *MinGap* policies over the simpler *Random* rule. Although the output set is made of two wavelengths only, the use of the information about the state of the buffer results in a significant reduction in the number of losses. In particular, *MinGap* shows a consistent better performance than the other policies, and the optimal granularity is close to the value of the packet size. The difference among the minimum LRs reached by the policies diminishes as the load increases, but the optimal granularity is more robust for the *MinGap* policy than for the others. Therefore this policy attains the minimum LR over a broad range of loads with the same granularity. From Figure 7.3 the effect of the number of FDLs becomes clear. In all cases, *MinH* and *MinGap* outperform the simpler *Random* policy, but the differences become more evident as the number of FDLs increases. Particularly the *MinGap* policy realizes an important performance difference as the number of FDLs increases, indicating a better use of the buffering resources.



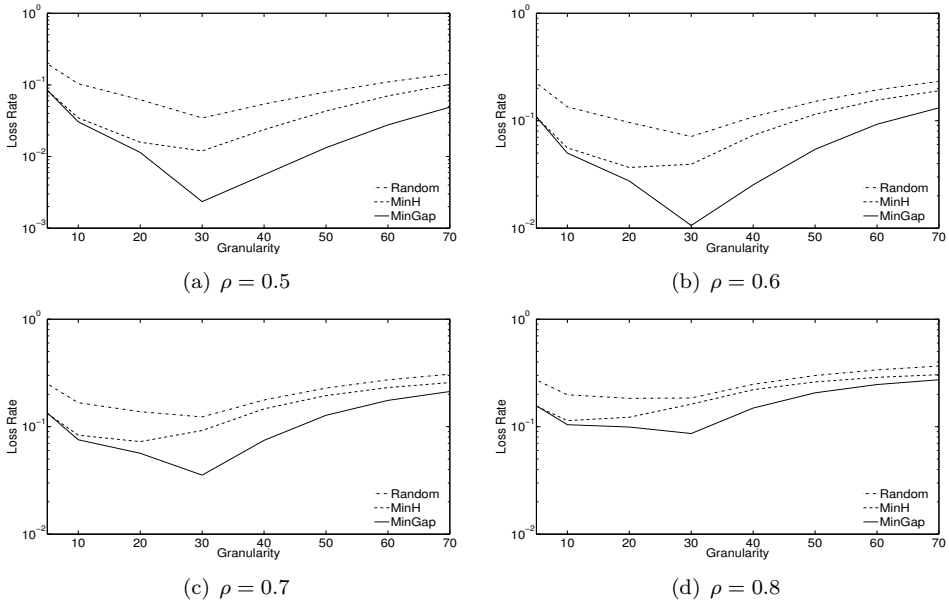
(a) $N = 1$ (b) $N = 3$

(c) $N = 5$ (d) $N = 7$

Figure 7.3: Loss Rate for Fixed Output Set with $B = 30$, $\rho = 0.6$ and geometric IATs

Similar results for the case of a highly variable arrival process can be seen in Figure 7.4. Also in this case, when the load increases the shapes of the curves change and the optimal granularity diminishes in value. Again the *MinGap* policy shows the best performance and the most robust behavior in relation to the optimal granularity. When comparing these results with those in Figure 7.2 it is clear that the curves keep the same shape, but the losses are larger in the more variable case. For mid loads the difference becomes as large as one order of magnitude, implying an important effect of the arrival process variability. However the difference between the geometric and the high variable cases narrows as the load increases. This means that for high loads the effect of the inter-arrival SCV is not as important as for the mid load case.

In Figure 7.5 we fix both the load and the number of FDLs in order to focus on the effect of the packet size distribution and the variability of the arrival process. Irrespective of the packet size distribution, higher variability causes an increase in the losses while the optimal granularity remains in the same region. For all the scenarios the *MinGap* policy outperforms the other ones, but the difference is smaller when the arrival process shows high variability. When the packet size can take values 10 or 50 with equal probability, the optimal granularity is located around the larger value, but there is a large region with an LR close to the optimal. If the packet size follows a uniform distribution between 20 and 40, the optimal granularity is around the expected value. Also in this case there is a set of possible values for the granularity with a performance close to optimal. As the value of the optimal granularity diminishes as the load increases, this parameter can be chosen such that it performs almost optimally for both mid and high loads.



(a) $\rho = 0.5$ (b) $\rho = 0.6$

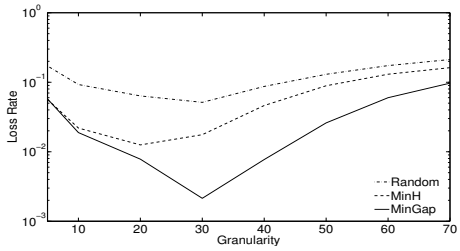
(c) $\rho = 0.7$ (d) $\rho = 0.8$

Figure 7.4: Loss Rate for Fixed Output Set with $B = 30$, $N = 5$ and IATs with SCV equal to 5

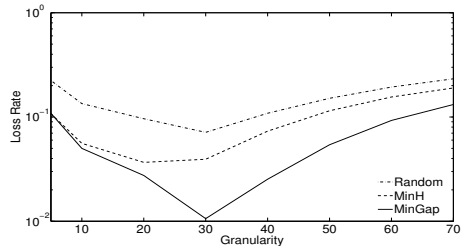
7.4 An Approximation for the Symmetric Set

As described in Section 7.1, the set of reachable wavelengths in the *symmetric* case is made of the adjacent wavelengths and the home wavelength itself. This configuration is particularly difficult to model analytically since the set of servers (output wavelengths) overlaps for the different queues (input wavelengths) in the system. Therefore it is not possible to isolate a subset of servers and queues, as was done for the fixed set case, since any subset of servers is affected by the queues of the adjacent servers. Furthermore the packets entering through the first and last wavelengths have less alternatives since the output set is smaller than for the central wavelengths. In order to analyze this system we performed several simulations focused on the *MinGap* policy since this policy performs the best among those already analyzed. The estimates of the LR were computed using the batch means method [77] and removing the effect of initial conditions by ignoring the warm-up period. The confidence intervals' half width is at most 2% of the mean. These simulations become very expensive when the LR is very small, since the number of events required to compute a good estimate can be computationally prohibitive.

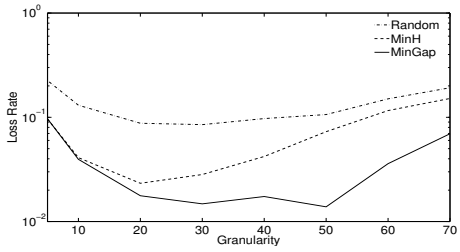
When comparing the results of the simulation for the symmetric system and the analytic model for the fixed output set, we found that the behavior (shape) of the LR as a function of the granularity is similar. This holds in particular for the fixed case with output sets of two wavelengths and the symmetric case with $d = 1$, for many different configurations. More specifically, we found a strong linear association between the natural logarithm of the LR for the fixed case and its symmetric counterpart. This is illustrated in Figure 7.6 where each point is the combination of the logarithm of the LR for the fixed



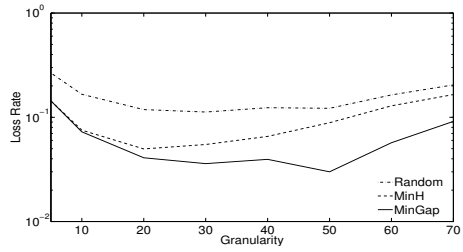
(a) $L = 30$ - Geo



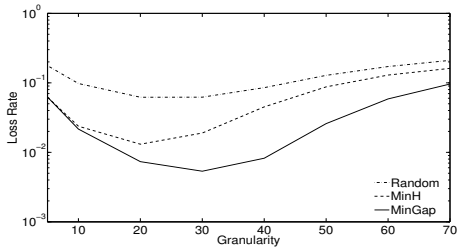
(b) $L = 30$ - SCV $= 5$



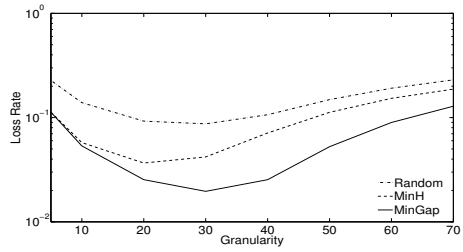
(c) $L = \{10, 50\}$ - Geo



(d) $L = \{10, 50\}$ - SCV $= 5$

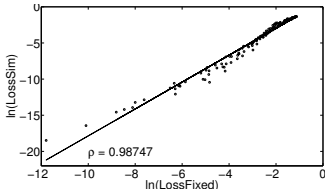


(e) $L \sim$ Unif$(20, 40)$ - Geo

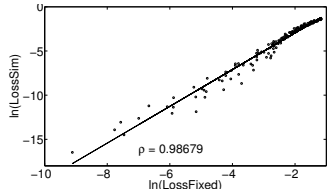


(f) $L \sim$ Unif$(20, 40)$ - SCV $= 5$

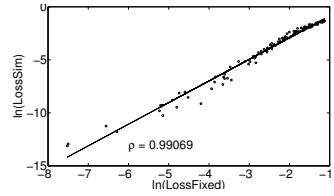
Figure 7.5: Loss Rate for Fixed Output Set with $N = 5$ and $\rho = 0.6$



(a) $L = 30$



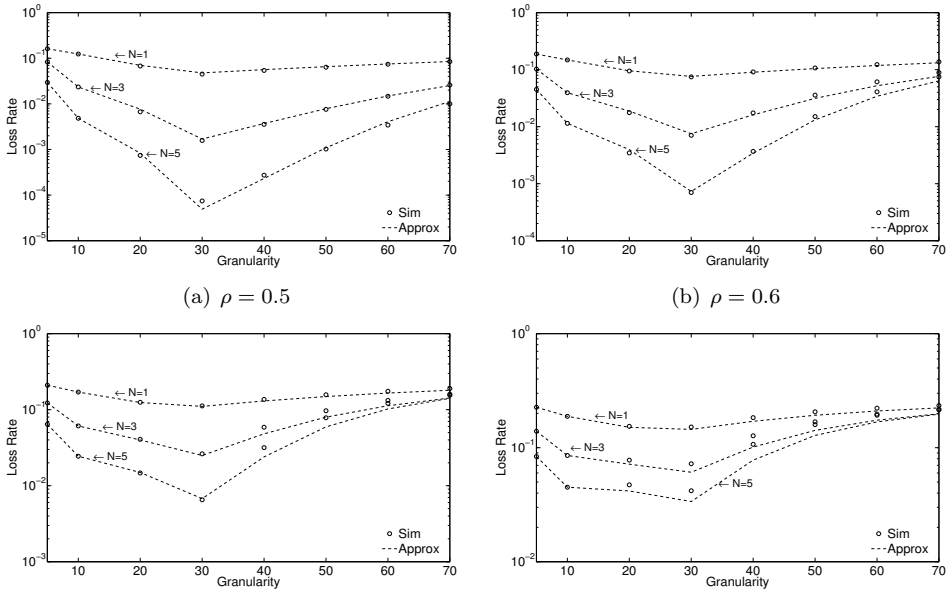
(b) $L = \{10, 50\}$



(c) $L \sim$ Unif$(20, 40)$

Figure 7.6: Linear relation between the logarithms of the Loss Rates of the Symmetric ($d = 1$) and the Fixed output sets

case and the same value for the symmetric case. The results are separated according to the packet size distribution and the scenarios include different values for the granularity (between 5 and 70), the number of FDLs (from 1 to 5), the load (between 50% and 90%) and the SCV of the arrival process (geometric case and SCV equal to 5). We include in each figure the coefficient of linear correlation, which is very high in all cases.



(a) $\rho = 0.5$ (b) $\rho = 0.6$

(c) $\rho = 0.7$ (d) $\rho = 0.8$

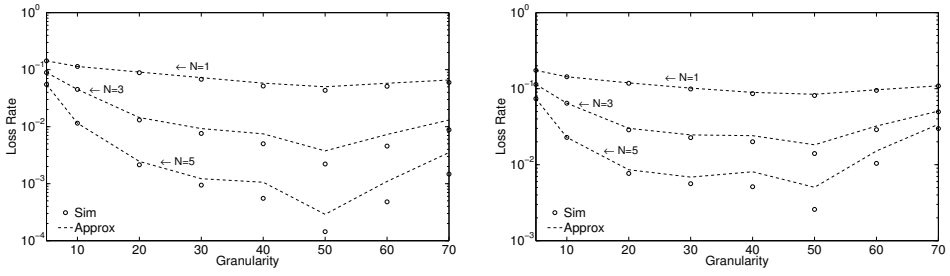
Figure 7.7: Approximation and simulation of the symmetric output set for $W = 32$, $L = 30$ and IATs with SCV equal to 5

This behavior suggest an approximation based on both the analytical model and on simulations, which we can apply for the parameters' range mentioned above. Given a specific configuration (number of FDLs, load, inter-arrival and packet size distribution) we propose to simulate the switch for the symmetric case for two different values of the granularity and to estimate the loss rate as follows. We use the logarithms of the LRs for these two cases (say y_1 and y_2) and compute the same results for the fixed case (x_1 and x_2) to estimate the parameters β_0 and β_1 of the approximate linear equation that relates these two quantities. These are given by

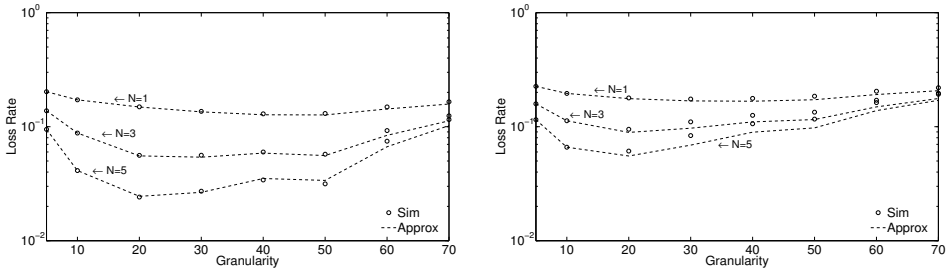
$$\beta_1 = \frac{y_1 - y_2}{x_1 - x_2}, \quad \beta_0 = y_1 - \beta_1 x_1.$$

Let LR_f and LR_s be the loss rate for the fixed and the symmetric cases, respectively. Then we can approximate the loss rate in the symmetric case using the relation $LR_s = \exp\{\beta_0 + \beta_1 \ln(LR_f)\}$. Hence we can approximate the LR for the symmetric case using the LR for the fixed case obtained with the analytic model and the estimated linear equation. It must be noted that the values of the granularity for the simulations are selected such that the LR estimates for the symmetric case can be computed fast. Even though the simulations are necessary for the proposed approximation, the time required to compute the approximated LR for the symmetric case is significantly smaller than simulating the system for each possible value of the granularity. In fact, this last option may be infeasible when the LR becomes very small.

Figures 7.7, 7.8 and 7.9 show the results of this approximation, one for each packet-size

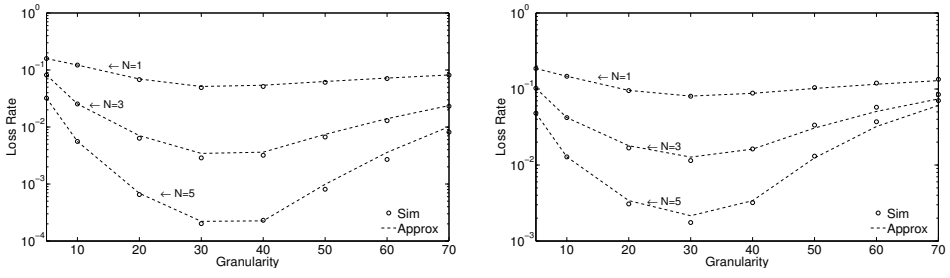


(a) $\rho = 0.5$ (b) $\rho = 0.6$

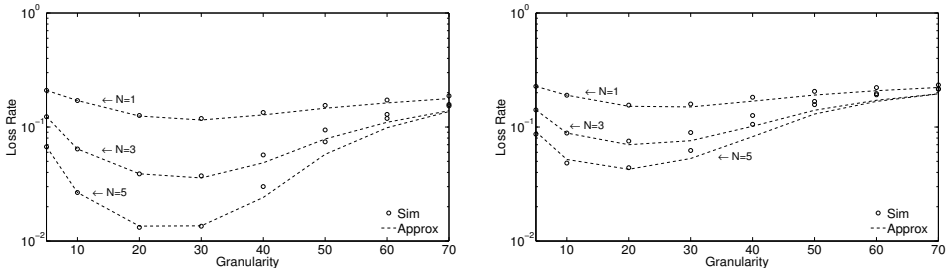


(c) $\rho = 0.7$ (d) $\rho = 0.8$

Figure 7.8: Approximation and simulation of the symmetric output set for $W = 32$, $L = \{10, 50\}$ and IATs with SCV equal to 5

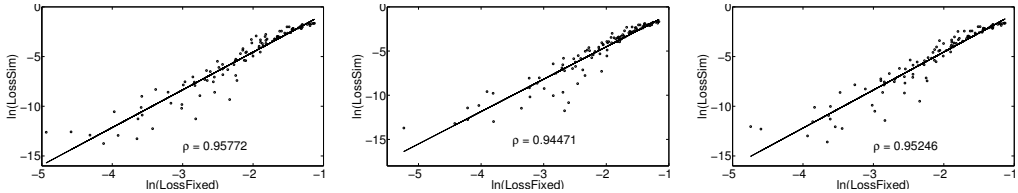


(a) $\rho = 0.5$ (b) $\rho = 0.6$



(c) $\rho = 0.7$ (d) $\rho = 0.8$

Figure 7.9: Approximation and simulation of the symmetric output set for $W = 32$, $L \sim Unif(20, 40)$ and IATs with SCV equal to 5



(a) $L = 30$ (b) $L = \{10, 50\}$ (c) $L \sim \text{Unif}(20, 40)$

Figure 7.10: Linear relation between the logarithms of the Loss Rates of the Symmetric ($d = 2$) and the Fixed output sets

distribution considered in Section 7.3. All the results in these figures assume an inter-arrival time distribution with SCV equal to 5, but similar results were obtained for other coefficients of variation and for the geometric case. To approximate the linear line that relates the logarithms of the LRs we use the results of the simulation with the granularity set equal to 5 and 10, since for these values a good estimate of the LR can be obtained fast. As can be seen in Figure 7.7, the approximation works adequately for the different configurations shown. The approximation is closer to the actual value for the case of mid loads and a small number of FDLs. It must be noted that the approximation is especially useful for the mid-load case to avoid prohibitive simulation times to compute a very small LR. For the case of high loads the approximation becomes somewhat pessimistic, a result that holds for the different packet size distributions, as well as for larger loads than those shown here.

Among the different configurations we tried, those with a higher variability in the packet size distribution show the worst performance of the approximation. This is the case in Figure 7.8, where for loads 50% and 60% the approximation follows the shape of the simulated LR, but around its minimum value the loss is overestimated. For the fixed and uniformly distributed packet size, not only the shape of the approximation and the simulation agree, but also the values of the approximated LR are very close to those obtained through simulations. Even though this approximation requires to setup and run simulations, the parameters can be chosen such that reliable estimates of the LR can be computed in a short run. Additionally we have found that the linear relation that supports this approximation also holds between the LRs of the fixed output set and the symmetric one with $d = 2$, as can be seen in Figure 7.10. Even though the linear correlation is smaller in this case, the approach introduced here can still provide a good approximation for the LR in the symmetric case.

Appendices

A.1 Markovian distributions and point processes

In this appendix we provide a few definitions regarding Phase-Type (PH) distributions and Markovian Arrival Processes (MAPs), which are the building blocks used to describe the stochastic behavior encountered in many of the systems considered in this thesis. Starting from these blocks, one can model the evolution of a system by means of Markov chains (MCs), in order to compute transient and stationary measures, with a particular interest on the stationary probability distribution. For an introduction to Markov chains we refer to [29, 65, 66, 70].

A.1.1 Phase-Type distributions

A continuous Phase-Type (CPH) distribution describes the time until absorption in a continuous-time MC (CTMC) with state space $\{0, 1, ..., m\}$, where the states $\{1, \ldots, m\}$ are transient and the state 0 is absorbing. The initial probability distribution of this chain is given by the $(m+1) \times 1$ vector $[\alpha_0, \alpha]$, while its $(m+1) \times (m+1)$ generator matrix is

$$Q = \begin{bmatrix} 0 & 0 \\ t & T \end{bmatrix}.$$

Here T is the subgenerator matrix describing the transitions among the transient states, and the i-th entry of t holds the absorption rate into state 0 from state i, for $1 \leq i \leq m$. For the matrix Q to be a proper generator matrix the off-diagonal entries of T and all the entries of t must be nonnegative, while the diagonal entries of T must be negative and such that Q has zero row-sum. Therefore, the vector t is given by $t = -Te$, where e is a column vector of ones. Also, for the vector $[\alpha_0, \alpha]$ to be a proper distribution its entries must be nonnegative and their sum must equal one. As a result, α_0 must be equal to αe, and the CPH distribution is fully determined by the parameters (m, α, T), where the parameter m is referred to as the order of the distribution. The cumulative distribution function (CDF) of a CPH distribution is given by $F(x) = 1 - \alpha \exp(Tx)e$, for $x \geq 0$. Also, its k-th moment is given by $k!\alpha(-T)^k e$, for $k \geq 1$. A similar case can be made for discrete PH (DPH) distributions, which describe the time until absorption in a discrete-time MC (DTMC). In this case the distribution is fully described by the parameters (m, α, T), where m is a positive integer, α is the $1 \times m$ vector holding the

initial probability distribution on the m transient states, and T is the $m \times m$ sub-stochastic matrix holding the one-step transition probabilities among the transient states.

PH (both CPH and DPH) distributions were first introduced by Neuts [89] and have been in use since then for two main reasons: first, they allow us to model more general behaviors than the simple exponential (or, in discrete time, geometric) distribution; second, they can still be used as part of Markovian models, which are computationally tractable. Also, their use has benefited from the development of statistical techniques to find the parameters of a PH distribution that matches some characteristics of a data trace. These techniques can be split in two groups: maximum-likelihood methods [9, 101, 112], and moment-matching methods [21, 62, 63, 93, 120]. Some of the moment-matching methods will be used in our experiments to obtain a PH distribution with a particular set of moments. A detailed treatment of PH distributions can be found in [76, 91].

A.1.2 The Markovian Arrival Process

With PH distributions we can model general *independent* inter-arrival times (IATs), giving rise to a PH renewal process. Notwithstanding, one can go one step further to introduce correlation while keeping the Markovian tractability. This can be achieved by using the Markovian Arrival Process (MAP) [81, 92], first introduced by Neuts [90]. A MAP(m, D_0, D_1) is a point process driven by an underlying CTMC with $m \times m$ generator matrix $D = D_0 + D_1$. The (i, j)-th entry of the matrix D_1 holds the rate at which, when the underlying chain is in state i, a customer arrives and the chain makes a transition to state j, for $1 \leq i, j \leq m$. The off-diagonal entries of the matrix D_0 hold the rates related to transitions without arrivals, and its (negative) diagonal entries are such that $De = 0$, where 0 is a column vector with all its entries equal to 0. Therefore, this process results from a CTMC whose transitions are marked [52] with either of two labels: the label '0' for the transitions that generate no arrivals, and the label '1' for the transitions that trigger an arrival. The matrices D_0 and D_1 hold the rates associated with the transitions labeled '0' and '1', respectively. These markings can be generalized to include other information about the arriving customers. As a first generalization let us consider the case where the arrivals occur in batches, and therefore the markings can be used to describe the number of arrivals generated by a single transition. In this case the arrival process is characterized by the $m \times m$ matrices $\{D_0, D_1, \ldots D_{\bar{L}}\}$, where $\bar{L}$ is the maximum batch size. The matrix D_j holds the rates related to transitions of the underlying chain that trigger a batch arrival of size j, for $1 \leq j \leq \bar{L}$. Since in this case the markings are related to the batch size only, this process is called Batch Markovian Arrival Process (BMAP) [81]. In general, this process is able to model correlation between the IATs and the batch size distribution.

Another feature that can be captured with the use of markings is the existence of different types of customers. For instance, there could be two types of customers, whose IATs and batch sizes are correlated. To deal with this, the markings must include not only the size of the batch but also the type of the customers in that batch. We assume that each batch is made of customers of only one type and the maximum size of a batch of customers of type one (resp. two) is $\bar{L}_1$ (resp. $\bar{L}_2$). Hence this arrival process can be modeled as a BMAP[2] characterized by the matrices $\{D_0, D_1^{j_1}, D_2^{j_2}, 1 \leq j_1 \leq \bar{L}_1, 1 \leq j_2 \leq \bar{L}_2\}$. The

notation BMAP[2] reflects the fact that the markings in this case describe both the batch size and the type of the customers in the batch (two possible types). In this case the matrix D_1^j (resp. D_2^j) holds the transition rates associated with the arrival of a batch of j type-one (resp. type-two) customers, for $1 \leq j \leq \bar{L}_1$ (resp. $1 \leq j \leq \bar{L}_2$). In a similar manner, these markings can be further generalized [51] to consider arrivals in batches made of an arbitrary combination of customers of different types. In general, this arrival process is referred to as Marked MAP (MMAP[K]), where K is the number of customer types involved. The states of the underlying Markov chain, for both PH distribution and MAPs, are also referred to as the *phases* of the distribution or the arrival process.

As a final note, let us recall that in Kendall's notation an A/S/1 queue corresponds to a single server queue with an infinite buffer, where A and S determine the arrival and service processes, respectively. In this thesis we are typically interested on queues fed by MAPs or PH arrival processes. Also, the service times are typically assumed to follow a PH distribution. If these times follow an exponential distribution, the service process is denoted with an M. For instance, a single-server queue where customers of two different classes arrive in batches, according to a BMAP[2], and each class receives exponential service times, will be refer to as a BMAP[2]/M[2]/1 queue. Adding a buffer of size C and letting the service times to follow a PH distribution, would turn it into a BMAP[2]/PH[2]/1/C queue.

A.2 The BMAP[2]/PH[2]/1 preemptive priority queue

The purpose of this appendix is to provide the definition of the blocks that make up the generator matrix Q (Eq. (2.1)) of the MC that describes the BMAP/PH/1 priority queue with two customer classes. The parameters of the arrival process and the service-time distributions can be found in Section 2.3. First, let the matrix $\bar{D}_1^j$ be defined as $\bar{D}_1^j = \sum_{k=j}^{\bar{L}_1} D_1^k$, for $1 \leq j \leq \bar{L}_1$, and let $\bar{D}_1^0 = D_0 + \bar{D}_1^1$. The $m_b \times m_b$ block B_0 is given by

$$B_0 = \begin{bmatrix}
D_0 & D_1^1 \otimes \alpha & D_1^2 \otimes \alpha & \cdots & D_1^{\bar{L}_1} \otimes \alpha & & & & 0 \\
I \otimes t & D_0 \oplus T & D_1^1 \otimes I & \cdots & D_1^{\bar{L}_1-1} \otimes I & \ddots & & & \\
 & I \otimes t\alpha & D_0 \oplus T & \cdots & D_1^{\bar{L}_1-2} \otimes I & \ddots & & & \\
 & & \ddots & \ddots & \ddots & \ddots & & & \\
 & & & I \otimes t\alpha & D_0 \oplus T & \cdots & D_1^{\bar{L}_1-2} \otimes I & D_1^{\bar{L}_1-1} \otimes I & \bar{D}_1^{\bar{L}_1} \otimes I \\
 & & & & \ddots & \ddots & \vdots & \vdots & \vdots \\
 & & & & & I \otimes t\alpha & D_0 \oplus T & D_1^1 \otimes I & \bar{D}_1^2 \otimes I \\
 & & & & & & I \otimes t\alpha & D_0 \oplus T & \bar{D}_1^1 \otimes I \\
0 & & & & & & & I \otimes t\alpha & \bar{D}_1^0 \oplus T
\end{bmatrix},$$

where $m_b = m_a + C m_a m_1$ and I is the identity matrix of appropriate dimension. Here $\otimes$ denotes the Kronecker product and $\oplus$ the Kronecker sum [16]. Since in level zero there are no low-priority customers, all the transitions in B_0 are associated to arrivals and service completions of high-priority customers. In addition, the finiteness of the high-priority buffer is reflected in the last block column of B_0, e.g., if there are $C-2$ high-priority

customers in the queue and 2 or more of these customers arrive in a batch ($\bar{D}_1^2$), the queue will only accept 2 and the others will be dropped. The transitions from level zero to upper levels are governed by the $m_b \times m$ blocks B_j, defined as

$$B_j = \begin{bmatrix} D_2^j \otimes \beta & 0 & \cdots & 0 \\ 0 & D_2^j \otimes \beta \otimes I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & D_2^j \otimes \beta \otimes I \end{bmatrix}, \quad j = 1, \ldots, \bar{L}_2,$$

where $m = m_a m_2 + C m_a m_2 m_1$. Since in level zero there are no low-priority customers, if one of these arrives and finds the server idle, it starts service immediately, selecting an initial service phase according to β. If there is a high-priority customer in service, the incoming customer selects the phase in which it will eventually start service. The $m \times m_b$ block C_0 holds the transitions from level one to level zero and is given by

$$C_0 = \begin{bmatrix} I_{m_a} \otimes s & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 \end{bmatrix},$$

which reflects the fact that the completion of a low-priority service can only occur if there are no high priority customers in the system.

Before determining the remaining blocks we define the operator $R(\cdot)$ applied on a matrix M as $R(M) = I_{m_2} \otimes M$, which will help to make the notation simpler. The $m \times m$ block A_1 is given by

$$A_1 = \begin{bmatrix} D_0 \oplus S & D_1^1 \otimes R(\alpha) & D_1^2 \otimes R(\alpha) & \cdots & D_1^{\bar{L}_1} \otimes R(\alpha) & & & 0 \\ I \otimes t & D_0 \oplus R(T) & D_1^1 \otimes I & \cdots & D_1^{\bar{L}_1 - 1} \otimes I & \ddots & & \\ & I \otimes t\alpha & D_0 \oplus R(T) & \cdots & D_1^{\bar{L}_1 - 2} \otimes I & \ddots & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & I \otimes t\alpha & D_0 \oplus R(T) & \cdots & D_1^{\bar{L}_1 - 1} \otimes I & \bar{D}_1^{\bar{L}_1} \otimes I \\ & & & & \ddots & \ddots & \vdots & \vdots \\ & & & & & I \otimes t\alpha & D_0 \oplus R(T) & \bar{D}_1^1 \otimes I \\ 0 & & & & & & I \otimes t\alpha & \bar{D}_1^0 \oplus R(T) \end{bmatrix}.$$

The first block row of this matrix reflects how a low-priority customer that is being attended may be preempted by the arrival of a batch of high-priority customers, and the first customer in the batch starts service with phase selected according to α. The use of $R(\cdot)$ reflects that the system 'remembers' the phase in which the first low-priority customer in the queue will eventually re-start its service. The transitions to upper levels are driven by the blocks

$$A_{j+1} = \begin{bmatrix} D_2^j \otimes I & 0 & \cdots & 0 \\ 0 & D_2^j \otimes I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & D_2^j \otimes I \end{bmatrix}, \quad j = 1, \ldots, \bar{L}_2, \tag{A.6}$$

which are related only to arrivals of low-priority customers. This concludes the description of the blocks since the block A_0, that holds the transitions from level k to level $k-1$, was already described in Section 2.3.

A.3 Mean field analysis

A mean field model aims at describing the behavior of a system composed of a large number of interacting particles. A system of multiple particles is hard to analyze in general because the state of each component must be traced separately. Assuming that each particle can be in one of r states, then to describe a system with N particles it would be necessary to consider a state space of size r^N. With a mean field model it is possible to analyze this type of system by simplifying the description of the interactions among the particles when their number tends to infinity. There are various types of mean field models, which differ on how the effect of the "environment" on a single particle is described, with the environment being made of (at least) the other particles in the system [13,79]. Mean field models have been extensively used in statistical physics [13,34] to describe the behavior of gases and other systems of particles where the interaction among these is assumed to be weak. In recent years, there has been an increasing interest in using mean field models to analyze systems such as buffers implementing active queue management with multiple TCP connections [12], networks of queues with load balancing mechanisms [34], medium access control protocols [22], reputation systems in ad-hoc networks [88], among others.

In this thesis we have made use of mean-field models to compute relevant metrics for a grid computing network (Chapter 4), and for an optical switch with either a pool of centralized converters (Chapter 5) or a set of buffers and converters per output port (Chapter 6). All these models fall within the framework put forward in [79] to describe the interaction of a general system of interacting particles. Therefore we have found it adequate to summarize in this appendix the main definitions and results of [79]. For the proofs of these results, examples of systems that can be modeled within this framework and a more thorough discussion we refer the reader to [79].

Let a system consist of N particles that evolve in discrete time, where each of the particles can be in one of r states: $\{1,\ldots,r\}$. Let $X_n^N(t)$ be the state of particle n at time t in a system with N particles, for $n=1,\ldots,N$. Also, let $M_j^N(t)$ be the proportion of particles in state j at time t, i.e., $M_j^N(t)=\frac{1}{N}\sum_{n=1}^{N}\mathbf{1}_{\{X_n^N(t)=j\}}$, for $j=1,\ldots,r$, where $\mathbf{1}_{\{\cdot\}}$ is the indicator function. The vector $M^N(t)=\left[M_1^N(t),\ldots,M_r^N(t)\right]$ is referred to as the *occupancy vector* and holds the proportion of objects in each state. The transitions of a single object are allowed to depend on its current state and the value of the occupancy vector. Let $R_{ij}^N(m)$ be the probability that a single object makes a transition from state i to state j when the occupancy vector is equal to m, i.e.,

$$R_{ij}^N(m)=P\left\{X_n^N(t+1)=j|X_n^N(t)=i,M^N(t)=m\right\},$$

for every $n=1,\ldots,N$, and $1\leq i,j\leq r$. Therefore $R^N(m)=[R_{ij}^N(m)]_{i,j=1}^r$ is the transition matrix for a single object, which depends on the value m of the occupancy vector and the number of objects N.

The main result in [79] states that the occupancy vector $M^N(t)$ converges to a deterministic dynamic system described by the vector $\mu(t)$, for every $t \geq 0$. For this result to hold it is necessary to make the following assumptions. The entries $R_{ij}^N(m)$ converge uniformly in m to some $R_{ij}(m)$, which is a continuous function of m, as $N \to \infty$, for every $1 \leq i, j \leq r$. In addition, the initial occupancy measure $M^N(0)$ converges almost surely to a deterministic limit $\mu(0)$. Now, define $\mu(t)$ iteratively as $\mu(t+1) = \mu(t)R(\mu(t))$, for $t \geq 0$. Under these assumptions, [79, Theorem 4.1] states that for any fixed time t, almost surely,

$$\lim_{N \to \infty} M^N(t) = \mu(t).$$

In consequence, if the assumptions hold, we can use the vector $\mu(t)$ to approximate $M^N(t)$ when N is large. Moreover, the size of $\mu(t)$ and $R(\cdot)$ is equal to the cardinality of a single object's state space r, which means that the dimensionality problems mentioned in the beginning of this section are avoided. To compute an approximation for $M^N(t)$ we start with $\mu(0)$, then compute $R(\mu(0))$ and multiply these two to obtain $\mu(1)$, and repeat the same operation t times until $\mu(t)$ is found. As these operations can be performed quickly, it is possible to compute $\mu(t)$ very fast even for large values of t. What we have shown here is a simplified version of the model in [79], which also allows the particles to belong to different classes, a characteristic that is exploited in the models in this thesis (e.g. in Section 4.3). Additionally, the framework in [79] also includes a global memory on which the transition of the objects may depend. This property however is not used in any of our models.

A final word must be said on the limitations of this framework, as they have direct consequences on the models that can benefit from it. First, the main result in [79] is related to the limit of the occupancy vector when the number of objects tends to infinity, but nothing is said about the limit when t tends to infinity. The framework is therefore very well suited to evaluate the performance of a system for any finite time (transient regime). In many cases, as those analyzed along this thesis, we are interested in the long-run behavior of the system, and in its stationary state if it exists. In the systems we have considered, we experimentally found that the value of $\mu(t)$ tends toward a fixed point that matches very well with results obtained via simulations. We have therefore used those results to evaluate the performance of the system, always obtaining a single or multiple (periodic) fixed point. The details can be found in the chapters where these models are introduced. A second aspect to highlight is the dependence of the single-object transition on the occupancy vector and not on the state of any specific object. In other words, the evolution of an object cannot be affected by any other particular object directly, but only by the fraction of objects in each state. This implies that, for instance, in the case of the grid network (see Chapter 4) the topology can only be marginally treated.

A.4 Sylvester matrix equations

In this section we describe how to solve a Sylvester matrix equation of the form $AXB + X = E$ using the Hessenberg-Schur decomposition proposed in [47]. This equation has appeared in Part I of this thesis, specifically in equations (1.12), (1.13), (2.10) and (2.11).

Consider Equation (1.12) and let $n = m - r$, then X and E are $n \times r$ matrices, while A and B are square matrices of size n and r, respectively. The first step to solve this linear system is to find orthogonal matrices U and V such that $U'AU = P$ and $V'BV = R$, where P is an upper-Hessenberg matrix, R is a upper quasi-triangular matrix and $'$ denotes the transpose operator. A matrix P is upper-Hessenberg if its entries $P_{ij} = 0$ for $i > j+1$. A upper quasi-triangular matrix, also called real Schur form, is block-triangular with 1×1 (resp. 2×2) diagonal blocks that correspond to the real (resp. complex) eigenvalues [48]. While the Hessenberg decomposition to obtain U can be done using Householder transformations, the real Schur decomposition to compute V makes use of the QR algorithm, see [48, Chapter 7]. Let $F = U'EV$ and $Y = U'XV$, then the linear system becomes $PYR + Y = F$. Therefore, to find Y_k, the k-th column of the matrix Y, we need to solve the system

$$P \sum_{j=1}^{\max(k+1,r)} R_{jk} Y_j + Y_k = F_k,$$

for $1 \leq k \leq r$. However, the upper quasi-triangular form of R greatly simplifies this system. For $k < r$ there are two possible cases, either $R_{k+1,k} = 0$ or not. If $R_{k+1,k} = 0$, then Y_k is the solution to the $n \times n$ Hessenberg system

$$(PR_{k,k} + I)Y_k = F_k - \sum_{j=1}^{k-1} R_{jk} P Y_j, \tag{A.7}$$

which can be solved in $O(n^2)$ time. On the other hand, $R_{k+1,k} \neq 0$ implies $R_{k+2,k+1} = 0$, and hence we need to solve

$$\begin{bmatrix} PR_{k,k} + I & PR_{k+1,k} \\ PR_{k,k+1} & PR_{k+1,k+1} + I \end{bmatrix} \begin{bmatrix} Y_k \\ Y_{k+1} \end{bmatrix} = \begin{bmatrix} \hat{F}_k^{k-1} \\ \hat{F}_{k+1}^{k-1} \end{bmatrix}, \tag{A.8}$$

where $\hat{F}_k^l = F_k - \sum_{j=1}^{l} R_{jk} P Y_j$, for $1 \leq l \leq k-1$ and $1 \leq k \leq r$. This $2n \times 2n$ linear system is upper-triangular with two nonzero subdiagonals that can be solved in $O(n^2)$ time [47]. Notice, to determine Y_k it is necessary to know $Y_1, \ldots, Y_{k-1}$. Therefore, the algorithm starts by computing the first (or first two) column(s), and then works forward until the last column of Y has been computed. After finding Y, the matrix X can be computed as $X = UYV'$.

It is possible to apply this procedure to either the original or the transpose $B'X'A' + X' = E'$ system. In the first case A is transformed into Hessenberg form and B into real Schur form, while the opposite happens in the second case. The choice directly affects the computation times since for a matrix of size b the Schur decomposition can be done in $10b^3$ operations, while it takes $\frac{5}{3}b^3$ operations to compute the Hessenberg decomposition using Householder transformations [47,48]. Therefore, to solve Equation (1.12) it is better to use the original system since the Hessenberg decomposition is applied on the $n \times n$ matrix A, which is larger than the $r \times r$ matrix B under the assumption that $r \ll m$. On the other hand, to solve Equation (1.13) it is preferable to first transpose the system since in that case B is an $n \times n$ matrix given by $B = A_0^{--}(I - A_1^{--})^{-1}$.

An additional issue to take into account when solving equations (1.12) and (1.13) is the actual computation of the matrices A, B and E. Take for example Equation (1.12), where $A = (I - A_1^{--})^{-1}A_2^{--}$, $B = G_+$ and $E = (I - A_1^{--})^{-1}(A_0^{-+} + A_1^{-+}G_+ + A_2^{-+}G_+^2)$. Although all the matrices involved are already computed it is still necessary to perform two matrix multiplications to determine A and E. In the examples shown in sections 1.4 and 2.3, the A_i^{-+} blocks are sparse or even zero. In some cases however these blocks can be dense and therefore these matrix multiplications may require considerably more time. A way to avoid this in Equation (1.12), is to solve the slightly different equation

$$(I - A_1^{--})G_0 - A_2^{--}G_0G_+ = A_0^{-+} + A_1^{-+}G_+ + A_2^{-+}G_+^2,$$

which is a Sylvester matrix equation of the type $AXB+CX = E$. The procedure to solve this equation is very similar to the one shown above, but in this case the first step of the QZ algorithm [48] is applied to the pair (A, C). As a result A is reduced to Hessenberg form while C is transformed into upper-triangular form. This, together with a reduction of B to quasi-upper triangular form, allows the solution of this equation in a similar way as done in (A.7) and (A.8). A detailed explanation can be found in [44]. Since the matrices A, B, C and E are already computed, this algorithm may perform better when the blocks A_i^{-+} are dense. We have found instances of random QBDs with dense blocks where this last algorithm outperforms the one based on the equation $AXB + X = E$.

A.5 Dual processes

In this section we describe two dual relationships between discrete-time M/G/1- and GI/M/1-type processes. In both cases the dual process can be seen as the time-reverse of the original process with respect to an invariant measure [11, 23]. We consider the computation of an M/G/1-type MC as the dual of a GI/M/1-type MC, but the opposite relationship can be defined in a similar manner. The Ramaswami dual was introduced in [98] and its probabilistic interpretation given in [11]. Let the set of matrices $(A_i)_{i\geq 0}$ describe a GI/M/1-type MC, such that $A = \sum_{i=0}^{\infty} A_i$ is stochastic and irreducible. Then A is the transition matrix of a discrete-time MC with stationary probability vector α, i.e., $\alpha A = \alpha$ and $\alpha e = 1$. The Ramaswami dual is an M/G/1-type MC characterized by the set of matrices $(A_i^R)_{i\geq 0}$ given by $A_i^R = \Delta_R^{-1}A_i'\Delta_R$, where $\Delta_R = diag(\alpha)$. The G matrix of this process, denoted G_R, is related to the R matrix of the original process by $G_R = \Delta_R^{-1}R'\Delta_R$. Let $\rho(M)$ denote the spectral radius of a matrix M. Since the matrix G_R has the same eigenvalues as R, if the original GI/M/1-type MC is positive recurrent ($\rho(R) < 1$) the dual process is transient ($\rho(G_R) < 1$), and vice versa. The dual process will be null recurrent if and only if the original process is also null recurrent. In this case the dual process is the time-reverse process with respect to the invariant measure α.

We now turn to the Bright dual [23], which is defined as the time-reverse process with respect to a different invariant measure. If the GI/M/1-type MC is positive recurrent, the eigenvalue of maximum real part of R is $\eta = \rho(R) < 1$. It has been shown that the spectral radius of the matrix $\sum_{i=0}^{\infty} A_i\eta^i$ is equal to one [92]. Therefore there exists a

unique positive vector w_η such that

$$w_\eta \left(\sum_{i=0}^{\infty} A_i \eta^i \right) = w_\eta.$$

The Bright dual is an M/G/1-type MC characterized by the matrices $(A_i^B)_{i\geq 0}$ defined as $A_i^B = \eta^{i-1}\Delta_B^{-1} A_i' \Delta_B$, where $\Delta_B = diag(w_\eta)$. The matrix R of the original GI/M/1-type MC and the matrix G_B of the dual process are related by $G_B = \eta^{-1}\Delta_B^{-1} R' \Delta_B$. In this case the eigenvalues of the matrix G_B are the eigenvalues of R divided by η. Hence the spectral radius of G_B is equal to one and the dual process is positive recurrent [23]. When the process is positive recurrent, as in the examples shown in Section 1.4 for loads less than one, the Ramaswami dual will be transient while the Bright dual will be positive recurrent. As explained in detail in [109], the Bright dual can therefore reduce the computation times achieved by the Ramaswami dual considerably. This is confirmed numerically in Section 1.4, especially when the load of the overflow queue is small, which results in a large number of blocks for the GI/M/1-type MC and a small value of η. The computation time for the reduced process increases with the number of blocks, but the gain that can be realized by using the Bright dual is larger when η is smaller [109]. Therefore, the Bright dual becomes especially useful in this case as it compensates the larger computation times caused by the number of blocks.

A.6 Moments of first passage times in a finite QBD

A relevant issue for the approximation methods introduced in Chapter 3 is the computation of the moments of an inter-event time distribution. The purpose of this appendix is to provide an algorithm to compute these moments based on [46]. In both methods, the inter-event time distribution has a PH or ME representation characterized by a matrix with a QBD structure. For the method in Section 3.2, this matrix is shown in Equation (3.2), while for the ON-OFF method it is given by Equation (3.6). In both cases the inter-event time distribution can be seen as a first-passage time distribution to a higher level in a finite QBD (level k is the set of states $\{(k,l), 1 \leq l \leq m_i\}$). In the ON-OFF approximation the inter-event times correspond to the length of the OFF periods. An OFF period starts when the system moves from level C_i to level $C_i - 1$, where the initial phase in level $C_i - 1$ is selected according to η^i, as defined in Equation (3.7). The OFF period ends as soon as the process reaches level C_i again, i.e., the first time it visits level C_i starting from level $C_i - 1$. A similar observation can be made for the approximation based on the overflow process. Since an overflow can only occur when the system is in level C_i, an inter-overflow time always starts in this level. In this case, however, there is no higher level than C_i, but we can define a fictitious level $C_i + 1$ of absorbing states that can only be accessed from level C_i with rates $D_+^i - D_s^i$. Then, the inter-overflow time can be seen as the first-passage time from level C_i to level $C_i + 1$, where the vector β^i in Equation (3.4) defines the initial phase in level C_i.

As both inter-event time distributions can be regarded as first-passage times to higher levels in a finite QBD, we can use the results in [46] to compute the moments of these

Algorithm A.1 Algorithm to compute N moments of the OFF period length distribution

1: $K_1 \leftarrow -D_0^{-1}$
2: $u_1^{(1)} \leftarrow K_1 e$
3: **for** $k = 2$ to N **do**
4: $\quad u_1^{(k)} \leftarrow k K_1 u_1^{(k-1)}$
5: **end for**
6: $U_1^{(1)} \leftarrow K_1^2 D_+$
7: **for** $k = 2$ to $N-1$ **do**
8: $\quad U_1^{(k)} \leftarrow k K_1 U_1^{(k-1)}$
9: **end for**
10: **for** $n = 2$ to C **do**
11: $\quad K_n \leftarrow -\left(D_0 - (n-1)\mu I + (n-1)\mu K_{n-1} D_+\right)^{-1}$
12: $\quad T_n^{(1)} \leftarrow K_n \left(I + (n-1)\mu U_{n-1}^{(1)}\right)$
13: $\quad$ **for** $k = 2$ to $N-1$ **do**
14: $\quad\quad T_n^{(k)} \leftarrow (n-1)\mu K_n U_{n-1}^{(k)}$
15: $\quad$ **end for**
16: $\quad u_n^{(1)} \leftarrow K_n \left(e + (n-1)\mu u_{n-1}^{(1)}\right)$
17: $\quad$ **for** $k = 2$ to N **do**
18: $\quad\quad u_n^{(k)} \leftarrow (n-1)\mu K_n u_{n-1}^{(k)} + \sum_{j=1}^{k-1} \binom{k}{j} T_n^{(k-j)} u_n^{(j)}$
19: $\quad$ **end for**
20: $\quad$ **for** $k = 1$ to $N-1$ **do**
21: $\quad\quad U_n^{(k)} \leftarrow T_n^{(k)} K_n D_+ + \sum_{j=1}^{k-1} \binom{k}{j} T_n^{(k-j)} U_n^{(j)}$
22: $\quad$ **end for**
23: **end for**
24: **for** $k = 1$ to N **do**
25: $\quad r_k \leftarrow \eta u_C^{(k)}$
26: **end for**

distributions. In [46] the authors determine the generating function of the first-passage times to higher levels in a finite level-dependent QBD. Based on this, they derive an algorithm to compute the first two moments, but any higher moment can be computed from the generating function in a similar manner. Here we focus on the computation of N moments of the OFF-period distribution, but this can be easily modified for the inter-overflow time distribution. Algorithm A.1 presents the steps to compute these moments in an arbitrary station, dropping the super(sub)script i. For a better understanding of the algorithm we introduce a few definitions. As described in Section 3.1, the state of a station can be represented by the process $\{(N(t), J(t)), t \geq 0\}$ on the state space $\{(k,l), 0 \leq k \leq C, 1 \leq l \leq m\}$. Now let X_n be the first-passage time from level $n-1$ to level n, for $n = 1, \ldots, C$, i.e., $X_n = \inf\{t > 0 : N(t) = n | N(0) = n-1\}$. Therefore, the length of the OFF-period is given by X_C. Let $U_n^{(k)}$ be the $m \times m$ matrix with entries

$$[U_n^{(k)}]_{ij} = E[X_n^k, J(X_n) = j | N(0) = n-1, J(0) = i],$$

i.e, the k-th moment of the first-passage time from level $n-1$ to level n, when the process

starts in phase i and the first visit to level n occurs in phase j, for $1 \leq n \leq C$, $1 \leq i, j \leq m$ and $k \geq 1$. Also, let $u_n^{(k)}$ be the $m \times 1$ vector with entries

$$[u_n^{(k)}]_i = E[X_n^k | N(0) = n-1, J(0) = i],$$

i.e, the k-th moment of the first-passage time from level $n-1$ to level n, given that the process starts in phase i. Finally, let r_k be the k-th moment of the first-passage time distribution from level $C-1$ to level C, i.e.,

$$r_k = E[X_C^k | N(0) = C-1] = \eta u_C^{(k)},$$

where η is the probability distribution of the initial phase in level $C-1$. Algorithm A.1 computes r_k, for $1 \leq k \leq N$. The procedure starts by computing the moments of the first-passage times from level 0 to level 1, resulting in vectors $u_1^{(k)}$. The algorithm then computes the same quantities for levels 2 to C. To obtain the N vectors $\{u_n^{(k)}, 1 \leq k \leq N\}$ at each step, the algorithm requires the $N-1$ matrices $\{U_n^{(k)}, 1 \leq k \leq N-1\}$ and the matrix K_n. The matrix $-K_n^{-1}$ is the generator of the process restricted to level $n-1$ before the first visit to level n. Additionally, the $N-1$ auxiliary matrices $\{T_n^{(k)}, 1 \leq k \leq N-1\}$ are used to simplify the expressions.

Bibliography

[1] The enabling grids for e-science project. Online: `http://www.eu-egee.org`.

[2] N. Akar, E. Karasan, and K. Dogan. Wavelength converter sharing in asynchronous optical packet/burst switching: an exact blocking analysis for markovian arrivals. *IEEE Journal on Selected Areas in Communications*, 24:69–80, 2006.

[3] N. Akar, E. Karasan, G. Muretto, and C. Raffaelli. Performance analysis of an optical packet switch employing full/limited range share per node wavelength conversion. In *Proceedings of IEEE Globecom 2007*, 2007.

[4] N. Akar, E. Karasan, and C. Rafaelli. Fixed point analysis of limited range share per node wavelength conversion in asynchronous optical packet switching systems. *Photonic Network Communications*, 18:255–263, 2009.

[5] N. Akar and K. Sohraby. An invariant subspace approach in M/G/1 and G/M/1 type Markov chains. *Communications in Statistics Stochastic Models*, 13:381–416, 1997.

[6] A. S. Alfa. Matrix-geometric solution of discrete time MAP/PH/1 priority queue. *Naval Research Logistics*, 45:23–50, 1998.

[7] S. Asmussen and M. Bladt. Renewal theory and queueing algorithms for matrix-exponential distributions. In S. Chakravarthy and A. S. Alfa, editors, *Matrix-Analytic Methods in Stochastic Models*, pages 313–341. Marcel Dekker, New York, 1996.

[8] S. Asmussen and M. Bladt. Point processes with finite-dimensional conditional probabilities. *Stochastic Processes and their Applications*, 82:127–142, 1999.

[9] S. Asmussen, O. Nerman, and M. Olsson. Fitting Phase-type distributions via the EM algorithm. *Scandinavian Journal of Statistics*, 23:419–441, 1996.

[10] S. Asmussen and C. A. O'Cinneide. Matrix-exponential distributions. In S. Kotz, C. B. Read, and D. L. Banks, editors, *Encyclopedia of Statistical Sciences*, volume 2, pages 435–440. Wiley, New York, 1998.

[11] S. Asmussen and V. Ramaswami. Probabilistic interpretations of some duality results for the matrix paradigms in queueing theory. *Communications in Statistics Stochastic Models*, 6:715–733, 1990.

[12] F. Baccelli, D. R. McDonald, and J. Reynier. A mean-field model for multiple tcp connections through a buffer implementing red. *Performance Evaluation*, 49:77–97, 2002.

[13] R.J. Baxter. *Exactly solved models in statistical mechanics*. Academic Press, 1982.

[14] BEA Systems, IBM, Microsoft Corporation Inc, and TIBCO Software Inc. Web services reliable messaging protocol (WS-ReliableMessaging), 2005.

[15] N. G. Bean and B. F. Nielsen. Quasi-birth-and-death processes with rational arrival process components. Technical Report 2007-20, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, 2007.

[16] R. Bellman. *Introduction to Matrix Analysis*. McGraw-Hill, 1970.

[17] D. Bini, G. Latouche, and B. Meini. Solving nonlinear matrix equations arising in tree-like stochastic processes. *Linear Algebra and its Applications*, 366:39–64, 2003.

[18] D. Bini, G. Latouche, and B. Meini. *Numerical Methods for Structured Markov Chains*. Oxford University Press, 2005.

[19] D. Bini and B. Meini. On the solution of a nonlinear matrix equation arising in queueing problems. *SIAM Journal of Matrix Analysis and Applications*, 17:906–926, 1996.

[20] M. Bladt and M. F. Neuts. Matrix-exponential distributions: calculus and interpretations via flows. *Stochastic Models*, 19:113–124, 2003.

[21] A. Bobbio, A. Horvath, and M. Telek. Matching three moments with minimal acyclic Phase-type distributions. *Stochastic Models*, 21:303–326, 2005.

[22] C. Bordenave, D. McDonald, and A. Proutiere. Performance of random medium access control, an asymptotic approach. In *SIGMETRICS '08: Proceedings of the 2008 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 1–12, New York, NY, USA, 2008. ACM.

[23] L. Bright. *Matrix-Analytic methods in applied probability*. PhD thesis, Department of Applied Mathematics, University of Adelaide, 1996.

[24] C. J. Burke and M. Rosenblatt. A Markovian function of a Markov chain. *The Annals of Mathematical Statistics*, 29:1112–1122, 1958.

[25] J. Cai, A. S. Alfa, P. Ren, X. Shen, and J. W. Mark. Packet level performance analysis in wireless user-relaying networks. *IEEE Transactions on Wireless Communications*, 7:12, 2008.

[26] F. Callegati. Approximate modeling of optical buffers for variable length packets. *Photonic Network Communications*, 3:383–390, 2001.

[27] F. Callegati, W. Cerroni, G. Corazza, C. Develder, M. Pickavet, and P. Demeester. Scheduling algorithms for a slotted packet switch with either fixed or variable lengths packets. *Photonic Network Communications*, 8:163–176, 2004.

[28] F. Callegati, W. Cerroni, C. Rafaelli, and P. Zaffoni. Wavelength and time domain exploitation for QoS management in optical packet switches. *Computer Networks*, 44:569–582, 2004.

[29] E. Çinlar. *Introduction to stochastic processes*. Prentice-Hall, 1975.

[30] Y. Chandramouli, M. Neuts, and V. Ramaswami. A queueing model for meteor burst packet communication systems. *IEEE Transactions on Communications*, 37:1024–1030, 1989.

[31] K. Christodoulopoulus, M. Varvarigos, C. Develder, M. De Leenheer, and B. Dhoedt. Job demand models for optical grid research. In *Proc. 11th Conference on Optical Network Design and Modelling (ONDM)*, Athens, Greece, 2007.

[32] G. Ciardo and E. Smirni. ETAQA: an efficient technique for the analysis of QBD-processes by aggregation. *Performance Evaluation*, 36-37:71–93, 1999.

[33] A. Cumani. On the canonical representation of homogeneous Markov processes modeling failure-time distributions. *Microelectronics Reliability*, 22:583–602, 1982.

[34] D. A. Dawson, J. Tang, and Y. Zhao. Balancing queues by mean field interaction. *Queueing Systems*, 49:335–361, 2005.

[35] M. De Leenheer, C. Develder, T. Stevens, B. Dhoedt, M. Pickavet, and P. Demeester. Design and control of optical grid networks (invited). In *Proc. 4th Int. Conf. on Broadband Networks (Broadnets 2007)*, Raleigh, NC, Sep. 2007.

[36] M. De Leenheer, F. Farahmand, P. Thysebaert, B. Volckaert, F. De Turck, B. Dhoedt, P. Demeester, and J. Jue. Anycast routing in optical burst switched grid networks. In *Proc. 31st European Conference on Optical Communication (ECOC)*, Glasgow, Scotland, 2005.

[37] K. De Turck, S. De Vuyst, D. Fiems, S. Wittevrongel, and H. Bruneel. Performance of the sleep-mode mechanism of the new IEEE 802.16m proposal for correlated downlink traffic. In *NET-COOP*, 2009.

[38] C. Develder, M. De Leenheer, T. Stevens, B. Dhoedt, F. De Turck, and P. Demeester. Scheduling in optical grids: a dimensioning point of view. In *Proc. Conference on the Optical Internet - Australian Conference on Optical Fibre Technology (COIN-ACOFT)*, Melbourne, Australia, 2007.

[39] C. Develder, B. Dhoedt, B. Mukherjee, and P. Demeester. On dimensioning optical grids and the impact of scheduling. *Photonic Network Communications*, 17:255–265, 2009.

[40] J. E. Diamond and A. S. Alfa. On approximating higher order MAPs with MAPs of order two. *Queueing Systems*, 34:269–288, 2000.

[41] K. Dogan, Y. Gunulay, and N. Akar. A comparative study of limited range wavelength conversion policies for asynchronous optical packet switching. *Journal of Optical Networking*, 6:134–145, 2007.

[42] V. Eramo and M. Listanti. Packet loss in a bufferless optical WDM switch emplying shared tunable wavelength converters. *Journal of Lightwave Technology*, 18:1818–1833, 2000.

[43] M. Fackrell. Fitting with matrix-exponential distributions. *Stochastic Models*, 21:377–400, 2005.

[44] J. D. Gardiner, A. J. Laub, J. J. Amato, and C. B. Moler. Solution of the Sylvester matrix equation AXBT + CXDT = E. *ACM Transactions on Mathematical Software*, 18:223–231, 1992.

[45] C. M. Gauger. Optimized combination of converter pools and FDL buffers for contention resolution in optical burst switching. *Photonic Network Communications*, 8:139–148, 2004.

[46] D. P. Gaver, P. A. Jacobs, and G. Latouche. Finite birth-and-death models in randomly changing environments. *Advances in Applied Probability*, 16:715–731, 1984.

[47] G. H. Golub, S. Nash, and C. Van Loan. A Hessenberg-Schur method for the problem AX+XB=C. *IEEE Transactions on Automatic Control*, 24:909–913, 1979.

[48] G. H. Golub and C. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 1996.

[49] W. B. Gragg and A. Lindquist. On the partial realization problem. *Linear Algebra and its Applications*, 50:277–319, 1983.

[50] W. K. Grassmann and J. Tavakoli. Solving QBD processes when levels can increase only in certain phases. Manuscript in preparation, presented at the MAM6 conference, Beijing (China), June 2008.

[51] Q. He. The versatility of MMAP[K] and the MMAP[K]/G[K]/1 queue. *Queueing Systems: Theory and Applications*, 38:397–418, 2001.

[52] Q. He and M. F. Neuts. Markov chains with marked transitions. *Stochastic Processes and their Applications*, 74:37–52, 1998.

[53] Q. He and H. Zhang. On matrix exponential distributions. *Advances in Applied Probability*, 39:271–292, 2007.

[54] H. Heffes. A class of data traffic processes - covariance function characterization and related queuing results. *Bell System Technical Journal*, 59:897–929, 1980.

[55] A. Heindl. Decomposition of general queueing networks with MMPP inputs and customer losses. *Perform. Eval.*, 51(2-4):117–136, 2003.

[56] A. Heindl. Inverse characterization of hyperexponential MAP(2)s. In *Proc. 11th Int. Conference on Analytical and Stochastic Modelling Techniques and Applications (ASMTA)*, 2004.

[57] A. Heindl, K. Mitchell, and A. van de Liefvoort. Correlation bounds for second-order MAPs with application to queueing network decomposition. *Perform. Eval.*, 63(6):553–577, 2006.

[58] A. Heindl, Q. Zhang, and E. Smirni. ETAQA truncation models for the MAP/MAP/1 departure process. In *QEST '04: Proceedings of the The Quantitative Evaluation of Systems, First International Conference*, pages 100–109, Washington, DC, USA, 2004.

[59] A. Horváth, G. Horváth, and M. Telek. Moments-based characterization of colored Markov arrival processes. http://webspn.hit.bme.hu/∼telek/techrep/cmap.pdf, Oct. 2007.

[60] A. Horváth, G. Horváth, and M. Telek. A joint moments based analysis of networks of MAP/MAP/1 queues. In *Proc. of the 5th International Conference on the Quantitative Evaluation of Systems (QEST)*, St Malo, France, Sept 2008.

[61] A. Horváth, G. Horváth, and M. Telek. A traffic based decomposition of two-class queueing networks with priority service. *Computer Networks*, 53:1235–1248, 2009.

[62] A. Horváth and M. Telek. Matching more than three moments with acyclic phase type distributions. *Stochastic Models*, 23:167–194, 2007.

[63] M. A. Johnson and M. R. Taaffe. Matching moments to phase distributions: mixtures of Erlang distributions of common order. *Comm. Statist. Stochastic Models*, 5(4):711–743, 1989.

[64] R. Johnson and D. Wichern. *Applied Multivariate Statistical Analysis*. Prentice-Hall, 1998.

[65] J. G. Kemeny and J. L. Snell. *Finite Markov chains*. Springer-Verlag, 1960.

[66] J. G. Kemeny, J. L. Snell, and A. W. Knapp. *Denumerable Markov chains*. Springer-Verlag, 1976.

[67] A. Kuczura. The interrupted Poisson process as an overflow process. *Bell System Technical Journal*, 52:437–448, 1973.

[68] A. Kuczura and D. Bajaj. A method of moments for the analysis of a switched communication network's performance. *IEEE Transactions on Communications*, COM-25:185–193, 1977.

[69] P. J. Kuehn. Approximate analysis of general queueing networks by decomposition. *IEEE Transactions on Communications*, COM-27:113–126, 1979.

[70] V. G. Kulkarni. *Modeling and analysis of stochastic systems*. CRC Press, 1995.

[71] K. Laevens, M. Moeneclaey, and H. Bruneel. Queueing analysis of a single-wavelength fiber-delay-line buffer. *Telecommunication Systems*, 31:259–287, 2006.

[72] J. Lambert. *Performance Analysis of Optical Fiber Delay Line Buffers and DOCSIS Cable Modem Networks*. PhD thesis, University of Antwerp, 2008.

[73] J. Lambert, B. Van Houdt, and C. Blondia. Queues with correlated inter-arrival and service times and its application to optical buffers. *Stochastic Models*, 22(2):233–251, 2006.

[74] G. Latouche. Algorithms for infinite Markov chains with repeating columns. In C. D. Meyer and R. J. Plemmons, editors, *Linear Algebra, Markov chains and Queueing Models*, pages 231–265. Springer-Verlag, 1993.

[75] G. Latouche and V. Ramaswami. A logarithmic reduction algorithm for quasi-birth-and-death processes. *Journal of Applied Probability*, 30:650–674, 1993.

[76] G. Latouche and V. Ramaswami. *Introduction to Matrix Analytic Methods in Stochastic Modeling*. ASA-SIAM Series on Statistics and Applied Probability. SIAM, Philadelphia, PA, 1999.

[77] A. W. Law and W. D. Kelton. *Simulation modeling and analysis*. McGraw-Hill, third edition, 2000.

[78] G. Lawton. Moving the OS to the web. *IEEE Computer*, 41(3):16–19, March 2008.

[79] J. Le Boudec, D. McDonald, and J. Mundinger. A generic mean field convergence result for systems of interacting objects. In *Proc. 4th Int. Conf. on the Quantitative Evaluation of SysTems (QEST 2007)*, pages 3–15, Edinburgh, UK, 16–19 Sep. 2007.

[80] L. Lipsky. *Queueing Theory: a Linear Algebraic Approach*. Macmillan, New York, 1992.

[81] D. Lucantoni. New results on the single server queue with a batch markovian arrival process. *Stochastic Models*, 7:1–46, 1991.

[82] J. Matsumoto and Y. Watanabe. Individual traffic characteristics of queuing systems with multiple Poisson and overflow inputs. *IEEE Transactions on Communications*, COM-33:1–9, 1985.

[83] K. S. Meier-Hellstern. The analysis of a queue arising in overflow models. *IEEE Transactions on Communications*, 37:367–372, 1989.

[84] B. Meini. An improved FFT-based version of Ramaswami's formula. *Stochastic Models*, 13(2):223–238, 1997.

[85] H. Michiel and K. Laevens. Teletraffic engineering in a broad-band era. In *Proceedings of the IEEE*, volume 85, pages 2007–2033, 1997.

[86] D. R. Miller. Computation of steady-state probabilities for M/M/1 priority queues. *Operations Research*, 29:945–958, 1981.

[87] B. Mukherjee, D. Banerjee, S. Ramamurthy, and A. Mukherjee. Some principles for designing a wide-area WDM optical network. *IEEE/ACM Transactions on Networking*, 4(5):684–696, October 1996.

[88] J. Mundinger and J. Le Boudec. Analysis of a reputation system for mobile ad-hoc networks with liars. *Performane Evaluation*, 65(3-4):212–226, 2008.

[89] M. F. Neuts. Probability distributions of phase type. In *Liber Amicorum Prof. Emeritus H. Florin*. Katholieke Universiteit Leuven, 1975.

[90] M. F. Neuts. A versatile Markovian point process. *Journal of Applied Probability*, 16:764–779, 1979.

[91] M. F. Neuts. *Matrix-Geometric Solutions in Stochastic Models*. The John Hopkins University Press, Baltimore, 1981.

[92] M. F. Neuts. *Structured stochastic matrices of M/G/1 type and their applications*. Marcel Dekker Inc., 1989.

[93] T. Osogami and M. Harchol. Closed form solutions for mapping general distributions to quasi-minimal PH distributions. *Performance Evaluation*, 63:524–552, 2006.

[94] H. Perros. *Connection-oriented Networks: SONET/SDH, ATM, MPLS and Optical Networks*. John Wiley, 2005.

[95] V. Puttasubbapa and H. Perros. Performance analysis of limited-range wavelength conversion in an OBS switch. *Telecommunication Systems*, 31:227–246, 2006.

[96] J. Qiao and M. Yoo. Optical burst switching: A new paradigm for an optical Internet. *Journal of High-Speed Networks*, 8:69–84, 1999.

[97] V. Ramaswami. A stable recursion for the steady state vector in Markov chains of M/G/1 type. *Stochastic Models*, 4:183–188, 1988.

[98] V. Ramaswami. A duality theorem for the matrix paradigms in queueing theory. *Communications in Statistics Stochastic Models*, 6:151–161, 1990.

[99] Philipp Reinecke and Katinka Wolter. Acyclic phase-type distribution models for WSRM. `http://www2.informatik.hu-berlin.de/~preineck/acphmodels/`.

[100] Philipp Reinecke and Katinka Wolter. Phase-type approximations for message transmission times in web services reliable messaging. In *SIPEW '08: Proceedings of the SPEC international workshop on Performance Evaluation*, 2008.

[101] A. Riska, V. Diev, and E. Smirni. An EM-based technique for approximating long-tailed data sets with PH distributions. *Performance Evaluation*, 54:147–164, 2004.

[102] T. Robertazzi. *Computer Networks and Systems*. Springer, 2000.

[103] W. Rogiest, D. Fiems, K. Laevens, and H. Bruneel. Tracing an optical buffer's performance: an effective approach. In *Proceedings of the First Euro-FGI International Conference on Network Control and Optimization, NET-COOP 2007*, 2007.

[104] W. Rogiest, K. Laevens, D. Fiems, and H. Bruneel. Quantifying the impact of wavelength conversion on the performance of fiber delay line buffers. In *Proceedings of the Sixth International Workshop on Optical Burst/Packet Switching, WOBS 2006*, 2006.

[105] G. N. Rouskas and L. Xu. Optical packet switching. In K. Sivalingam and S. Subramaniam, editors, *Emerging Optical Network Technologies: Architectures, Protocols and Performance*. Springer, 2004.

[106] V. Sharma and E. Varvarigos. Limited wavelength translation in all-optical WDM mesh networks. In *Proceedings of the IEEE Infocom'98*, 1998.

[107] G. Shen, S. Bose, T. Cheng, C. Lu, and T. Chai. Performance study on a WDM packet switch with limited-range wavelength converters. *IEEE Commun. Lett.*, 5(10):432–434, 2001.

[108] J. Stapleton. *Models for probability and statistical inference: theory and applications*. John Wiley & Sons, 2008.

[109] P. G. Taylor and B. Van Houdt. On the dual relationship between Markov chains of GI/M/1 and M/G/1 type. to appear in Advances in Applied Probability.

[110] M. Telek and A. Heindl. Matching moments for acyclic discrete and continuous phase-type distributions of second order. *International Journal of Simulation Systems, Science & Technology*, 3:47–57, 2002.

[111] M. Telek and G. Horváth. A minimal representation of Markov arrival processes and a moment matching method. *Performance Evaluation*, 64:1153–1168, 2007.

[112] A. Thümmler, P. Buchholz, and M. Telek. A novel approach for fitting probability distributions to real trace data with the EM algorithm. In *Proceedings of the International Conference on Dependable Systems and Networks (DSN'05)*, pages 712–721, 2005.

[113] P. Thysebaert, F. De Turck, B. Dhoedt, and P. Demeester. Using divisible load theory to dimension optical transport networks for grid excess load handling. In *Proc. Int. Conf. on Autonomic and Autonomous Systems & Int. Conf. on Networking and Systems (ICAS/ICNS 2005)*, Papeete, Tahiti, 23–28 Oct. 2005.

[114] J. Turner. Terabit burst switching. *Journal of High-Speed Networks*, 8:3–16, 1999.

[115] A. van de Liefvoort. The moment problem for continuous distributions. Technical Report CM-1990-02, University of Missouri - Kansas City, 1990.

[116] B. Van Houdt, K. Laevens, J. Lambert, C. Blondia, and H. Bruneel. Channel utilization and loss rate in a single-wavelength Fibre Delay Line (FDL) buffer. In *Proceedings of IEEE Globecom 2004*, 2004.

[117] B. Van Houdt and J. S. H. van Leeuwaarden. Triangular M/G/1-type and tree-like QBD Markov chains. To appear in INFORMS Journal on Computing.

[118] J. S. H. van Leeuwaarden, M. S. Squillante, and E. M. M. Winands. Quasi-birth-and-death processes, lattice path counting and hypergeometric functions. *Journal of Applied Probability*, 46:507–520, 2009.

[119] V. Wallace. *The solution of quasi birth and death processes arising from multiple access computer systems*. PhD thesis, Systems Engineering Laboratory, University of Michigan, 1969.

[120] W. Whitt. Approximating a point process by a renewal process, I: Two basic methods. *Operations Research*, 30:125–147, 1982.

[121] N. Wilkins-Diehr, D. Gannon, G. Klimeck, S. Oster, and S. Pamidighantam. Teragrid science gateways and their impact on science. *IEEE Computer*, 41:32–41, 2008.

[122] R. I. Wilkinson. Theories for toll traffic engineering in the U.S.A. *Bell System Technical Journal*, 35:421–514, 1956.

[123] L. Xu and H. Perros. Performance analysis of an ingress switch in a JumpStart optical burst switching network. *Performance Evaluation*, 64:315–346, 2007.

[124] L. Xu, H. Perros, and G. N. Rouskas. A queueing network model of an edge optical burst switching node. In *Proceedings of the IEEE Infocom 2003*, 2003.

[125] S. Yao, B. Mukherjee, and S. Dixit. Contention resolution in optical packet switching. In S. Dixit, editor, *IP Over WDM: building the next-generation optical internet*. Wiley-Interscience, New York, NY, USA, 2003.

[126] S. Yao, B. Mukherjee, S. J. B. Yoo, and S. Dixit. All-optical packet-switched networks: a study of contention-resolution schemes in an irregular mesh network with variable-sized packets. In *Proc. of OPTICOMM 2000*, 2000.

[127] J. Yates, J. Lacey, D. Everitt, and M. Summerfield. Limited-range wavelength translation in all-optical networks. In *Proceedings of the IEEE Infocom'96*, 1996.

[128] T. Zhang, K. Lu, and J. P. Jue. An analytical model for shared fiber-delay line buffers in asynchronous optical packet and burst switches. In *Proceedings of the IEEE International Conference on Communications 2005*, volume 3, pages 1636–1640, 2005.

[129] J. Zhao, B. Li, X. Cao, and I. Ahmad. A matrix-analytic solution for the DBMAP/PH/1 priority queue. *Queueing Systems*, 53(3):127–145, 2006.

Related Publications

Journal Papers

- **J. F. Pérez** and B. Van Houdt. *Wavelength Allocation in an Optical Switch with a Fiber Delay Line Buffer and Limited-Range Wavelength Conversion.* Telecommunication Systems, Vol. 41 (1), pp. 37-49, 2009.
- **J. F. Pérez** and B. Van Houdt. *Markovian approximations for a grid computing network with a ring structure.* To appear in Stochastic Models.
- B. Van Houdt, C. Develder, **J. F. Pérez**, M. Pickavet and B. Dhoedt. *Mean Field Calculation for Optical Grid Dimensioning.* To appear in IEEE/OSA Journal of Optical Communications and Networking (former IEEE JSAC-OCN series).

Conference Papers

- **J. F. Pérez** and B. Van Houdt. *Exploiting restricted transitions in Quasi-Birth-and-Death processes.* Proceedings of the Sixth QEST, Budapest, Hungary, 2009.
- **J. F. Pérez** and B. Van Houdt. *Dimensioning an OBS switch with Partial Wavelength Conversion and Fiber Delay Lines via a Mean Field Model.* Proceedings of the IEEE INFOCOM 2009, Rio de Janeiro, Brazil, 2009.
- **J. F. Pérez**, J. Van Velthoven and B. Van Houdt. *Q-MAM: A Tool for Solving Infinite Queues using Matrix-Analytic Methods.* Proceedings of the SMCtools 2008, Athens, Greece, 2008.

Papers under revision

- **J. F. Pérez** and B. Van Houdt. *The effect of Partial Conversion and Fiber Delay Lines in an OBS switch with a large number of wavelengths.* August 2009.
- **J. F. Pérez** and B. Van Houdt. *The M/G/1-type Markov chain with restricted transitions and its application to queues with batch arrivals.* November 2009.
- **J. F. Pérez** and B. Van Houdt. *Quasi-Birth-and-Death processes with restricted transitions and its applications.* Submitted to Performance Evaluation, special issue on QEST 2009. February 2010.
- **J. F. Pérez** and B. Van Houdt. *A Mean Field Model for an Optical Switch with a large number of wavelengths and centralized partial conversion.* March 2010.

Nederlandse Samenvatting

In dit proefschrift richten we ons op drie thema's: gestructureerde Markov ketens, optische grids en schakelaars. Gestructureerde Markov ketens vormen een krachtige klasse van modeleer tools voor het analyzeren van stochastische systemen. Heel wat van deze Markov ketens worden gekenmerkt door een blok transitiematrix met een repeterende structuur die kan worden benut om zijn stationaire vector efficiënt te berekenen, het gaat hierbij om de zogenaamde Quasi-Birth-Death, M/G/1- en GI/M/1-type Markov ketens. In het bijzonder beschouwen we het geval waarin de blokken waaruit de transitiematrix is opgebouwd eveneens een interne structuur hebben en ontwikkelen we snelle algoritmen die deze extra structuur uitbuiten bij de berekening van de stationaire vector. Deze extra interne structuur treedt vaak op bij het analyseren van wachtrijmodellen die veelvuldig gebruikt worden voor het bepalen van de prestatiematen van communicatie-systemen.

Het tweede onderwerp in dit proefschrift is de analyze van optische grid netwerken. Deze netwerken verbinden eindgebruikers met rekencentra, die zelf met elkaar verbonden zijn. Het meest kenmerkende aspect van een grid is dat de eindgebruikers geen voorkeur hebben bij de keuze van het rekencentra, zoals die hun verzoek maar tijdig uitvoeren. Wanneer een centrum zelf over onvoldoende capaciteit meer beschikt om een inkomende taak te verwerken, dan zal deze taak doorgestuurd worden naar één van de overige centra. De eerste toepassingen van Grid netwerken situeerden zich voornamelijk in die onderzoeksdomeinen die de analyze van grote hoeveelheden van data met zich mee brachten, zoals de astrofysica, deeltjes fysica, scheikunde en medische biologie.

Het laatste onderwerp van dit proefschrift betreft het modeleren en analyzeren van optisch geschakelde technologieën. Een optische schakelaar kan een inkomend signaal meestal in het optische domein verwerken. Op deze wijze kunnen opto-elektronische vertalingen, die nodig zijn wanneer een optisch signaal binnenkomt in een elektromagnetische schakelaar, worden vermeden. Dit maakt dat optisch schakelen een oplossing kan bieden voor het backbone-netwerk, waar de schakelaars het steeds moeilijker krijgen om aan de steeds toenemende snelheid van de inkomende optische vezels te voldoen. We zijn vooral geïntereseerd in de analyse van "contention resolution" strategieën in een optische schakelaar. In dit type schakelaar ontstaat er contention wanneer twee of meer pakketten een transmissie via dezelfde uitgangspoort en dezelfde golflengte aanvragen. Er zijn twee belangrijke manieren waarop contention resolution in het optische domein kan gebeuren: optische buffering en golflengte conversie. We beschouwen drie verschillende architecturen voor optische schakelaars die deze vormen van contention resolution ondersteunen. Voor elk van deze architecturen analyzeren we het effect van enerzijds het ontwerp en anderzijds de eigenschappen van het netwerkverkeer op de prestatie van de optische schakelaar.