

STOCHASTIC MODELING WITH CONTINUOUS FEEDBACK MARKOV FLUID QUEUES

A DISSERTATION SUBMITTED TO
THE DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By
Mehmet Akif Yazıcı
January, 2014

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Assoc. Prof. Dr. Nail Akar (Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Prof. Dr. Ezhan Karařan

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Prof. Dr. Tuęrul Dayar

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Prof. Dr. Erdal Arıkan

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Prof. Dr. Murat Alanyalı

Approved for the Graduate School of Engineering and Science:

Prof. Dr. Levent Onural
Director of the Graduate School

ABSTRACT

STOCHASTIC MODELING WITH CONTINUOUS FEEDBACK MARKOV FLUID QUEUES

Mehmet Akif Yazıcı

Ph.D. in Electrical and Electronics Engineering

Supervisor: Assoc. Prof. Dr. Nail Akar

January, 2014

Markov fluid queues (MFQ) are systems in which a continuous-time Markov chain determines the net rate into (or out of) a buffer. We deal with continuous feedback MFQs (CFMFQ) for which the infinitesimal generator of the background process and the drifts in each state are allowed to depend on the buffer level through continuous functions. Explicit solutions of CFMFQs for a few special cases has been reported, but usually numerical methods are preferred.

A numerically stable solution method based on ordered Schur decomposition is already known for multi-regime MFQs (MRMFQ). We propose a framework for approximating CFMFQs by MRMFQs via discretizing the buffer space. The parameters of the CFMFQ are approximated by piecewise constant functions. Then, the solution is obtained by block-tridiagonal LU decomposition for the related MRMFQ. Moreover, we describe a numerical method that enables us to solve large scale systems efficiently.

We model basically two different stochastic systems with CFMFQs. The first is the workload-bounded MAP/PH/1 queue, to which the arrivals occur according to a workload-dependent MAP (Markovian Arrival Process), and the arriving job size distribution is phase-type. The jobs that would cause the buffer to overflow are rejected partially or completely. Also, the service speed is allowed to depend on the buffer level. As the second application, we model the horizon-based delayed reservation mechanism in Optical Burst Switching networks with or without fiber delay lines. We allow multiple traffic classes and the effect of offset-based and FDL-based differentiation among traffic classes in terms of burst blocking is investigated.

Lastly, we propose a distributed algorithm for air-time fairness in multi-rate WLANs that overcomes the performance anomaly in IEEE 802.11 WLANs. We also give a stochastic model of the proposed model, and provide a novel and elaborate

proof for its effectiveness. We also present an extensive simulation study.

Keywords: Continuous feedback Markov fluid queues, Block-tridiagonal LU decomposition, Workload-bounded buffer, Horizon-based reservation, Air-time fairness.

ÖZET

SÜREKLİ GERİBESLEMELİ MARKOV AKIŞKAN KUYRUKLARLA RASSAL MODELLEME

Mehmet Akif Yazıcı

Elektrik Elektronik Mühendisliği, Doktora

Tez Yöneticisi: Assoc. Prof. Dr. Nail Akar

Ocak, 2014

Markov akışkan kuyrukları (MAK), bir kuyruğun dolma/boşalma hızının, sürekli zamanlı bir Markov zinciri tarafından belirlendiği sistemlerdir. Bu çalışmada, sürekli geribeslemeli MAK'lar (SGMAK) ön planda çalışılmıştır. Bu sistemlerde arkaplan sürecinin üretici ve kuyruğun dolma/boşalma hızı, sürekli fonksiyonlarla kuyruk doluluğuna bağlıdır. Çok özel bazı durumlarda analitik olarak çözülebilen bu sistemlerde genellikle sayısal yöntemler tercih edilir.

SGMAK'ları çoklu rejimli MAK'lar (ÇRMAK) ile yaklaşıklayıp, ÇRMAK'lar için literatürde var olan Schur ayrıştırmasına dayalı ve sayısal olarak kararlı olduğu bilinen yöntemi kullanmak üzerine bir çerçeve sunuyoruz. Bu yöntemde, SGMAK parametreleri parçalı sabit fonksiyonlarla yaklaşıklanarak bir ÇRMAK elde edilir. Bu ÇRMAK, blok-üç bant köşegen LU ayrıştırması kullanılarak çözülebilir. Bunun yanı sıra, çok büyük sistemleri zaman açısından verimli bir biçimde çözebilen sayısal bir yöntem önermekteyiz.

SGMAK kullanarak iki değişik sistem modellemekteyiz. Bunlardan ilki, işyükü bağımlı MAP/PH/1 kuyruklardır. Paket gelişi MAP, gelen iş yükü uzunluğu dağılımı da faz tipidir. Kuyruğa tam olarak sığmayan paketler tamamen veya kısmen reddedilir. Ayrıca, kuyruğun sunucu hızı da kuyruk doluluğuna bağlı olabilir. Modellediğimiz ikinci sistem, optik çoğuşum ağlarında ufuk parametresi tabanlı kaynak tahsisi yönteminin fiber gecikme hatları varlığı ya da yokluğundaki davranışdır. Çoklu trafik sınıflarını da hesaba kattığımız model kullanılarak gecikme zamanı ve fiber gecikme hatlarına dayalı servis kademelendirmesi metotları incelenmektedir.

Son olarak, IEEE 802.11 kablosuz ağlarında yaşanan başarım anomalisi sorununa kanal zamanı adaleti sağlayarak çözüm getiren dağıtımlı bir algoritma önermekteyiz. Bu metodun çalıştığı, Markov zinciri tabanlı bir isbatla ve kapsamlı bir benzetim çalışmasıyla teyit edilmektedir.

Anahtar sözcükler: Sürekli geribeslemeli Markov akışkan kuyrukları, blok-üç bant köşegen LU ayrıştırması, işyükü sınırlı kuyruk, ufuk parametresi tabanlı kaynak tahsisi, kanal zamanı adaleti.

Acknowledgement

I am grateful to my advisor, Dr. Nail Akar, for his guidance throughout my studies. I also would like to thank the members of my thesis monitoring committee, Dr. Ezhan Karařan and Dr. Tuęrul Dayar for their invaluable input.

I acknowledge the financial aid provided by TBİTAK as a monthly scholarship, as well as travel grants provided for my attendance to the SİU 2008, SİU 2013, and ITC 2013 conferences under the TBİTAK project EEEAG-111E106.

I would like to thank my family, and my wife's family for their continued support and patience during my studies. I dedicate this thesis to two special ladies, my lovely wife Elif Nur, and my grandmother Huriye Yazıcı.

Contents

1	Introduction	1
1.1	Stochastic Modeling and Markov Fluid Queues	1
1.2	The Workload-Bounded MAP/PH/1 Queue	8
1.3	Modeling Horizon-based Reservation in OBS Networks	11
1.4	Air-time Fairness in Multi-rate WLANs	16
1.5	Contribution Summary and Organization	24
2	Markov Fluid Queues	27
2.1	Introduction	27
2.2	Single-Regime Markov Fluid Queues	31
2.2.1	Boundary Conditions	33
2.2.2	Spectral Solution	34
2.2.3	Ordered Schur Form	36
2.2.4	Additive Decomposition Method	37
2.3	Multi-Regime Markov Fluid Queues	40

2.3.1	Solution of Multi-Regime Markov Fluid Queues	44
2.3.2	Efficient Solution of Boundary Conditions	46
2.3.3	Solution of MRMFQs with Temporarily-Absorbing States	49
2.4	Continuous Feedback Markov Fluid Queues	50
3	The Workload-Bounded MAP/PH/1 Queue	55
3.1	Introduction	55
3.2	Stochastic Model for the Workload-dependent MAP/PH/1 Queue . . .	57
3.2.1	Infinite Buffer (IB)	59
3.2.2	Finite Buffer with Partial Rejection (FB-PR)	64
3.2.3	Finite Buffer with Complete Rejection (FB-CR)	65
3.3	Numerical Examples	68
3.4	Conclusion	79
4	Analysis of Horizon-Based Reservation in OBS Networks	81
4.1	Introduction	81
4.2	Stochastic Model for Horizon-based Reservation with No FDLs	84
4.3	Numerical Examples for Horizon-based Reservation with No FDLs . . .	90
4.4	Stochastic Model for Horizon-based Reservation with FDLs	97
4.5	Numerical Examples for Horizon-based Reservation with FDLs	103
4.6	Conclusion	108

5 Achieving Air-time Fairness in Multi-rate IEEE802.11 WLANs	112
5.1 Introduction	112
5.2 Existing Methods for Air-time Fairness	114
5.3 Proposed MAC and Its Analytical Modeling	117
5.4 Air-time Fairness of MDCF for Non-integer N_i	120
5.5 Practical Aspects	126
5.6 Simulation Study	128
5.7 Conclusion	138
6 Conclusion	143

List of Figures

1.1	A sample path for an SRMFQ	3
1.2	A sample path for an MRMFQ	4
1.3	A sample path for a CFMFQ	4
1.4	A sample path for the M/M/1 queue	7
1.5	Complete and partial rejection policies	9
1.6	A sample scenario with JIT, Horizon and JET	12
1.7	The evolution of the horizon parameter	13
1.8	A sample scenario with FDLs	15
2.1	A sample path for an SRMFQ	29
2.2	A sample path for an MRMFQ	30
2.3	A sample path for an CFMFQ	31
2.4	The structure of an MRMFQ	41
3.1	An example for the background process of the CFMFQ	60
3.2	Buffer level pdf vs. Number of regimes	70

3.3	Structure of the matrix H for $K = 4$	71
3.4	Buffer level pdf for FB-PR and FB-CR under low loading	72
3.5	Buffer level pdf for FB-PR and FB-CR under high loading	73
3.6	Buffer level pdf for IB	75
3.7	Job loss probability plot for IPP scenario	76
3.8	Buffer level pdf for the system with MMPP arrivals	77
3.9	Buffer level pdf for the two class system in FB-CR setting	79
4.1	The transformed horizon	84
4.2	The transformed horizon	85
4.3	Horizon pdf plots for 3 different load values.	91
4.4	Burst blocking probability vs. number of regimes	92
4.5	Burst blocking probability: Low loads	92
4.6	Burst blocking probability: Moderate to high loads	93
4.7	Conditional burst blocking probability	93
4.8	The effect of shifting the offset time distribution	94
4.9	The effect of the offset time variation.	95
4.10	Overall blocking probability with two classes.	96
4.11	Ratios of the blocking probabilities of the two classes.	96
4.12	Blocking probability vs. Number of FDLs	104
4.13	Structure of the matrix H	104

4.14 The pdf of the horizon for a general case	105
4.15 The pdf of the horizon for scenario i.	106
4.16 The pdf of the horizon for scenario ii.	107
4.17 The pdf of the horizon for scenario iii.	107
4.18 QoS differentiation with FDL access limitation	109
5.1 A snapshot of the air-time utilization of a random access WLAN.	117
5.2 The Markov chain for the two-user scenario	121
5.3 Standard DCF vs. MDCF: Air-time fairness	131
5.4 Standard DCF vs. MDCF: Cumulative throughput	132
5.5 Standard DCF vs. MDCF: Channel utilization	134
5.6 Air-time fairness with non-integer N_i	135
5.7 Standard DCF vs. MDCF in large WLANs	135
5.8 Air-time utilization under stress conditions	140
5.9 MDCF vs. CW_{min} adaptation: Throughput ratios	140
5.10 MDCF vs. CW_{min} adaptation: Channel utilization	141

List of Tables

3.1	Workload loss probabilities for FB-PR	72
3.2	Workload and job loss probabilities for FB-CR	74
3.3	Run times vs. System size and number of regimes	78
4.1	Blocking probabilities for the three scenarios	106
5.1	Simulation parameters	129
5.2	Simulation scenario for Example IV	139
5.3	MDCF with frame aggregation	142

Chapter 1

Introduction

1.1 Stochastic Modeling and Markov Fluid Queues

Stochastic modeling deals with systems involving random events. From an engineer's perspective, it involves evaluating certain performance measures based on the statistical behavior of the system at hand. This usually has two main purposes: (i) to understand how the stochastic system behaves under certain conditions, and (ii) to be able to design the system, if possible, so that the performance measures are improved.

Queueing systems are one of the most important stochastic systems. Such systems are described by a number of parameters:

- The arrival process of the jobs (also called clients),
- The service time distribution, which has to do with the service rate of the server and/or the job size,
- The number of servers in the system,
- The amount of buffer space, which can be finite or infinite, and can be defined in terms of the number of jobs waiting to be serviced (in which case it is also called waiting room), or the amount of workload remaining,

- The queueing discipline such as First-Come-First-Served, or Processor Sharing.

Queueing theory, which studies queueing systems, is an important tool for communications networks engineering. Queueing systems are encountered in almost every topic concerning communications networks from telephone networks and satellite communications to cellular communications, optical networks and the Internet. The focus of this thesis is on a more specific class of queueing systems, namely the *Markov fluid queues* (MFQ) [1, 2].

Markov fluid queues are stochastic processes with two components:

- (i) The background process: This is a continuous-time Markov chain (CTMC) with a finite state space.
- (ii) The buffer level: For each state of the background process, there is a drift rate that the buffer is filled (or depleted if the drift is negative). The buffer capacity can be finite or infinite.

Three categories of MFQs are considered in this thesis:

1. Single-regime Markov fluid queues (SRMFQ): These are the simplest type of MFQs. The term *single-regime* means that the behavior of the MFQ is the same throughout the whole buffer space. The infinitesimal generator of the background process is a constant matrix, and the drifts for each state is a fixed quantity whatever the buffer level is.
2. Multi-regime Markov fluid queues (MRMFQ): These queues have *multiple regimes* in the sense that there are a number of portions of the buffer level and in each of these regimes, the behavior of the fluid queue is different from the others. Within each regime, the infinitesimal generator of the background process and the drifts for each state are still constant, but they vary from regime to regime. Therefore, there is a feedback from the buffer level on the behavior of the queue.

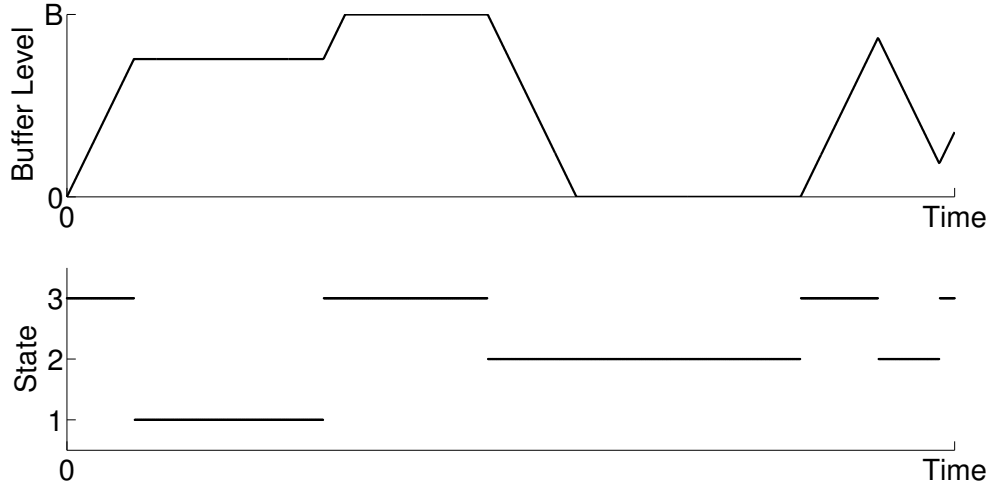


Figure 1.1: A sample path for the buffer level of an SRMFQ with three states. The drifts in states 1, 2 and 3 are 0, -1 and 1 respectively.

3. Continuous feedback Markov fluid queues (CFMFQ): The behavior of CFMFQs has a continuous dependence on the buffer level. The infinitesimal generator of the background process and the drifts for each state are allowed to be functions of the buffer level. Moreover, these functions are not piecewise-constant as in MRMFQ.

Now, we will give an example for each category to make the distinction clearer. A sample path for the buffer level of an SRMFQ with three states is given in Figure 1.1. The drifts in states 1, 2 and 3 are 0, -1 , and 1 respectively, and they remain constant throughout the whole buffer space. Moreover, the infinitesimal generator of the background process is also constant, although it is hard to infer that from this sample path only.

In Figure 1.2, a sample path for the buffer level of an MRMFQ with two states and two regimes is given. b is the regime boundary between regimes 1 (the buffer portion between 0 and b) and 2 (the buffer portion between b and B). It is obvious that drifts for each state vary between regimes but remain constant within each regime. Also, there are more state transitions in regime 2, suggesting that the infinitesimal generator of the background process also varies with regime.

Lastly, Figure 1.3 shows a sample path for the buffer level of a CFMFQ with two

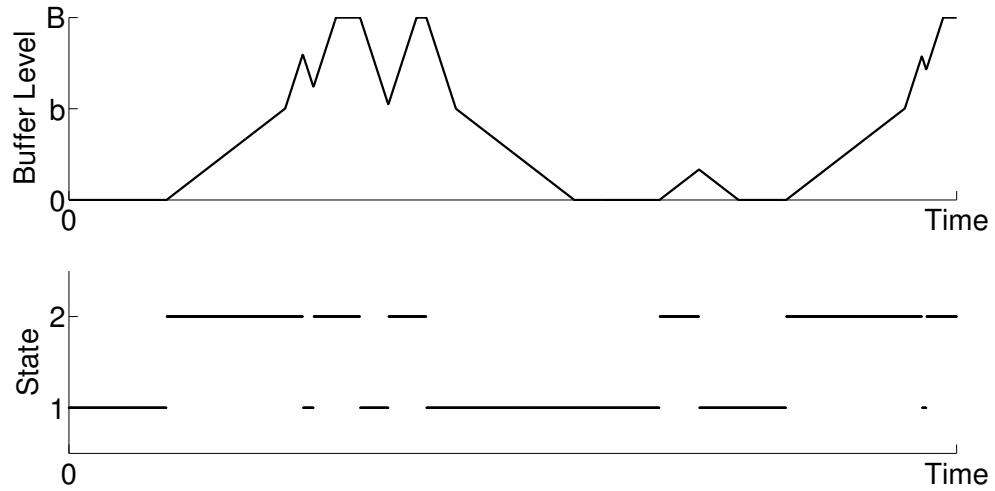


Figure 1.2: A sample path for the buffer level of an MRMFQ with two states and two regimes.

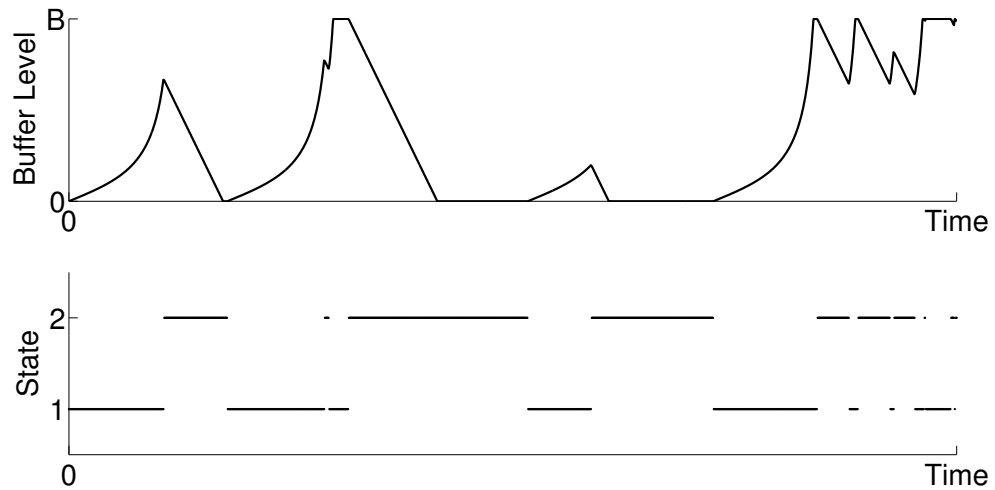


Figure 1.3: A sample path for the buffer level of a CFMFQ with two states. The drift in state 1 is a function the buffer level.

states. The drift in state 1 is a function the buffer level, whereas the drift in state 2 is a constant quantity. Moreover, state transitions get more frequent as the buffer level approaches the upper boundary B , meaning that the behavior of the background process also depends on the buffer level. Note that the dependence of only one parameter on the buffer level is sufficient for an MFQ to be classified as a CFMFQ. For instance, if the infinitesimal generator of the background process is fixed throughout the buffer space and all the drifts but one are constants, this system is still a CFMFQ.

The study of SRMFQs that was pioneered by the works of Anick et al. [1] and Kosten [3] has been around for more than three decades. The spectral approach to the solution of SRMFQs in its most general sense with finite or infinite buffer capacity is laid out in the work of Kulkarni [2]. This work also touches on MRMFQs, and even suggests approximating non-step dependence of the drifts on the buffer level with step functions, which is a starting point to the framework we present with this thesis. In the work of Akar and Sohraby [4], the numerical shortcomings of the spectral solution is identified, and a novel method, the additive decomposition method is proposed.

MRMFQs have been studied in different contexts, and they are also referred to as “level dependent” [5], “multi-layer” [6],[7] or “multi-threshold” [8]. In the work of Mandjes et al. [9], the spectral approach for SRMFQs is extended to solve MRMFQs. Kankaya and Akar [10] employ the additive decomposition method to provide a general solution to MRMFQs. Further contributions of this paper include incorporating the ordered Schur decomposition into the additive decomposition method, relaxing the set of assumptions on the MRMFQ to obtain a framework that supports new types of boundaries and giving the whole framework (including the differential equation system and the boundary conditions) in terms of the steady-state joint pdf vector of the buffer. When the problem is expressed in the pdf form, the boundary conditions can be represented in a shortened way.

CFMFQs have been formulated in the paper by Scheinhardt et al. [11]. However, the solution of CFMFQs is quite complex. Scheinhardt et al. [11] give explicit solution for only a system with two states and recommend numerical methods in [12] for

more complex systems. German et al. [13] gives a numerical method based on series expansion for solving CFMFQs. There is an important assumption in this study: the drift rates do not ever change sign. This avoids probability mass accumulations within the buffer space. Building on this study, the paper by Gribaudo and Telek [14] relaxes this assumption by defining boundaries at points of drift sign changes. The common denominator in all these studies is that they treat the problem as a boundary value problem and apply numerical methods known for such problems. In contrast, we approximate the continuous dependence of the drifts and the background process on the buffer level as stepwise-constant functions. This effectively means approximating a CFMFQ with an MRMFQ. Then, the existing methods for solving MRMFQs can be employed to obtain the solution to the CFMFQ. For this purpose, we have preferred the method by Kankaya and Akar [10]. In this way, we keep the problem within the fluid queue framework rather than delving into the boundary value problems setting.

In order to obtain a sufficiently accurate solution to the CFMFQ problem with the MRMFQ approximation, the number of regimes should be as large as possible. In this way, the difference between the original functions that describe the CFMFQ parameters and their stepwise-constant counterparts is minimized. This means solving large scale MRMFQs. The original study by Kankaya and Akar [10] make the implicit assumption that size of the problem in the number of regimes is small. Even though they present an example with 2^{10} states [10, pp. 442-3], the number of regimes in this example is 2 and there is no mention of systems with large number of regimes.

As will be seen later, the solution of fluid queues involves solving a linear system of equations. In the case of MRMFQs, the size of this linear system is in the order of the number of states multiplied by the number of regimes. It is well known that classical solutions such as Gaussian elimination to linear systems of equations with size n require number of operations in the order of n^3 [15, pp. 98-100]. This fact becomes prohibitive to the increase of the number of regimes. Fortunately, the structure of the linear system of equations turns out to be block-banded. It is possible to exploit this structure to reduce the time complexity of the operation. We will

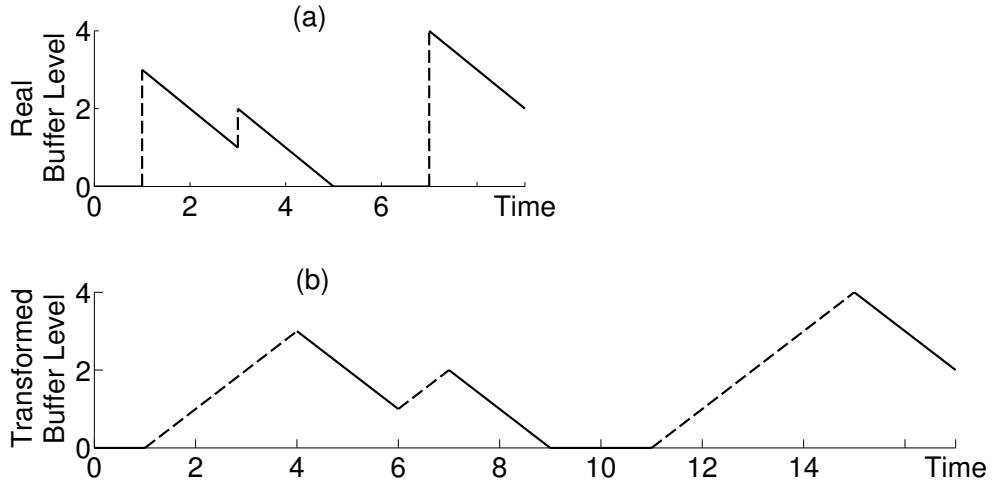


Figure 1.4: A sample path for the M/M/1 queue buffer level and its transformed counterpart. Arrivals occur at times 1, 3 and 7 with sizes 3, 1 and 4 respectively.

describe a method based on the block-tridiagonal LU factorization to take advantage of the structure of the linear system of equations. The time complexity of this method is linear in the number of regimes as opposed to cubic. Therefore, it is possible to use quite large numbers of regimes. There are examples in this thesis with numbers of regimes as large as 2^{14} .

To close this section, we will describe the procedure to model stochastic systems that involve jumps [16]. Consider the very simple system of the M/M/1 queue; the inter-arrival and service times are exponentially distributed. Assume that the service rate is constant, and the job sizes are exponentially distributed, hence the exponentially distributed service times. A sample path of this queue is given in Figure 1.4(a). This system does not readily lend itself to fluid queue analysis as there are abrupt jumps at arrival epochs. If the jumps are replaced with linear ascents of slope 1 having durations equal to the size of the arriving job, we obtain the transformed process as demonstrated in Figure 1.4(b). Now, the transformed system can be described by an SRMFQ with two states, one for the linear ascents and one representing the normal operation of the queue. The rate out of the first state is equal to the rate parameter of the job size distribution, and the rate out of the second state is equal to the arrival rate. After this SRMFQ is solved, the distribution of the real M/M/1 queue can be obtained by censoring out the first state. In this manner,

systems with jumps can be modeled using MFQs.

1.2 The Workload-Bounded MAP/PH/1 Queue

The first application of the CFMFQ that we will present in this thesis is the *workload-bounded MAP/PH/1 queue*. This is a single-server queuing system in which the job arrivals are modeled by a workload-dependent Markovian Arrival Process (MAP). A workload-dependent MAP differs from an ordinary MAP [17],[18] by its matrix parameters not being fixed but allowed to vary with the instantaneous buffer level. The workload brought by an individual job, namely the job size, has a phase type (PH-type) distribution. The queue service discipline is FIFO (first-in-first-out). The queue is drained at a rate $c(x)$ when the buffer level takes the value $x > 0$. In the infinite queue capacity case, a new job arrival is always admitted and it increases the buffer level (or workload) by the job size. We will also present the case of finite queue capacity. Although most finite queue capacity models pose a limit on the maximum number of jobs allowed in the system, the interest in this study will be in models in which there is an upper limit on the overall workload that the buffer can hold, say B . Such buffers are called *workload-bounded* in which case different policies can take action depending on what to reject when the workload limit gets to be exceeded:

- Partial rejection policy: If the current workload plus the job size of an arriving job exceeds the workload capacity B , then the workload is increased up to B , which amounts to rejecting part of the arriving job.
- Complete rejection policy: The job is completely rejected in the same situation.

The complete rejection policy is especially of importance since it models the exact behavior of the queues found in communication networks. In these systems, packets cannot be accepted partially as the packet chunks would be useless anyway due to obvious reasons (check-sum error, loss of header and/or trailer). For a discussion

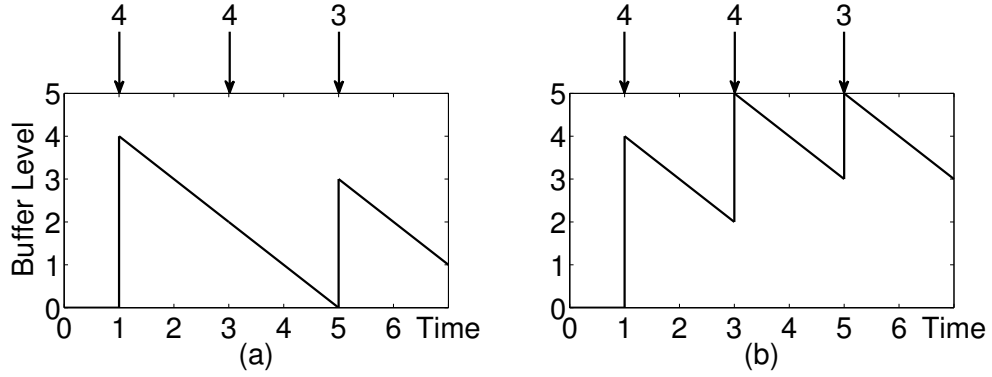


Figure 1.5: A sample scenario with (a) complete and (b) partial rejection policies. The buffer capacity is 5. Three arrivals occur at times 1, 3 and 5 with job sizes 4, 4 and 3 respectively.

of various rejection policies for finite buffer systems, refer to the paper by Perry and Asmussen [19].

A sample scenario with complete and partial rejection policies is presented in Figure 1.5. The buffer capacity is 5. Three arrivals occur at times 1, 3 and 5 with job sizes 4, 4 and 3 respectively. Under complete rejection policy, the first and the third arrivals are accepted into the buffer whereas the second one is rejected completely. Under partial rejection policy, the first arrival is accepted entirely. However, the second and the third arrivals do not fit into the buffer completely. One unit of workload apiece are lost from each arrival.

The goal of this part of the study is the numerical calculation of the steady-state distribution of the system workload in the infinite and finite queue capacity scenarios, the latter for both rejection policies. Other performance measures of interest including job loss probability, workload loss probability, etc., can then be derived from this distribution. The main method we propose to find the steady-state distribution of the workload-dependent MAP/PH/1 queue comprises the following three main steps:

- (i) The workload-dependent MAP/PH/1 queue for the infinite and finite queue capacity cases is described by a CFMFQ using sample path arguments. In the finite queue capacity case, this is done for both partial and complete rejection

policies.

- (ii) The resulting CFMFQ is approximated by an MRMFQ using discretization.
- (iii) The boundary conditions for this MRMFQ are solved using block-tridiagonal LU factorization [15] to obtain the steady-state distribution of the queue occupancy.

For related work, Bekker et al. [20] study an M/G/1 queue with workload-dependent arrivals and service rates for the infinite queue capacity case. The workload-bounded M/G/1 buffer under complete rejection policy was studied in the work by Perry et al. [21] with closed form expressions for the M/M/1 case. Bekker [22] studies M/G/1 queues with finite buffers with workload-dependent arrival rate, service speed, and both partial and complete rejection policies. Level crossings and Volterra integral equations play a key role in [22] in which closed-form expressions are also given. The goal of our study is to extend the model of Bekker [22] to allow a more general arrival process, namely MAP, and develop a numerically stable and computationally efficient algorithm to solve the steady-state workload distribution.

On the other hand, Sharma and Virtamo [23] investigate a workload-bounded buffer using complete rejection policy with MMPP (Markov Modulated Poisson Process) arrivals which is a sub-case of MAP. However, neither the MMPP nor the service speed is allowed to depend on the workload in this work. The model in [23] has also been extended to systems with multiple priority classes in [24]. Multi-class MAP arrivals with workload-dependent acceptance policies are recently studied in the work of Horváth and van Houdt [25] in the context of modeling customer impatience. The arrivals to the system studied in [25] are of an adaptive Markovian arrival process with marked customers, the adaptiveness stemming from the different state transition rates depending on whether the arriving job enters the system or not according to an impatience model. The impatience is allowed to be a continuous function, which is then discretized into a piecewise-constant function. The system is modeled as an MRMFQ and the boundary conditions are solved efficiently by exploiting the block-tridiagonal structure of the resulting matrix. In comparison, our model solves a more general system in which the workload-dependency is potentially intrinsic to the system at hand rather than being a result of customer

impatience. Moreover, we also let the service speed to be a function of the workload. Although these generalizations do not bring further complications in terms of the numerical algorithm, i.e., discretizing the buffer space and solving a block-tridiagonal matrix equation, we provide the framework for workload-dependent MAP/PH/1 queues in the most general setting, and provide the mathematical model for queues in which jobs that do not fit to the available buffer space are rejected in their entirety, which is a novelty in its own accord.

1.3 Modeling Horizon-based Reservation in OBS Networks

Optical Burst Switching (OBS) [26] has been the focus of research in the field of optical networks since it offers a compromise between optical packet switching that has yet to be implemented due to lack of optical buffers, and the relatively IP-unfriendly optical circuit switching. Recognizing the inherent burstiness of broadband multimedia traffic, in the OBS paradigm, a number of packets are merged into a single payload, called a burst. A control packet which is called the Burst Control Packet (BCP) is sent in the electronic domain in advance of the burst with relevant information in order to reserve resources for the burst. Optical nodes receiving the BCP configure themselves to accommodate the burst to arrive, if possible. The burst is then sent in the optical domain after an amount of time, called the offset time.

There are different reservation mechanisms for node configuration. The simplest is the Just-in-Time (JIT) [27] mechanism that reserves a node for an incoming burst and rejects all arrivals until the burst is entirely transmitted. With this *immediate reservation* method, bursts that would arrive after the node becomes available may be blocked if their BCPs arrive when the node is busy. On the other hand, *delayed reservation* methods such as Just-Enough-Time (JET) [28, 26] and Horizon [29] keep track of the time that the node will become idle, allowing bursts that will arrive later than this value (scheduling horizon) to be accommodated. Horizon is easier to implement than JET that supports void-filling, which is the process of allocating the idle times between consecutive reservations to bursts that can fit in.

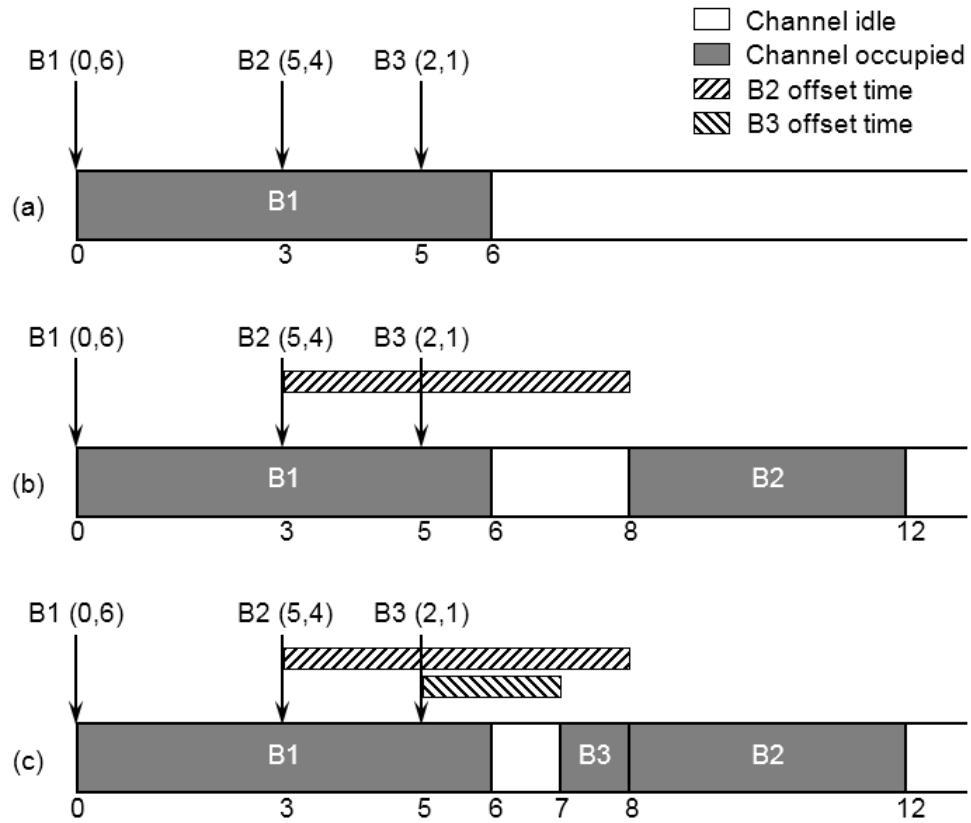


Figure 1.6: A sample scenario with the three reservation mechanisms in action: (a) Just-in-Time, (b) Horizon, (c) Just-Enough-Time. Numbers in the parenthesis represent the offset time and the burst length, respectively.

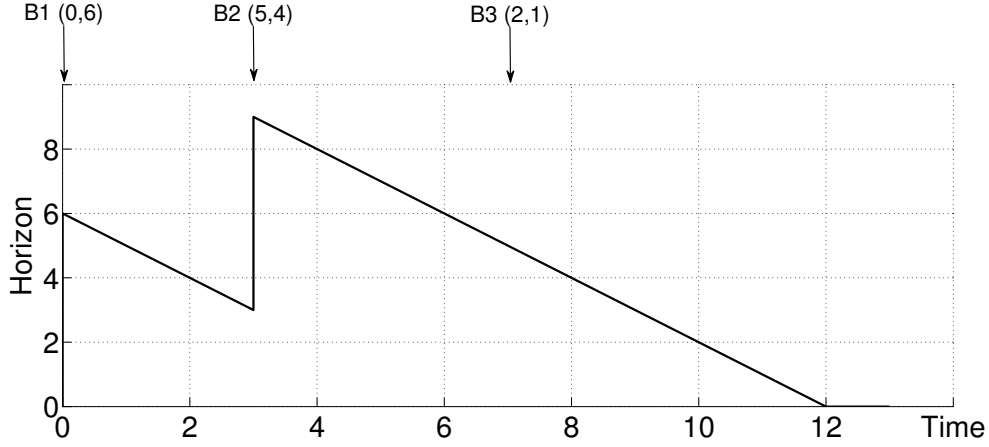


Figure 1.7: The evolution of the horizon parameter for the scenario in Figure 1.6. B1 and B2 are accepted whereas B3 is rejected since its offset parameter, 2, is less than the horizon value upon the arrival of its BCP.

A sample scenario with the three reservation mechanisms is depicted in Figure 1.6. Three BCPs are received at times 0, 3, and 5 associated with bursts B1, B2, and B3, respectively. The offset times of the bursts are 0, 5, and 2; and the burst lengths are 6, 4, and 2, respectively. Under JIT operation depicted in Figure 1.6(a), B1 is accepted. As soon as the setup packet for B1 is received, the channel is reserved. As a result, the channel is reserved when the setup packets for B2 and B3 arrive. As JIT is an immediate reservation mechanism and the channel is already reserved, B2 and B3 are rejected, even if they are to arrive after the channel becomes free.

With Horizon-based reservation, which is demonstrated in Figure 1.6(b), B1 is accepted. The horizon parameter is updated to 6 and starts decreasing with time. When the BCP for B2 arrives at time 3, the horizon parameter is 3, which is less than the offset time declared in the BCP. Therefore, B2 is accepted and the horizon parameter is updated to 9, which is the sum of the offset time and the length of B2. At time 5, the BCP for B3 arrives and the horizon parameter has the value 7 at this point. Therefore, B3 is blocked. The evolution of the horizon parameter is plotted in Figure 1.7. JET, on the other hand, has a void-filling mechanism. Therefore, B3 is accepted under JET operation, given in Figure 1.6(c), and is served before B2 even if its BCP arrives later than that of B2.

The offset time can also be used as a means of providing loss probability-based

service differentiation. By assigning larger offset times to high priority clients, their burst blocking probability can be reduced. Yoo and Qiao [30] propose such a scheme and provide a method for finding the required offset time for a certain level of isolation between different priority classes. Other service differentiation techniques such as burst segmentation [31], deflection routing [32] and preemption [33] have also been proposed.

An important problem in OBS networks is the contention among multiple bursts that arrive at a node on the same wavelength within each other's duration. One solution to this problem could be wavelength converters in multi-wavelength links. Using wavelength converters, an incoming burst can be directed to a different wavelength channel from which it is arriving on, since the node it arrives is already transmitting another burst on that channel [34]. However, we consider single-wavelength links in this study. In such links, a well-known solution to contention is fiber delay lines (FDL) [35]. FDLs are basically coils of fiber that induce a fixed amount of delay on a burst that traverses it. In case of contention between say two bursts, one of the contending bursts is chosen to be transmitted right away (or it is being transmitted already), and the other one is "stored" within an FDL, i.e. it is fed to an FDL so that its arrival is delayed. Obviously, the additional delay imposed on this burst should be long enough to ensure that when it comes out of the FDL, the transmission of the other burst is completed and the channel is available.

For better performance, a number of FDLs can be available at an optical node. These FDLs can be configured to provide degenerate buffering or non-degenerate buffering. In degenerate buffering, each FDL provides an integer multiple of a fixed delay. In other words, if there are N FDLs, the delay line i , $1 \leq i \leq N$, provides a delay of $i\Delta$, where Δ is a fixed quantity and called the granularity parameter. On the other hand, in non-degenerate buffering, the delays each FDL provides can be arbitrary.

A sample scenario with the horizon-based reservation mechanism in the presence of FDLs is given in Figure 1.8. Four BCPs are received at times 0, 2, 4, and 6, associated with bursts B1, B2, B3, and B4, respectively. The offset times of the bursts are 1, 1, 1, and 0; and the burst lengths are 3, 4, 3, and 3, respectively. B1 is accepted and occupies the channel between times 1 and 4. Within this time the BCP for B2

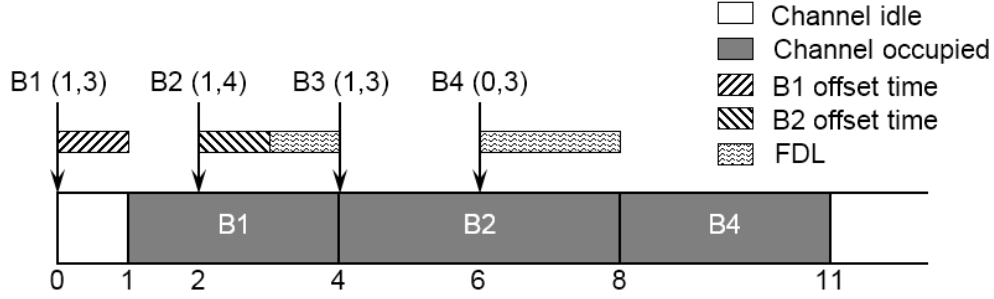


Figure 1.8: A sample scenario with the horizon-based reservation mechanism in the presence of FDLs. Numbers in the parenthesis represent the offset time and the burst length respectively. Two FDLs are available with delays 1 and 2.

arrives at time 2 with an offset value of 1, indicating that B2 is to arrive at time 3. However, the channel is occupied by B1 at time 3. Therefore, B2 is channeled into the FDL with delay 1 upon its arrival. As B2 leaves the FDL, the channel becomes idle and B2 is transmitted. The BCP associated with B3 arrives at time 4 with an offset time of 1, meaning that B3 will be arriving at time 5. The channel is being occupied by B2 until time 8, so even with the delay of 2 time units that can be provided by the FDL, B3 cannot be accommodated. Therefore, it is blocked. B4 arrives at time 6 with no offset time. Since the channel is busy transmitting B2 until time 8, B4 is channeled into the FDL with delay 2, and is transmitted at time 8.

The burst blocking probability in OBS networks are studied extensively in different settings. Ref. [36] analyzes JET with generally distributed burst lengths and deterministic offset times under low blocking assumption. Morató et al. [37] investigate the blocking time distribution in the existence of FDLs with multiple wavelength channels. In [38], an M/G/k/k approximation is used to analyze a JET system with exponential burst sizes and complete isolation between the multiple classes. Ref. [39] gives a comparison of JIT, JET, and horizon-based reservations based on the Erlang-B loss formula. A similar analysis is carried out in [40]. A slotted OBS-JET approximate model is solved using a non-homogeneous Markov chain in [41]. A JET system with uniformly distributed offset times and deterministic burst lengths is analyzed in [42]. All of these studies assume Poisson burst arrivals. In [43], the horizon reservation scheme is studied for a single-channel system with Poisson arrivals, PH-type distributed burst lengths and deterministic offset times. The analysis

is based on Markov fluid queues similar to our study. Moreover, as pointed out in [43], analyses based on the Erlang-B loss formula such as [38, 39, 40] cannot reflect the effects of higher order statistics of the burst lengths, or the offset times.

The study of the horizon-based reservation that will be presented in this thesis comprises two steps:

- (i) The horizon reservation scheme on a single-channel OBS system is investigated. The offset times are allowed to have general distributions. The steady-state distribution of the horizon parameter is solved, and the blocking probability is computed. This analysis is also extended to multiple traffic classes with different arrival processes, and burst length and offset time distributions.
- (ii) The horizon reservation scheme is investigated in the presence of FDLs. This analysis is again extended to multiple traffic classes. Each class is assumed to have access to different sets of FDLs and the effect of this setting on the QoS in terms of blocking is observed.

1.4 Air-time Fairness in Multi-rate WLANs

The IEEE 802.11 Working Group publishes the most widely deployed suite of protocols for Wireless Local Area Networks (WLAN). On the Medium Access Control (MAC) side, IEEE 802.11 employs a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) MAC protocol with binary exponential back-off, known as Distributed Coordination Function (DCF) [44]. DCF defines a mandatory basic access mechanism and an optional Request-To-Send/Clear-To-Send (RTS/CTS) mechanism which is less often used in practice. The focus of our study is on the basic access mechanism in which an 802.11 node with a frame to transmit listens to the channel first to detect an idle period of length at least equal to the Distributed Inter-Frame Space (DIFS). The node then sets its back-off timer value to an integer that is uniformly chosen in the interval $[0, CW - 1]$, where CW is set to the minimum contention window size, CW_{min} , at the first transmission attempt. The back-off timer is then decremented at each slot as long as the channel is idle whereas it is stopped

when a transmission is detected on the channel. Re-activation of the timer upon a transmission detection is done after the channel is sensed idle after this transmission for at least a DIFS. The back-off timer hitting zero triggers the frame's first transmission. Once the destination host successfully receives the frame, it transmits an acknowledgment frame (ACK) after a short inter-frame space (SIFS) time. If the transmitting node does not receive an ACK within a specified ACK timeout for the transmitted frame, a collision is said to have taken place. Upon each collision, CW is doubled until a maximum contention window size CW_{max} value is reached and the above back-off mechanism is repeatedly applied at each unsuccessful transmission.

Physical layer enhancements to the original 802.11 standard [44] made it possible to support raw data rates up to 54 Megabits per second (Mbps) [45],[46]. Despite the substantial increases in raw data rates for WLANs, since the used MAC (Medium Access Control) is the same, the actual throughput is much lower due to 802.11 overhead whose reduction is crucial for IEEE 802.11 standards to achieve higher throughputs [47]. Novel MAC-layer techniques besides PHY-layer enhancements have been explored in the IEEE 802.11n working group to reduce overhead so as to achieve a throughput surpassing 100 Mbps [48]. Frame aggregation in which multiple frames are aggregated and transmitted at a single transmission opportunity as a burst is one such technique to reduce overhead [49].

IEEE 802.11 standards support multiple raw data rates and hence such networks are called multi-rate WLANs. As an example, the IEEE 802.11b supports data rates in the set $\{1, 2, 5.5, 11\}$ where the IEEE 802.11a standard supports data rates in the set $\{6, 9, 12, 18, 24, 36, 48, 54\}$, all rates being in units of Mbps [45],[50]. Moreover, the 802.11 standards support link adaptation by which a host selects one of the available transmission rates at a given transmission opportunity based on channel conditions and/or application traffic type. Various link adaptation algorithms are developed to increase throughput and vendors use proprietary link adaptation algorithms [51]. Although link adaptation appears to be a powerful means to enhance throughput in multi-rate WLANs, its effective use in multi-user 802.11 WLANs has been shown to be limited [52]. To explain, consider a scenario of multiple hosts with a higher raw bit rate in addition to a single host with a lower bit rate as used in [52] with all frame sizes assumed to be the same. Since the CSMA/CA algorithm of DCF provides

the same equal channel access probability to all hosts, the throughput of high rate hosts will be the same as the slow host. Therefore, DCF penalizes fast hosts and instead favors the slow host. This artifact is known as the performance anomaly problem of 802.11 DCF, which impedes a direct relationship between the raw data rate and the actual throughput in scenarios with multiple users with different data rates [52]. Actually, DCF is throughput-fair when frame sizes used by different nodes are the same on the average. Time-based fairness is proposed in [53],[54],[55] as an alternative to throughput fairness to cope with the performance anomaly problem. With time-based fairness, each competing node receives an equal share of the wireless channel occupancy time, i.e., air-time. A system achieving time-based fairness is called air-time fair. When air-time fair mechanisms are employed, the throughput of an individual node becomes strictly proportional with its raw bit rate and therefore high rate nodes will no longer be dragged down by slower ones, which leads to significantly higher cumulative throughputs [54].

We will demonstrate this situation with a very simple example. Consider the scenario with two nodes. Node 1 has data rate r_1 whereas node 2 has data rate $r_2 = k r_1$, where $k > 1$. Assuming each node transmits the same amount of payload, denoted p , each transmission, the air-time required by node 1 is p/r_1 , and the air-time required by node 2 is $p/(k r_1)$. Employing standard DCF, the two nodes will transmit equal number of frames in the long run. Assume each node transmits n frames. In total, the time required for the transmission of n frames per each station is $np(1/r_1 + 1/r_2)$. So, ignoring the idle times, the average throughput is

$$\frac{2np}{np\left(\frac{1}{r_1} + \frac{1}{k r_1}\right)} = \frac{2k}{k+1} r_1. \quad (1.1)$$

On the other hand, if air-time fairness is achieved, for every frame node 1 transmits, node 2 will transmit k frames on the average since the air-time required by node 1 is k times that of node 2. Hence, among the $2n$ frames transmitted in this scenario, only $2n/(k+1)$ of them will belong to node 1, and the rest will belong to node 2. Therefore, ignoring the idle times once again, the average throughput in this case is

$$\frac{2np}{\frac{2n}{k+1} \frac{p}{r_1} + \frac{2nk}{k+1} \frac{p}{k r_1}} = \frac{k+1}{2} r_1. \quad (1.2)$$

Comparing (1.1) to (1.2), it is obvious that however large k is, the throughput in the case of standard DCF is upper bounded by $2r_1$, demonstrating the fact that the fast nodes are dragged down by the slow ones, whereas the throughput of the air-time fair WLAN is linear in k .

A substantial amount of research has been dedicated to building air-time fair WLANs most of which focus on systems that require as minimal modification as possible to the existing widely deployed DCF. The first approach is based on the use of contention window parameter CW_{min} as an instrument to achieve air-time fairness. The references [56] and [57] analytically show for DCF that under certain assumptions, the nodal throughput is inversely proportional with the CW_{min} value of the node. In particular, air-time fairness can be achieved if the initial contention window size CW_{min} is chosen to be inversely proportional with the raw bit rate. Using CW_{min} adjustment for more general service differentiation purposes has also appeared in [58],[59],[60]. In [61], an algorithm for selecting optimal CW_{min} values is proposed. This reference also explores the usage of the Arbitration Interframe Space (AIFS) value defined in IEEE 802.11e for air-time usage control. The disadvantage of the method given in this study is that it requires recomputation whenever a station joins or leaves the network, or changes its rate. A neural network-based solution for finding CW_{min} and AIFS values to achieve air-time fairness is proposed in [62]. The main advantage of the CW_{min} -approach to deliver air-time fairness is in its simplicity of implementation and the preservation of the DCF mechanism. Several drawbacks of this approach within the scope of air-time based fairness are given below:

- The relationship between CW_{min} and the nodal throughput is valid only for regimes where the collision probabilities are small. Actually, the relationship between CW_{min} and the nodal throughput is sensitive to system parameters such as number of nodes, choice of initial congestion windows, etc. For example, a simulation study of [58] demonstrates that the throughput ratio between two classes of nodes with a fixed CW_{min} ratio is slightly sensitive to the number of nodes in each class. Similar results also appear in [61].
- Large initial contention windows are necessary when the ratio between the

lowest and highest raw bit rates is relatively large. This can lead to a considerable under-utilization of the channel [63].

- In actual implementations, CW_{min} needs to be a power of two [59]. Therefore, perfect air-time fairness between two nodes can not be achieved if the raw bit rate ratio is not a power of two.

In order to attack the long contention window sizes problem, in [63], the authors propose an on-line extension of the 802.11 DCF that dynamically adapts the minimum contention window of contending stations to achieve air-time fairness. However, each node is assumed to be aware of the number of competing nodes in the network which is difficult to manage in a distributed way. In [64], the authors propose a modification to the original CSMA/CA algorithm in which the contention windows of contending nodes are adjusted based on an estimator of the number of idle slots and the authors demonstrate high cumulative throughput as well as improved time-based fairness relative to the DCF. Despite the merits demonstrated in [64], deviation from the widely accepted CSMA/CA appears to be a drawback.

Packet fragmentation is another approach to achieve air-time fairness. The reference [65] proposes a solution where packets from higher layers are fragmented based on the raw bit rate. In this solution, nodes with high bit rates use a frame size equal to the MTU (maximum transmission unit) whereas slow nodes fragment their packets so as to transmit smaller frames at each transmission opportunity. A similar cross-layer scheme is proposed in [66] that uses IP path MTU discovery so as to reduce the number of bytes per frame sent by lower bit rate nodes while allowing higher bit rate nodes to send full size frames. An immediate drawback of the fragmentation-based approach is an increase in overhead due to fragmentation especially when most nodes are slow. Implementation complexity is another drawback due to need for cross-layer interaction.

Another category of solutions is the frame aggregation approach which is proposed in the IEEE 802.11e standard in which a transmission opportunity (TXOP), also referred to as the maximum channel occupation time, is broadcasted by the base station to each contending node. Consequently, nodes can aggregate their awaiting frames for transmission as long as the channel occupancy time does not

exceed TXOP. Frame aggregation is also a crucial component of 802.11n due to the benefits it offers due to the significant reduction of overhead [48]. Frame aggregation can be used as a means of achieving air-time fairness and nodes with better channel conditions are allowed to send multiple frames at a transmission opportunity as opposed to low bit rate nodes that do not perform aggregation. The reference [67] proposes a dynamic and distributed aggregation mechanism which addresses the performance anomaly in both UDP and TCP scenarios by achieving time-based fairness in nearly all of the tested configurations. There are also existing results on optimal aggregation policies in 802.11n that can substantially increase aggregate throughput [68]. The reference [69] formulates DCF with respect to mixed data rates and packet sizes, and offers an adaptive packet size adjustment method. The reference [70] demonstrates the advantages of TXOP operations over the legacy 802.11 DCF and compare different TXOP managing policies in order to obtain the optimal one. Although TXOP can be used as an effective means of providing air-time fairness, the following drawbacks are identified:

- Frames are typically of variable size and further mechanisms including fragmentation are needed to transmit a number of frames within TXOP.
- Frame aggregation is generally used as a means of reducing overhead and thus enhancing cumulative throughput. If this method is used for air-time fairness, then slow nodes would not benefit from aggregation as much in case they dominate the user type.
- Let us assume all frames to be of the same length for the sake of simplicity. In the TXOP approach to deliver air-time fairness, the TXOP may be defined to be the time required for the slowest node to transmit a single frame. Let us now assume a 802.11b WLAN occupied by two nodes with 11 Mbps raw bit rates. In this case, when a node has channel access, it will transmit 11 back-to-back frames. Clearly, such a scheme presents unfairness between these two nodes in the short term. The situation worsens when the ratio between the lowest and highest raw bit rates is even larger.
- Frame aggregation may lead to relatively poor delay performance as shown in [62].

A discrete-time Markov model for the performance analysis of the Enhanced Distributed Channel Access (EDCA) function of the IEEE 802.11e standard with EDCA parameters including CW_{min} , TXOP, and AIFS is presented in [71]. The reference [72] describes another algorithm called TCP Friendly Rate Control (TFRC) in a network in which all the nodes use TFRC as a transport layer protocol and air-time fairness can be achieved by adjusting the sending rate at the transport layer. The reference [73] solves the performance anomaly problem with a combination of contention window scaling approaches and TCP rate control approach. According to TCP rate control, each window adjusts its contention window and air-time fairness for the system is exhibited and aggregate throughput of the system is shown to improve. Another cross-layer approach is presented in [74] where CW_{min} adaptation in the MAC layer is coupled with video bit rate adaptation in the application layer. The drawback of these solutions in general lies in their cross-layer design and in the cases of TFRC [72] and TCP rate control [73], the fact that they cannot support transport protocols other than TCP.

Further literature on air-time fairness include Keceli et al. [75] who study the unfairness problem between the up-link and the down-link flows in the IEEE 802.11e infrastructure Basic Service Set (BSS) when the default settings are employed, the work by Lim et al. [76] that considers the unfairness problem among up-link and down-link flows in error-prone environments, and Wang et al. [77] who investigate fairness in terms of throughput and packet delays among users with diverse channel conditions due to the mobility and fading effects in WLANs.

In this part of the thesis, we will propose a novel approach for achieving air-time fairness in IEEE 802.11 WLANs which is relatively simple to implement. In our proposed approach, multiple instances of the standard back-off algorithm are run at each node. Equivalently, a competing node behaves as a collection of multiple virtual nodes where each virtual node has its own DCF instant. When the back-off timer of a virtual node hits zero, then its controlling physical node decides to transmit the awaiting frame on behalf of the virtual node. Having multiple instances of the back-off algorithm at a given node increases the channel access probability when compared with ordinary nodes with a single DCF. The method we propose employs the multiplicity of back-off algorithm instances as an instrument to deliver

air-time fairness.

To explain further, consider a competing node i that runs N_i instances of the basic back-off algorithm. Let us assume that each node i requires an average air-time $E[A_i]$ at each of its transmission opportunities. Let $A_{max} \geq E[A_j], \forall j$ be a value known to all nodes. We propose that the parameter N_i is set to $N_i = A_{max}/E[A_i]$, which can be done in a distributed manner since all nodes have the value of A_{max} . Of course, for ideal performance, A_{max} should be set to $\max_i E[A_i]$. However, this would require the dissemination of the $E[A_i]$ values of all nodes within the WLAN and impose communication overhead. Moreover, in cases which $E[A_i]$ values change such as rate adaptation, the new $E[A_i]$ values should also be announced. So, instead of using $A_{max} = \max_i E[A_i]$, we opted for the ratio of the maximum supported frame size to the minimum supported data rate in the protocol employed. In this manner, A_{max} becomes known to all nodes through protocol parameters and the condition $A_{max} \geq E[A_j], \forall j$ is satisfied.

Note that the parameter N_i need not be an integer. For such cases, we will describe a novel distributed mechanism that appropriately switches between $N_{i-} = \lfloor N_i \rfloor$ and $N_{i+} = \lceil N_i \rceil$ back-off algorithms. A node with non-integer N_i runs N_{i-} and N_{i+} back-off algorithms for appropriate durations so that on the average, the node obtains transmission opportunities proportional to N_i . Moreover, we will provide a novel Markov chain-based analytical model for the validation of this switching mechanism. In addition, we show that our method achieves air-time fairness at the expense of an acceptable reduction in channel utilization through an extensive simulation study. The proposed method can also be used in conjunction with frame aggregation to substantially mitigate this utilization reduction.

This part of the thesis has a different theme compared to the rest. Rather than providing methodology (as in the solution of CFMFQs) or demonstrating applications to the methodology (such as the workload-bounded queue or the analysis of horizon-based reservation), we aim to solve a more practical engineering problem in this part. However, it is still coupled to the general idea of the thesis in the aspect that we provide a stochastic model in the form of a Markov chain-based analysis for the switching mechanism to validate its effectiveness.

1.5 Contribution Summary and Organization

The contributions of this thesis can be summarized as follows:

A. Markov Fluid Queues

- A general framework for the solution of CFMFQs is given. Rather than approaching the task at hand as a boundary value problem, we chose to approximate the CFMFQ by a MRMFQ and apply existing methods.
- We provide an efficient algorithm based on block-tridiagonal LU decomposition for the solution of the linear system of equations stemming from the boundary conditions. The algorithm has linear complexity with respect to the number of regimes. Thus, a large number of regimes can be used for the MRMFQ approximation to achieve satisfactory levels of accuracy. Since the solution method for the MRMFQ we employ is based on the numerically stable additive decomposition method, the resulting solution framework for the CFMFQs is both stable and efficient. Also note that the algorithm based on block-tridiagonal LU decomposition can also be employed to solve ordinary MRMFQs, which has not been addressed before.
- We give a treatment for MRMFQs which have temporarily-absorbing states, that is there are some states that the background process cannot leave unless the buffer level reaches a certain value. These types of systems are encountered in our analysis for the horizon-based reservation mechanism.

B. The Workload-bounded MAP/PH/1 Queue

- We give a framework for modeling workload-bounded queues in which job arrival process and the service speed are functions of the unfinished workload. The arrival process is a MAP and the job size has a PH-type distribution, which is the most general case possible in Markov fluid queue framework. Two different rejection policies, namely partial and complete rejection, are formulated.
- Mathematical model for the workload-bounded MAP/PH/1 queue with complete rejection policy is given. This derivation includes formulating

the infinitesimal generator of the background process in terms of the arrival process and job size distribution parameters. Due to the complete rejection policy, even if none of the parameters depend on the buffer level, the model still is a CFMFQ.

- We also give the solution to the workload-bounded MAP/PH/1 queue for multiple customer classes.

C. The Horizon-based Delayed Reservation Mechanism in OBS Networks

- The horizon-based delayed reservation mechanism with deterministic offset times has been solved before in the context of MRMFQs. We formulate and solve the horizon-based delayed reservation mechanism with generally distributed offset times. This is achieved by employing the hazard rate function in modeling the offset time. The resulting model is CFMFQ, which the methods proposed can be applied readily.
- We also formulate and solve the horizon-based delayed reservation mechanism with generally distributed offset times for multiple traffic classes. QoS differentiation in terms of blocking probability due to different offset time distributions for each class is investigated.
- We show that deterministic offset times provide better performance than stochastic offset times in terms of blocking in single class case, and both overall blocking and QoS differentiation in multiple class case.
- The horizon-based delayed reservation mechanism in the presence of FDLs is formulated and solved. The offset times can be deterministic, or have discrete distributions. As deterministic offset times were shown to be better than stochastic offset times, this choice seems logical.
- Service differentiation by means of FDL access limitation is investigated. In this method, high priority class has access to the full set of FDLs, whereas the low priority class can access only a subset of the FDLs. We show that this method is a good candidate for service differentiation.

D. Air-time Fairness in Multi-rate IEEE802.11 WLANs

- We propose a fully distributed novel algorithm that provides air-time fairness in multi-rate IEEE802.11 WLANs. The algorithm is based on each node

running multiple instances of the standard DCF of IEEE802.11. The number of algorithms a node runs is inversely proportional to the air-time required by that node for the transmission of a frame.

- The number of algorithms a node needs to run might turn out to be non-integer. For such cases, we propose a method for maintaining air-time fairness. The method involves switching between the two integer neighbors of the non-integer number of algorithms. We also give a stochastic model for this method, and we present a novel and elaborate proof through a Markov chain-based analysis that the method indeed maintains air-time fairness.
- The method we propose sacrifices a tolerable amount of air-time utilization in favor of air-time fairness and overall throughput. We also provide a method that employs frame aggregation to combat the reduction in air-time utilization.
- We present an extensive simulation study that shows the effectiveness of the proposed method even in scenarios in which nodes become online and go offline, and have non-deterministic frame size distributions.

In chapter II, Markov fluid queues are summarized and the framework for solving CFMFQs including the block-tridiagonal LU decomposition algorithm is described. The workload-bounded MAP/PH/1 queue is formulated and solved in chapter III. The stochastic model for horizon-based reservation mechanism with or without FDLs is given in chapter IV. Chapter V is on air-time fairness in multi-rate IEEE802.11 WLANs, and our method for achieving air-time fairness is described here. Chapter VI concludes and presents future research directions.

Chapter 2

Markov Fluid Queues

2.1 Introduction

Markov Fluid Queues (MFQ) are systems in which the drift into or out of a buffer is determined by a Markov process. This Markov process is called the modulating or the background process, and usually is a continuous-time Markov chain (CTMC). For every state of the CTMC, there is a drift value, possibly different from the rest of the drifts. The buffer may have finite or infinite capacity. MFQs are described by two parameters: the infinitesimal generator of the background process, and the set of drift values for each state. MFQs can be categorized into three classes with respect to the dependence of their parameters on the buffer level.

1. First, we have the single-regime MFQs (SRMFQ). The parameters of the SRMFQs are fixed and they are independent of the buffer level.
2. The second category is the multi-regime MFQs (MRMFQ). The parameters of an MRMFQ depend on the buffer level in a piece-wise constant manner. Therefore, the buffer can be partitioned into *regimes* that determine the parameters. The parameters are constant within each regime, however they differ from the ones associated with any other regime. As the parameters of the MRMFQ are determined by the buffer level, this can be regarded as a sort

of feedback. Hence, MRMFQs are sometimes called multi-regime feedback MFQs.

3. In the last category, there is the continuous feedback MFQs (CFMFQ). The parameters of a CFMFQ are functions of the buffer level. These functions are not piece-wise constant functions, so defining regimes as in MRMFQs is not possible.

Now, we will revisit the examples from chapter I this time by also indicating the infinitesimal generators of their background processes and the drift matrices.

Example I: SRMFQ The buffer capacity B is finite and equal to 2. The background process is a 3-state CTMC with the infinitesimal generator

$$Q = \begin{bmatrix} -0.5 & 0.25 & 0.25 \\ 0.1 & -0.5 & 0.4 \\ 0.1 & 0.4 & -0.5 \end{bmatrix}.$$

The drifts in states 1, 2 and 3 are 0, -1 and 1 respectively, which we will denote in diagonal matrix form as

$$R = \begin{bmatrix} 0 & & \\ & -1 & \\ & & 1 \end{bmatrix}.$$

A sample evolution of the buffer level for this SRMFQ is given in Figure 2.1. The system starts with an empty buffer and the background process in state 3. During the time the background process stays in state 3, the buffer is filled with rate 1. When the background process switches to state 1, the buffer level remains constant as the drift in state 1 is 0. Then, the background process switches to state 3 and after a while, the buffer level hits the upper boundary. As it cannot be filled any more, the buffer level stays at B until the background process switches to state 2. Afterwards, the buffer is depleted with rate -1 and it is completely drained before the next state transition. So, the buffer level stays at 0 as it can be depleted no further.

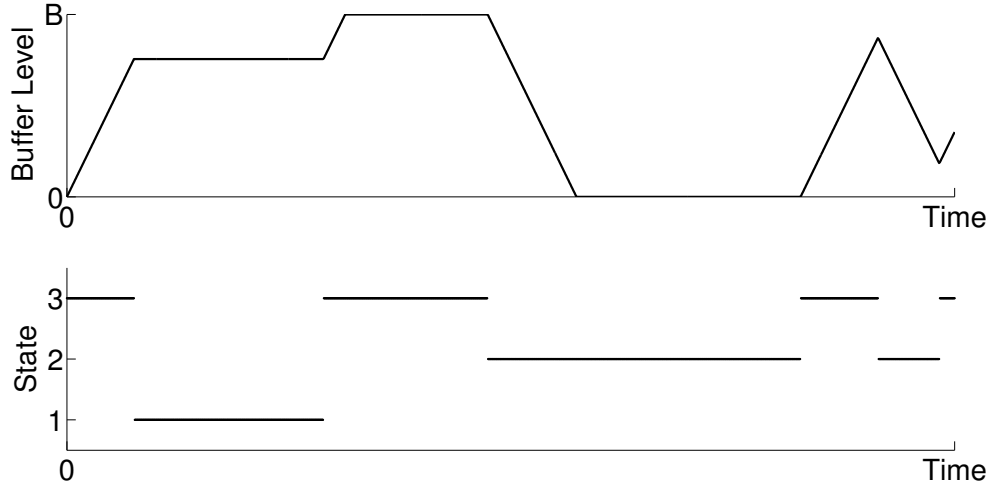


Figure 2.1: A sample path for the buffer level of the SRMFQ that has a background process with infinitesimal generator $Q = \begin{bmatrix} -0.5 & 0.25 & 0.25 \\ 0.1 & -0.5 & 0.4 \\ 0.1 & 0.4 & -0.5 \end{bmatrix}$. The drifts in states 1, 2 and 3 are 0, -1 and 1 respectively.

Example II: MRMFQ The buffer with capacity is again 2. The buffer is partitioned into two regimes; first regime being the region $(0, 1)$ and the second regime being $(1, 2)$. The background process is a 2-state CTMC with the infinitesimal generators

$$Q^{(1)} = \begin{bmatrix} -0.2 & 0.2 \\ 0.2 & -0.2 \end{bmatrix} \quad \text{and} \quad Q^{(2)} = \begin{bmatrix} -0.8 & 0.8 \\ 0.8 & -0.8 \end{bmatrix}$$

in regimes 1 and 2 respectively. Also, the drift matrices in regimes 1 and 2 are

$$R^{(1)} = \begin{bmatrix} -0.25 & \\ & 0.25 \end{bmatrix} \quad \text{and} \quad R^{(2)} = \begin{bmatrix} -1 & \\ & 1 \end{bmatrix}$$

respectively. Moreover, to completely describe the MRMFQ, we also need to define the behavior at the regime boundaries, which are at 0, 1 and 2 in this specific scenario. Let's assume that the MRMFQ behaves the same as in regime 1 in boundaries at 0 and 1, and as in regime 2 in boundary at 2.

A sample evolution of the buffer level for this MRMFQ is given in Figure 2.2. The system starts with an empty buffer and the background process in state 1. Therefore, the buffer level remains 0 until a state transition occurs as the drift in state 1 at boundary 0 is negative. When the background process switches to state 2, the buffer level starts increasing with rate 0.25. When the buffer level surpasses the

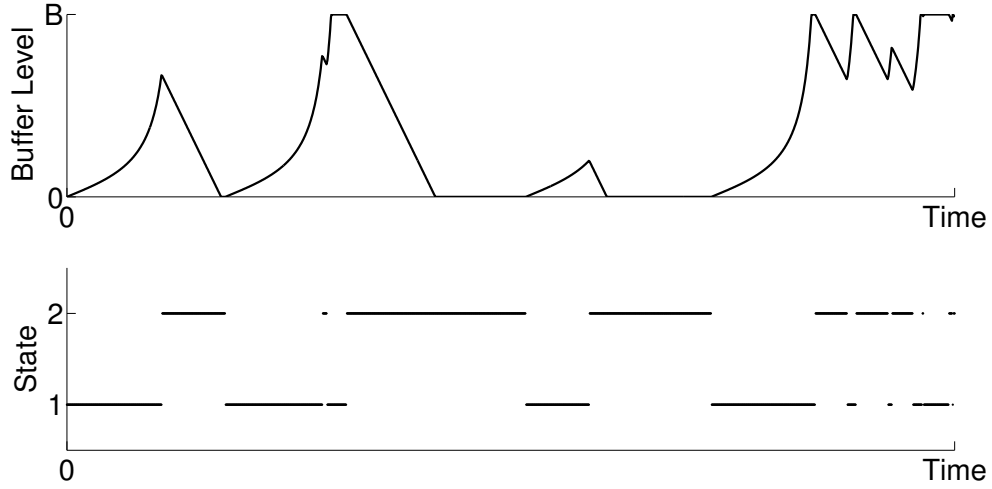


Figure 2.3: A sample path for the buffer level of the CFMFQ that has a background process with infinitesimal generator $Q(x) = (0.2 + x^2) \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}$, and the drifts are $R(x) = \begin{bmatrix} 0.2+2x^2 & \\ & -1 \end{bmatrix}$.

on the buffer level can be deduced from the figure as state transitions are more frequent when the buffer level is closer to the upper boundary.

In the remaining of this chapter, we will give the formal definitions of SRMFQs, MRMFQs and CFMFQs, and describe how to solve them.

2.2 Single-Regime Markov Fluid Queues

An SRMFQ is a joint process $\{X(t), Z(t)\}$, where $Z(t)$ is the background CTMC, and $X(t)$ is the buffer level. Let N denote the number of states of $Z(t)$. $Z(t)$ modulates the SRMFQ in the following manner. With every state $i, 1 \leq i \leq N$ of $Z(t)$, there is an associated drift, denoted by r_i . Then, the drift matrix of the SRMFQ, R , can be defined as the diagonal matrix with the drift rates as its diagonal entries:

$$R = \begin{bmatrix} r_1 & & & \\ & r_2 & & \\ & & \ddots & \\ & & & r_N \end{bmatrix}.$$

When $Z(t) = i$, $X(t)$ increases (decreases) with rate r_i ($-r_i$) if $r_i > 0$ ($r_i < 0$). Of course, when $X(t)$ is 0, it can be depleted no further. Therefore, for infinite-capacity SRMFQs, we have

$$\frac{d}{dt}X(t) = \begin{cases} r_i, & \text{when } Z(t) = i \text{ and } X(t) > 0, \\ \max\{r_i, 0\}, & \text{when } Z(t) = i \text{ and } X(t) = 0. \end{cases}$$

Similarly, if the buffer is of finite capacity, which we denote by B , we have

$$\frac{d}{dt}X(t) = \begin{cases} \min\{r_i, 0\}, & \text{when } Z(t) = i \text{ and } X(t) = B, \\ r_i, & \text{when } Z(t) = i \text{ and } B > X(t) > 0, \\ \max\{r_i, 0\}, & \text{when } Z(t) = i \text{ and } X(t) = 0. \end{cases}$$

The joint cdf vector at time t is defined as

$$F(x, t) = \begin{bmatrix} F_1(x, t) & \cdots & F_N(x, t) \end{bmatrix},$$

where

$$F_i(x, t) = \Pr\{X(t) \leq x, Z(t) = i\}, \quad 1 \leq i \leq N, \quad t \geq 0,$$

with $x \in [0, \infty)$ for the infinite buffer and $x \in [0, B]$ for the finite buffer. Assuming that $Z(t)$ is irreducible, the steady-state cdf vector $F(x) = \lim_{t \rightarrow \infty} F(x, t)$ always exists for the finite buffer case, and it exists under a stability condition for the infinite buffer case. It is well known [2] that the steady-state joint cdf vector satisfies

$$\frac{d}{dx}F(x)R = F(x)Q, \tag{2.1}$$

where Q is the infinitesimal generator of $Z(t)$.

It is also possible to write (2.1) using pdf form. Defining the steady-state joint pdf vector as $f(x) = \begin{bmatrix} f_1(x) & \cdots & f_N(x) \end{bmatrix}$ where $f_i(x) = dF_i(x)/dx$, $1 \leq i \leq N$, we have

$$\frac{d}{dx}f(x)R = f(x)Q. \tag{2.2}$$

If the solution is to be given in pdf form, the probability mass accumulations at 0, that is

$$c^{(0)} = \begin{bmatrix} c_1^{(0)} & \cdots & c_N^{(0)} \end{bmatrix}$$

where

$$c_i^{(0)} = \lim_{t \rightarrow \infty} \Pr \{X(t) = 0, Z(t) = i\}, \quad 1 \leq i \leq N$$

should also be specified. Note that $c^{(0)} = F(0)$. In the finite buffer case, the probability mass accumulations at the upper boundary B ,

$$c^{(B)} = \begin{bmatrix} c_1^{(B)} & \dots & c_N^{(B)} \end{bmatrix}$$

where

$$c_i^{(B)} = \lim_{t \rightarrow \infty} \Pr \{X(t) = B, Z(t) = i\}, \quad 1 \leq i \leq N$$

should also be specified.

In the remainder of this section, we will summarize the solution of SRMFQs.

2.2.1 Boundary Conditions

Infinite Buffer Case For the steady-state cdf to exist in the case of infinite buffer, the system should be stable. This means that the mean drift, $\bar{r} = \pi R \mathbf{1}$ should be negative. Here, π is the stationary distribution of $Z(t)$, and satisfies $\pi Q = \mathbf{0}$ and $\pi \mathbf{1} = 0$, where $\mathbf{1}$ and $\mathbf{0}$ represent vectors of ones and zeros of appropriate sizes respectively.

Let's also define three subsets of the set of all states, $S = \{1, \dots, N\}$,

$$S_- = \{i \mid i \in S, r_i < 0\},$$

$$S_0 = \{i \mid i \in S, r_i = 0\},$$

$$S_+ = \{i \mid i \in S, r_i > 0\}.$$

These sets include the states for which the drift is positive, zero and negative respectively. Obviously, the union of these three sets equal to S .

Once the stability condition is met, there are basically two sets of boundary conditions:

- (i) If the background process is in a state for which the drift is positive, the buffer level $X(t)$ cannot stay at zero. Therefore, $F_i(0) = 0, \forall i \in S_+$.

- (ii) In the solution of the either of the matrix differential equations (2.1) or (2.2), eigenvalues with positive real parts should be suppressed in order to have a stable cdf.

Finite Buffer Case Since the buffer is limited on both ends, we do not need a stability condition in this case. Similar to the infinite buffer case, there are two sets of boundary conditions.

- (i) If the background process is in a state for which the drift is positive, the buffer level $X(t)$ cannot stay at zero. Therefore, $F_i(0) = 0, \forall i \in S_+$.
- (ii) For the states that have negative drifts, the value of the steady-state cdf at the upper boundary point B should be equal to the corresponding entry of π , i.e. $F_i(B) = \pi_i, \forall i \in S_-$.

2.2.2 Spectral Solution

This method is due to the work of Kulkarni [2]. In this method, the following assumptions are made:

- (i) There are no states with zero drift, i.e. $S_0 = \emptyset$. This ensures that the matrix R is invertible.
- (ii) All eigenvalues of the matrix QR^{-1} are distinct.

Let $(\lambda_i, \phi_i), 1 \leq i \leq N$, denote the left eigenvalue-eigenvector pairs of the matrix QR^{-1} . First, rewrite the equation (2.1) as

$$\frac{d}{dx}F(x) = F(x)QR^{-1}. \quad (2.3)$$

Obviously, the general solution to (2.3) is given by

$$F(x) = \sum_{i=1}^N a_i e^{\lambda_i x} \phi_i = a e^{\Lambda x} \Phi, \quad (2.4)$$

where $a_i, 1 \leq i \leq N$, are scalar coefficients to be determined from the boundary conditions, and

$$a = \begin{bmatrix} a_1 & \cdots & a_N \end{bmatrix}, \quad \Lambda = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_N \end{bmatrix}, \quad \Phi = \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_N \end{bmatrix}.$$

The general solution (2.4) is valid for both infinite and finite buffers. However, the solution of a is different for each case.

Infinite Buffer Case Observe that $(0, \pi)$ is a left eigenvalue-eigenvector pair to QR^{-1} . Let $(\lambda_z, \phi_z) = (0, \pi)$. Then, a can be solved from

$$\begin{aligned} a_i &= 0 & \forall i \in S \text{ such that } \Re(\lambda_i) > 0, \\ \sum_{i=1}^N a_i \phi_{ij} &= 0 & \forall j \in S_+, \\ a_z &= 1. \end{aligned}$$

Notice that as x goes to infinity, all that remains nonzero in (2.4) is $a_z \pi$, which ensures $F(\infty) = \pi$ as expected.

Finite Buffer Case In this case, a can be solved from

$$\begin{aligned} \sum_{i=1}^N a_i \phi_{ij} &= 0 & \forall j \in S_+, \\ \sum_{i=1}^N a_i e^{\lambda_i B} \phi_{ij} &= \pi_j & \forall j \in S_-. \end{aligned}$$

The spectral solution has three main drawbacks:

- (i) It fails in the existence of repeated eigenvalues.

- (ii) Computation of eigenvalues of a matrix is problematic in general. This is especially amplified if there are eigenvalues close to each other. Detailed treatments using perturbation analysis can be found in various classical texts such as the books by Wilkinson [78], and Golub and Van Loan [15].
- (iii) The linear equation to solve for a is ill-conditioned, especially in the finite buffer case with large buffer capacities. This is due to the coexistence of growing and dying exponentials.

Akar and Sohraby [4] point out these drawbacks and propose a technique based on ordered Schur form that is numerically stable. Before we present this technique, we will go over the ordered Schur form.

2.2.3 Ordered Schur Form

Theorem 2.1 is stated in the book by Golub and Van Loan [15]. We refer the reader to this work for the proof, which will be left out here.

Theorem 2.1. *For every real square matrix A , there exists an orthogonal matrix U such that*

$$U^T A U = T$$

where T is upper block-diagonal, and each diagonal block of T is either 1×1 and an eigenvalue of A , or a 2×2 matrix with a complex conjugate pair of eigenvalues belonging to A .

Furthermore, the decomposition is not unique and U can be chosen so that the eigenvalues appear along the diagonal of T in any desired order.

For our purposes that will become clear in the next section, we would like to order the eigenvalues so that the zero(s) comes first, followed by the eigenvalues with negative and positive real parts respectively. So, we start with a Q matrix to

have

$$U^T A U = T = \begin{bmatrix} A_0 & A_{0-} & A_{0+} \\ & A_- & A_{-+} \\ & & A_+ \end{bmatrix}, \quad (2.5)$$

where A_0 , A_- and A_+ are upper triangular, and have the eigenvalues that are 0, and that have negative and positive real parts respectively on their diagonals. Furthermore, we would like to find a similarity transform on A that will make the off-diagonal blocks in (2.5), namely A_{0-} , A_{0+} and A_{-+} , all zeros.

A procedure for this purpose was described in Kankaya's work [79, pp. 28-31]. Letting

$$Y = U \begin{bmatrix} I & X_1 \\ 0 & I \end{bmatrix} \begin{bmatrix} I & 0 & 0 \\ 0 & I & X_2 \\ 0 & 0 & I \end{bmatrix}$$

produces

$$Y^{-1} A Y = \begin{bmatrix} A_0 & & \\ & A_- & \\ & & A_+ \end{bmatrix}. \quad (2.6)$$

Here, 0 and I represent zero and identity matrices of appropriate sizes respectively. It is easy to see that the matrices X_1 and X_2 can be solved from the Sylvester equations:

$$A_0 X_1 - X_1 \begin{bmatrix} A_- & A_{-+} \\ & A_+ \end{bmatrix} + \begin{bmatrix} A_{0-} & A_{0+} \end{bmatrix} = 0$$

$$A_- X_2 - X_2 A_+ + A_{-+} = 0.$$

These Sylvester equations are always solvable due to A_0 , A_- and A_+ having separate spectra [80].

2.2.4 Additive Decomposition Method

The additive decomposition method was proposed by Akar and Sohraby [4]. In this section and the rest of the thesis, we will prefer the pdf formulation.

First of all, assume that there are states with zero drifts, i.e. $S_0 \neq \emptyset$. In that case, without loss of generality, we can assume that the states are enumerated in such a way that all states with zero drifts come after the rest. This is always achievable through a similarity transform. Then, $f(x)$, Q and R can be partitioned as

$$f(x) = \begin{bmatrix} f_n(x) & f_z(x) \end{bmatrix}, \quad Q = \begin{bmatrix} Q_{nn} & Q_{nz} \\ Q_{zn} & Q_{zz} \end{bmatrix}, \quad R = \begin{bmatrix} R_n & \\ & 0 \end{bmatrix}$$

to write

$$\frac{d}{dx} \begin{bmatrix} f_n(x) & f_z(x) \end{bmatrix} \begin{bmatrix} R_n \\ 0 \end{bmatrix} = \begin{bmatrix} f_n(x) & f_z(x) \end{bmatrix} \begin{bmatrix} Q_{nn} & Q_{nz} \\ Q_{zn} & Q_{zz} \end{bmatrix}.$$

Simple algebra yields

$$f_z(x) = -f_n(x)Q_{nz}Q_{zz}^{-1} \quad (2.7)$$

$$\frac{d}{dx} f_n(x)R_n = f_n(x)(Q_{nn} - Q_{nz}Q_{zz}^{-1}Q_{zn}). \quad (2.8)$$

The equation pair (2.7) and (2.8) provide a treatment for SRMFQs that have states with zero drifts. Therefore, for the remainder of this chapter, we will assume that R is invertible.

Defining $A = QR^{-1}$, we can rewrite (2.2) as

$$\frac{d}{dx} f(x) = f(x)A. \quad (2.9)$$

Following the procedures given in section 2.2.3, we find a matrix Y satisfying (2.6).

Defining $\tilde{f}(x) = f(x)Y$, (2.9) can be rewritten as

$$\frac{d}{dx} \tilde{f}(x) = \tilde{f}(x) \begin{bmatrix} A_0 & & \\ & A_- & \\ & & A_+ \end{bmatrix}. \quad (2.10)$$

Assuming that the background process is irreducible and the mean drift $\bar{r}$ is nonzero, the matrix QR^{-1} has exactly one eigenvalue at 0 [81], meaning that A_0 is a scalar and equal to 0. Therefore (2.10) becomes

$$\frac{d}{dx} \tilde{f}(x) = \tilde{f}(x) \begin{bmatrix} 0 & & \\ & A_- & \\ & & A_+ \end{bmatrix}.$$

The solution of this equation is obviously $\bar{f}(x) = a_0 + a_- e^{A_- x} + a_+ e^{A_+ x}$ for some vector $a = \begin{bmatrix} a_0 & a_- & a_+ \end{bmatrix}$, which is to be solved from the boundary conditions.

Let's for the time being assume that the buffer is of finite capacity. Then, the growing exponential term $a_+ e^{A_+ x}$ can be written as

$$\begin{aligned} a_+ e^{A_+ x} &= a_+ e^{A_+ B} e^{-A_+ B} e^{A_+ x} \\ &= a_+ e^{A_+ B} e^{-A_+ (B-x)}, \end{aligned}$$

where the multiplier $e^{A_+ B}$ can be absorbed into a_+ . Then, the solution to (2.10) can be expressed as

$$\bar{f}(x) = a_0 + a_- e^{A_- x} + a_+ e^{-A_+ (B-x)}.$$

The only step at this point is to compute $f(x)$ from $\bar{f}(x) Y^{-1}$. Let

$$Y^{-1} = \begin{bmatrix} L_0 \\ L_- \\ L_+ \end{bmatrix},$$

where the number of rows of the blocks L_0 , L_- and L_+ are equal to 1, and the size of A_- and A_+ respectively. So, we can write the solution to (2.2) as

$$f(x) = \begin{bmatrix} a_0 & a_- & a_+ \end{bmatrix} \begin{bmatrix} 1 & & \\ & e^{A_- x} & \\ & & e^{-A_+ (B-x)} \end{bmatrix} \begin{bmatrix} L_0 \\ L_- \\ L_+ \end{bmatrix}. \quad (2.11)$$

For the infinite buffer case, only the dying exponential term can be present in the solution. Therefore, $a_0 = 0$ and $a_+ = \mathbf{0}$, and the solution to (2.2) is $f(x) = a_- e^{A_- x} L_-$.

In additive decomposition method, the computation of eigenvalues is eliminated altogether and the ordered Schur decomposition, which is known to be stable [82, pp. 17-18], is preferred. Moreover, only dying exponentials exist in the solution which leads to a well-conditioned equation in the unknown coefficient vector a .

2.3 Multi-Regime Markov Fluid Queues

Consider the Markov fluid queue with the buffer space partitioned in K regimes with the boundaries $0 = T^{(0)} < T^{(1)} < \dots < T^{(K-1)} < T^{(K)} = B$. When the buffer level x is in the range $(T^{(k-1)}, T^{(k)})$, the buffer is said to be in regime k . Within regime k , the infinitesimal generator of the background process is $Q^{(k)}$, and the diagonal drift matrix is $R^{(k)}$. In general, the pair $Q^{(j)}$ and $R^{(j)}$ is possibly different than $Q^{(k)}$ and $R^{(k)}$ whenever $j \neq k$, $1 \leq j, k \leq K$. In this manner, the behavior of the fluid queue at any point in time is determined by the regime at that time. Such systems are called multi-regime Markov fluid queues (MRMFQ).

In each regime k , $1 \leq k \leq K$, the infinitesimal generator of the background process and the drift matrix are denoted by $Q^{(k)}$ and $R^{(k)}$, respectively, and the drift in state i is denoted by $r^{(k)}$. At each regime boundary, $T^{(k)}$, $1 \leq k \leq K$, the infinitesimal generator of the background process and the drift matrix are denoted by $\tilde{Q}^{(k)}$ and $\tilde{R}^{(k)}$, respectively, and the drift in state i is denoted by $\tilde{r}^{(k)}$.

Similar to SRMFQs, we can define the joint cdf vector at time t in regime k as

$$\begin{aligned} F^{(k)}(x, t) &= \begin{bmatrix} F_1^{(k)}(x, t) & \cdots & F_N^{(k)}(x, t) \end{bmatrix}, & T^{(k-1)} < x < T^{(k)} \\ F_i^{(k)}(x, t) &= \Pr\{X(t) \leq x, Z(t) = i\}, & t \geq 0, \quad 1 \leq i \leq N, \end{aligned}$$

and the joint pdf vector at time t in regime k as

$$\begin{aligned} f^{(k)}(x, t) &= \begin{bmatrix} f_1^{(k)}(x, t) & \cdots & f_N^{(k)}(x, t) \end{bmatrix}, & T^{(k-1)} < x < T^{(k)} \\ f_i^{(k)}(x, t) &= \frac{d}{dx} F_i^{(k)}(x, t), & 1 \leq i \leq N. \end{aligned}$$

Then the steady-state joint cdf and pdf vectors in regime k are written as

$$\begin{aligned} F^{(k)}(x) &= \begin{bmatrix} F_1^{(k)}(x) & \cdots & F_N^{(k)}(x) \end{bmatrix}, & T^{(k-1)} < x < T^{(k)}, \\ F_i^{(k)}(x) &= \lim_{t \rightarrow \infty} F_i^{(k)}(x, t), & 1 \leq i \leq N, \\ f^{(k)}(x) &= \begin{bmatrix} f_1^{(k)}(x) & \cdots & f_N^{(k)}(x) \end{bmatrix}, & T^{(k-1)} < x < T^{(k)}, \\ f_i^{(k)}(x) &= \lim_{t \rightarrow \infty} f_i^{(k)}(x, t), & 1 \leq i \leq N. \end{aligned}$$

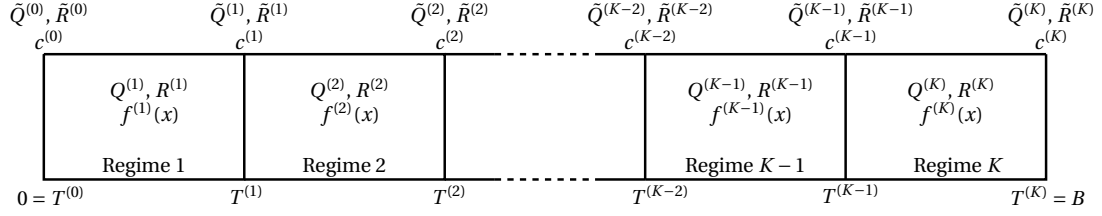


Figure 2.4: The structure of an MRMFQ

Lastly, the transient and steady-state probability mass accumulations at the regime boundaries can be written as

$$\begin{aligned}
 c^{(k)}(t) &= \begin{bmatrix} c_1^{(k)}(t) & \cdots & c_N^{(k)}(t) \end{bmatrix}, & 0 \leq k \leq K, \\
 c_i^{(k)}(t) &= \Pr \left\{ X(t) = T^{(k)}, Z(t) = i \right\}, & t \geq 0, \quad 1 \leq i \leq N, \\
 c^{(k)} &= \begin{bmatrix} c_1^{(k)} & \cdots & c_N^{(k)} \end{bmatrix}, & 0 \leq k \leq K, \\
 c_i^{(k)} &= \lim_{t \rightarrow \infty} c_i^{(k)}(t), & 1 \leq i \leq N.
 \end{aligned}$$

The parameters that describe an MRMFQ is summarized in Figure 2.4.

The sets of states with positive, zero and negative drifts in each regime k , $1 \leq k \leq K$, are defined as

$$\begin{aligned}
 S_-^{(k)} &= \left\{ i \mid i \in S, r_i^{(k)} < 0 \right\}, \\
 S_0^{(k)} &= \left\{ i \mid i \in S, r_i^{(k)} = 0 \right\}, \\
 S_+^{(k)} &= \left\{ i \mid i \in S, r_i^{(k)} > 0 \right\}.
 \end{aligned}$$

In addition, we need to define the counterparts of these sets for the regime boundaries:

$$\begin{aligned}
 \tilde{S}_-^{(k)} &= \left\{ i \mid i \in S, \tilde{r}_i^{(k)} < 0 \right\}, \\
 \tilde{S}_0^{(k)} &= \left\{ i \mid i \in S, \tilde{r}_i^{(k)} = 0 \right\}, \\
 \tilde{S}_+^{(k)} &= \left\{ i \mid i \in S, \tilde{r}_i^{(k)} > 0 \right\}.
 \end{aligned}$$

The states of the background process can be classified at the regime boundaries according to the signs of the drifts below and above the boundary.

1. A state is called *emitting* at regime boundary $T^{(k)}$ if $r^{(k)}$ and $r^{(k+1)}$ have the same sign. In such states, if $r^{(k)} > 0$, the buffer level increases towards $T^{(k)}$ from below in regime k and after it enters regime $k + 1$, continues to increase. Similarly, if $r^{(k+1)} < 0$, the buffer level decreases towards $T^{(k)}$ from above in regime $k + 1$ and after it enters regime k , continues to decrease.
2. A state is called *absorbing* at regime boundary $T^{(k)}$ if $r^{(k)} > 0$ and $r^{(k+1)} < 0$. In such states, the buffer level increases towards $T^{(k)}$ in regime k and decreases towards $T^{(k)}$ in regime $k + 1$.
3. A state is called *repulsive* at regime boundary $T^{(k)}$ if $r^{(k)} < 0$ and $r^{(k+1)} > 0$. In such states, the buffer level decreases away from $T^{(k)}$ in regime k and increases away from $T^{(k)}$ in regime $k + 1$.

A set of assumptions was laid out in the work of Kankaya and Akar [10] which we adopt here.

- (i) $S_+^{(k)} \neq \emptyset$, $S_-^{(k)} \neq \emptyset$, $1 \leq k \leq K$,
 $\tilde{S}_+^{(k)} \neq \emptyset$, $\tilde{S}_-^{(k)} \neq \emptyset$, $0 \leq k \leq K$.

This avoids the buffer level being stuck within a portion of the buffer or at a regime boundary.

- (ii) If $r_i^{(k)} > 0 > r_i^{(k+1)}$, then $\tilde{r}_i^{(k)} = 0$, $1 \leq k \leq K - 1$.

This means that state i is absorbing at regime boundary $T^{(k)}$. The assumption avoids infinitesimal oscillations around $T^{(k)}$.

- (iii) If state i is repulsive at regime boundary $T^{(k)}$, then one of the following is true:
 $\tilde{r}_i^{(k)} = 0$, or $\tilde{r}_i^{(k)} = r_i^{(k)}$, or $\tilde{r}_i^{(k)} = r_i^{(k+1)}$.

This means that the drifts at a repulsive boundary can be either zero, or left continuous or right continuous.

- (iv) If $r_i^{(k)} = 0$, then $\tilde{r}_i^{(k-1)} = \tilde{r}_i^{(k+1)} = 0$.

Like assumption (ii), this avoids infinitesimal oscillations around $T^{(k-1)}$ and $T^{(k)}$, and eliminates the possibility of multiple probability mass accumulations within arbitrarily small neighborhoods of the regime boundaries.

(v) The drifts of the remaining states are either left or right continuous.

It is natural to assume left or right continuity of the drift value at regime boundaries for emitting states. In this way, the possibility of multiple probability mass accumulations within arbitrarily small neighborhoods of the regime boundaries are also eliminated.

(vi) In all regimes, the mean drift is nonzero.

This ensures the additive decomposition method works by avoiding double eigenvalues at the origin. On the other hand, we would like to point out that this issue for the additive decomposition method has been resolved by the work of Telek and Vecsei [83]. However, as we do not encounter this problem throughout the problems attacked in this thesis, we will not elaborate and refer the reader to [83].

The set of differential equations that the joint cdf vector satisfies was derived by Mandjes et al. [9]. Kankaya and Akar [10] derive the equations in pdf form along with the boundary conditions. This choice of representation leads to a more concise representation of the boundary conditions. We also adopt the pdf representation for this purpose.

The steady-state joint pdf satisfies the differential equation

$$\frac{d}{dx} f^{(k)}(x) R^{(k)} = f^{(k)}(x) Q^{(k)}, \quad T^{(k-1)} < x < T^{(k)}, \quad 1 \leq k \leq K, \quad (2.12)$$

along with the following boundary conditions:

$$c_i^{(0)} = 0, \quad \forall i \in S_+^{(1)} \quad (2.13)$$

$$c_i^{(k)} = 0, \quad \forall i \in \left(S_+^{(k)} \cap S_+^{(k+1)} \right) \cup \left(S_-^{(k)} \cap S_-^{(k+1)} \right), \quad 1 \leq k < K \quad (2.14)$$

$$c_i^{(k)} = 0, \quad \forall i \in \left(S_-^{(k)} \cap S_+^{(k+1)} \right) \cap \left(\tilde{S}_+^{(k)} \cup \tilde{S}_-^{(k)} \right), \quad 1 \leq k < K \quad (2.15)$$

$$c_i^{(K)} = 0, \quad \forall i \in S_-^{(K)} \quad (2.16)$$

$$f^{(1)}(0+) R^{(1)} = c^{(0)} \tilde{Q}^{(0)} \quad (2.17)$$

$$f^{(k+1)}(T^{(k)}+) R^{(k+1)} - f^{(k)}(T^{(k)}-) R^{(k)} = c^{(k)} \tilde{Q}^{(k)}, \quad 1 \leq k < K \quad (2.18)$$

$$f_i^{(k)}(T^{(k)}-) = 0, \quad \forall i \in S_-^{(k)} \cap \left(\tilde{S}_0^{(k)} \cup \tilde{S}_+^{(k)} \right), \quad 1 \leq k < K \quad (2.19)$$

$$f_i^{(k+1)}(T^{(k)}+) = 0, \quad \forall i \in \left(\tilde{S}_0^{(k)} \cup \tilde{S}_-^{(k)} \right) \cap S_+^{(k+1)}, \quad 1 \leq k < K \quad (2.20)$$

$$f^{(K)}(B-)R^{(K)} = -c^{(K)}\tilde{Q}^{(K)} \quad (2.21)$$

$$\left(\sum_{k=1}^K \int_{T^{(k-1)}+}^{T^{(k)}-} f^{(k)}(x) dx + \sum_{k=0}^K c^{(k)} \right) \mathbf{1} = 1. \quad (2.22)$$

2.3.1 Solution of Multi-Regime Markov Fluid Queues

In this section, we will follow the additive decomposition method described in section 2.2.4. We still hold the assumption of nonzero drifts valid so that $R^{(k)}$, $1 \leq k \leq K$, matrices are invertible. Note that the treatment given in section 2.2.4 for states with zero drifts can easily be generalized to the multi-regime case.

Similar to the SRMFQ case, we start with defining $A^{(k)} = Q^{(k)} (R^{(k)})^{-1}$, $1 \leq k \leq K$, and expressing (2.12) as

$$\frac{d}{dx} f^{(k)}(x) = f^{(k)}(x) A^{(k)}.$$

Then, as described in section 2.2.3, we find a non-singular matrix $Y^{(k)}$ for each $A^{(k)}$ such that

$$\left(Y^{(k)} \right)^{-1} A^{(k)} Y^{(k)} = \begin{bmatrix} 0 & & \\ & A_-^{(k)} & \\ & & A_+^{(k)} \end{bmatrix}.$$

Here, $A_-^{(k)}$ and $A_+^{(k)}$ are upper-triangular matrices that have the eigenvalues of $A^{(k)}$ on their diagonals with negative and positive real parts respectively. Partitioning $\left(Y^{(k)} \right)^{-1}$ as

$$\left(Y^{(k)} \right)^{-1} = \begin{bmatrix} L_0^{(k)} \\ L_-^{(k)} \\ L_+^{(k)} \end{bmatrix},$$

where the number of rows of the blocks $L_0^{(k)}$, $L_-^{(k)}$ and $L_+^{(k)}$ are equal to 1, and the number of eigenvalues of $A^{(k)}$ with negative and positive real parts respectively, we can write the steady-state joint pdf in regime k , $1 \leq k \leq K$, as

$$f^{(k)}(x) = \begin{bmatrix} a_0^{(k)} & a_-^{(k)} & a_+^{(k)} \end{bmatrix} \begin{bmatrix} 1 \\ e^{A_-^{(k)}(x-T^{(k-1)})} \\ e^{-A_+^{(k)}(T^{(k)}-x)} \end{bmatrix} \begin{bmatrix} L_0^{(k)} \\ L_-^{(k)} \\ L_+^{(k)} \end{bmatrix}. \quad (2.23)$$

Assume for now that the MRMFQ has a finite buffer, i.e. $T^{(k)} = B < \infty$. Solving the MRMFQ involves computing $A_-^{(k)}$, $A_+^{(k)}$ and $(Y^{(k)})^{-1}$ for each regime k , $1 \leq k \leq K$, and then employing the boundary conditions (2.13)-(2.22) to solve for $a^{(k)} = \begin{bmatrix} a_0^{(k)} & a_-^{(k)} & a_+^{(k)} \end{bmatrix}$, $1 \leq k \leq K$, and $c^{(k)}$, $0 \leq k \leq K$.

Notice that setting the normalization condition (2.22) aside, each $a^{(k)}$ appears in equations involving only $a^{(k-1)}$, $a^{(k+1)}$, $c^{(k-1)}$ and $c^{(k)}$; and each $c^{(k)}$ appears in equations involving only $a^{(k)}$ and $a^{(k+1)}$ in the boundary conditions (2.13)-(2.21). Therefore, the system of linear equations can be made block-banded by ordering the unknown coefficient vectors as

$$\begin{bmatrix} c^{(0)} & a^{(1)} & c^{(1)} & a^{(2)} & c^{(2)} & \dots & a^{(K-1)} & c^{(K-1)} & a^{(K)} & c^{(K)} \end{bmatrix}.$$

The block-banded structure can be exploited to cut down the time required for the complete solution dramatically. A method for this purpose will be given in the next section. After this method, $a^{(k)}$ and $c^{(k)}$ are found upto a normalization constant. The complete solution is obtained by employing the normalization boundary condition (2.22). Note that the integration in (2.22) can be written explicitly using matrix algebra due to the form of $f^{(k)}(x)$, given in (2.23).

We conclude this section by describing the solution procedure for infinite buffers where $T^{(K)} = \infty$. Instead of the boundary conditions (2.16) and (2.21) that describe the behavior of the system at the upper boundary $T^{(K)} = B$ when $B < \infty$, we need to have conditions to ensure stability. Therefore, we should have $a_0^{(K)} = 0$, $a_+^{(K)} = \mathbf{0}$, so that the solution given in (2.23) remains bounded for all values of x . Moreover, there can not be any probability mass accumulation at infinity, which gives $c^{(K)} = \mathbf{0}$. The condition for the existence of a steady-state distribution for the infinite buffer case is that the mean drift in regime K should be strictly negative, i.e., $\pi^{(K)} R^{(K)} \mathbf{1} < 0$, where $\pi^{(k)}$ denotes the stationary distribution of the CTMC with the infinitesimal generator $Q^{(k)}$ [10].

2.3.2 Efficient Solution of Boundary Conditions

In this section, we will describe a method that makes use of the block-banded structure of the system of linear equations resulting from the boundary conditions. Defining

$$z = \begin{bmatrix} c^{(0)} & a^{(1)} & c^{(1)} & a^{(2)} & c^{(2)} & \dots & a^{(K-1)} & c^{(K-1)} & a^{(K)} & c^{(K)} \end{bmatrix},$$

the boundary conditions (2.13)-(2.21) (all but the normalization condition) yield a system of linear equations that can be expressed as

$$zH = \mathbf{0}, \quad (2.24)$$

where H is a square matrix that can be computed using the matrices $A_-^{(k)}$, $A_+^{(k)}$ and $(Y^{(k)})^{-1}$, $1 \leq k \leq K$.

To solve the equation (2.24), instead of using straightforward left null space computation methods which are usually cubic in computational complexity, we propose to use the block-tridiagonal LU factorization [15, pages 174-175].

In order to employ the block-tridiagonal LU factorization to solve the equation $zH = \mathbf{0}$, the matrix H must be invertible and the right hand side of the equation should be different from $\mathbf{0}$. For this purpose, we replace the very first column of H and the right hand side of the equation with the column vector $b = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}^T$. This means setting $c_1^{(0)} = 1$, and solving the rest of the unknowns accordingly. Notice that as the size of H is equal to the number of unknown constants and it does not involve the normalization condition, we can replace one of the equations in H arbitrarily with an equation that is linearly independent from the rest of the equations and that will not harm the block structure of H . This will not have any adverse effect on the final solution as $a^{(k)}$ and $c^{(k)}$ values will be normalized using (2.22).

Denoting the modified H matrix with $\tilde{H}$, the modified equation is $z\tilde{H} = b$. The

matrix $\tilde{H}$ can be partitioned as

$$\tilde{H} = \begin{bmatrix} H_1^M & H_1^U & & & \\ H_1^L & H_2^M & H_2^U & & \\ & H_2^L & H_3^M & \ddots & \\ & & \ddots & \ddots & H_{K-1}^U \\ & & & H_{K-1}^L & H_K^M \end{bmatrix}. \quad (2.25)$$

Then, the block-tridiagonal LU factorization of this matrix can be written as

$$\tilde{H} = \begin{bmatrix} I & & & \\ E_1 & I & & \\ & \ddots & \ddots & \\ & & E_{K-1} & I \end{bmatrix} \begin{bmatrix} F_1 & H_1^U & & \\ & F_2 & \ddots & \\ & & \ddots & H_{K-1}^U \\ & & & F_K \end{bmatrix},$$

where the matrix blocks E_i , $1 \leq i \leq K-1$ and F_i , $1 \leq i \leq K$ are to be found according to the following procedure:

$$\begin{aligned} F_1 &= H_1^M \\ y_1 &= b_1 F_1^{-1} \\ \mathbf{for } i &: 2 : K \\ E_{i-1} &= H_{i-1}^L F_{i-1}^{-1} \\ F_i &= H_i^M - E_{i-1} H_i^U \\ y_i &= (b_i - y_{i-1} H_i^U) F_i^{-1} \\ \mathbf{end} \end{aligned}$$

Here, $y = \begin{bmatrix} y_1 & \cdots & y_K \end{bmatrix}$ is a suitably partitioned vector that satisfies

$$z \begin{bmatrix} I & & & \\ E_1 & I & & \\ & \ddots & \ddots & \\ & & E_{K-1} & I \end{bmatrix} = y.$$

Then, the following backward substitution gives us the solution $z = \begin{bmatrix} z_1 & \cdots & z_K \end{bmatrix}$:

$$z_K = y_K$$

for $i = K - 1 : 1$

$$z_i = y_i - z_{i+1} E_i$$

end

Now, $a^{(k)}$, $1 \leq k \leq K$ and $c^{(k)}$, $0 \leq k \leq K$ can be computed from the normalization condition (2.22). Notice that this procedure has a computational complexity of $O(K)$.

Employing the procedure described for solving $a^{(k)}$ and $c^{(k)}$ values also requires the careful partitioning of the matrix $\tilde{H}$. Notice that not all $c^{(k)}$ values are nonzero due to the boundary conditions (2.13)-(2.16). The zero elements in $c^{(k)}$ vectors can be eliminated from H to reduce the computation time. However, this spoils structure of H in the sense that all “natural” blocks, that is the blocks corresponding to each $a^{(k)}$ and $c^{(k)}$ vector are no longer of equal size. Moreover, due to the boundary conditions (2.19)-(2.20), some $a^{(k)}$ vectors may have more than one natural blocks. All these factors should be taken into account when partitioning $\tilde{H}$.

In addition to all these, due to the elimination of some elements in $c^{(k)}$ vectors, the size of H may not be an integer multiple of K . Although Golub and Van Loan [15] assume that all the blocks in the partitioning (2.25) are square, this is not a necessary condition. Assume that the blocks H_i^L , $1 \leq i \leq K - 2$, H_i^M , $1 \leq i \leq K - 1$ and H_i^U , $1 \leq i \leq K - 2$ are $m \times m$; and the final block along the diagonal, H_K^M is $m' \times m'$, where $m' < m$. This means that H_{K-1}^L is $m' \times m$ and H_{K-1}^U is $m \times m'$. In this case, the procedure described above still holds. Therefore, in cases where $\tilde{H}$ cannot be partitioned into equal size blocks, the final block on the diagonal can be made smaller compared to the rest of the blocks on the diagonal, and the final off-diagonal blocks turn out to be rectangular matrices.

Obviously, the important point in all partitioning schemes is that all nonzero elements of $\tilde{H}$ should be included in the blocks.

2.3.3 Solution of MRMFQs with Temporarily-Absorbing States

In this section, we will give a treatment for a special case of MRMFQs in which $Q^{(k)}$ for some k , $1 \leq k \leq K$, has all-zero rows. Such MRMFQs will be encountered in chapter 4. The assumption of invertible $R^{(k)}$, $1 \leq k \leq K$, is still standing without loss of generality. Furthermore, we will assume the following. Let the background process have N states, and the states are enumerated so that $Q_{ij}^{(k)} = 0$ for $1 \leq i \leq N_0 < N$, $1 \leq j \leq N$. This means that when the background process enters a state $i \in \{1, \dots, N_0\}$ in regime k , it never leaves this state as long as the buffer level stays in regime k . When the buffer level exits regime k , it enters either $k-1$ or $k+1$ depending on the sign of $r_i^{(k)}$. Obviously, $r_i^{(k)} \neq 0$ as this would mean a permanent absorption. Upon hitting $T^{(k-1)}$ (or $T^{(k)}$), if $\tilde{r}_i^{(k-1)} = 0$ ($\tilde{r}_i^{(k)} = 0$) where $1 \leq i \leq N_0$, then we assume in general that $\tilde{Q}_{ii}^{(k-1)} \neq 0$ ($\tilde{Q}_{ii}^{(k)} \neq 0$) so that state i is not absorbing at boundary $T^{(k-1)}$ ($T^{(k)}$) and the background process can leave state i at this boundary. If $\tilde{r}_i^{(k-1)} \neq 0$ ($\tilde{r}_i^{(k)} \neq 0$), then the buffer level enters regime $k-1$ ($k+1$). In this case, we assume one of the following two conditions:

- (i) $Q_{ii}^{(k-1)} \neq 0$ ($Q_{ii}^{(k+1)} \neq 0$), and the background process can leave state i within regime $k-1$ ($k+1$).
- (ii) The state i is emitting at the boundary $T^{(k-1)}$ ($T^{(k)}$). If $Q_{ii}^{(k-1)} = 0$ ($Q_{ii}^{(k+1)} = 0$), then there exists a regime k' , $1 \leq k' < k-1$ ($k+1 < k' \leq K$), such that the background process can leave state i either at boundary $T^{(k'-1)}$ ($T^{(k')}$) or within regime k' .

These assumptions ensure that all states and buffer level values remain reachable.

Another assumption we will make is that within regime k , the background process can transit from states $j \in \{N_0 + 1, \dots, N\}$ into states $i \in \{1, \dots, N_0\}$. This ensures that states $\{1, \dots, N_0\}$ are accessible from the remaining states, and that the matrix $Q_{22}^{(k)}$ where

$$Q^{(k)} = \begin{bmatrix} 0 & 0 \\ Q_{21}^{(k)} & Q_{22}^{(k)} \end{bmatrix},$$

is not stochastic, and is invertible.

As $R^{(k)}$ is diagonal and invertible, the matrix

$$A^{(k)} = Q^{(k)} \left(R^{(k)} \right)^{-1} = \begin{bmatrix} 0 & 0 \\ A_{21}^{(k)} & A_{22}^{(k)} \end{bmatrix}$$

has the same number of all-zero rows as $Q^{(k)}$. Also, remember that there exists a non-singular $Y^{(k)}$ that satisfies

$$\left(Y^{(k)} \right)^{-1} A^{(k)} Y^{(k)} = \begin{bmatrix} A_0^{(k)} & & \\ & A_-^{(k)} & \\ & & A_+^{(k)} \end{bmatrix}.$$

Then, we can write

$$\begin{bmatrix} 0 & 0 \\ A_{21}^{(k)} & A_{22}^{(k)} \end{bmatrix} Y^{(k)} = Y^{(k)} \begin{bmatrix} A_0^{(k)} & & \\ & A_-^{(k)} & \\ & & A_+^{(k)} \end{bmatrix},$$

which clearly leads to the conclusion $A_0^{(k)} = 0$, as $Y^{(k)}$ is non-singular, whereas the left-hand-side is a matrix with as many all-zero rows as the size of $A_0^{(k)}$, and the matrices $A_-^{(k)}$ and $A_+^{(k)}$ are upper-triangular with nonzero diagonal entries.

Therefore, in MRMFQs that have background processes with N_0 absorbing states in regime k , the solution to the MRMFQ stays the same but a slight modification (compare to (2.23) on page 44):

$$f^{(k)}(x) = \begin{bmatrix} a_0^{(k)} & a_-^{(k)} & a_+^{(k)} \end{bmatrix} \begin{bmatrix} I & & \\ & e^{A_-^{(k)}(x-T^{(k)})} & \\ & & e^{-A_+^{(k)}(T^{(k+1)}-x)} \end{bmatrix} \begin{bmatrix} L_0^{(k)} \\ L_-^{(k)} \\ L_+^{(k)} \end{bmatrix}.$$

Here, $a_0^{(k)}$ is a row vector of length N_0 , and the number of rows of the block $L_0^{(k)}$ is N_0 .

2.4 Continuous Feedback Markov Fluid Queues

In CFMFQs, the transition rates within the states of the background process and/or the drifts depend on the buffer level in a continuous fashion. Therefore, instead of

the regime-dependent $Q^{(k)}$ and $R^{(k)}$ matrices of MRMFQs, we have $Q(x)$ and $R(x)$ matrices that depend continuously on the buffer level x . In this case, the CFMFQ is characterized with the matrix pair $(Q(x), R(x))$. Scheinhardt et al. [11] investigate CFMFQs with finite buffer capacity under the following assumptions:

- (i) $Q_{ij}(x)$, $\forall i, j \in S$, is continuous and finite on the domain $[0, B]$.
- (ii) $r_i(x)$, $\forall i \in S$, is continuously differentiable on the domain $(0, B)$, and finite on $[0, B]$.
- (iii) $r_i(x)$, $\forall i \in S$, are strictly bounded away from 0 on $(0, B)$, i.e. $\min_i \inf_{x \in (0, B)} |r_i(x)| > 0$.
- (iv) $S_- = \{i \mid i \in S, r_i(x) < 0\} \neq \emptyset$,
 $S_+ = \{i \mid i \in S, r_i(x) > 0\} \neq \emptyset$.
- (v) If $i \in S_-$, then $r_i(0) = 0$ and $r_i(B) = r_i(B-) < 0$.
 If $i \in S_+$, then $r_i(B) = 0$ and $r_i(0) = r_i(0+) > 0$.
- (vi) a. For all $j \in S_-$ ($j \in S_+$), there exists an $i \in S_+$ ($i \in S_-$) such that $Q_{ij}(B) > 0$ ($Q_{ij}(0) > 0$).
- b. Denoting the indicator function with $1_{\{\cdot\}}$, define the matrix $\tilde{Q}$ whose entries are given by $\tilde{Q}_{ij} = 1_{\{i \in S_-\}} Q_{ij}(0) + 1_{\{i \in S_+\}} Q_{ij}(B)$. The matrix $\tilde{Q}$ is irreducible.
- (vii) $F_i(x, t)$, $\forall i \in S$, is twice continuously differentiable on the domain $(0, B) \times [0, \infty)$.

The continuity assumptions (i)-(ii) can be relaxed in the following way. Regime boundaries are defined at points in $(0, B)$ that $Q_{ij}(x)$, $i, j \in S$, or $r_i(x)$, $i \in S$, has a discontinuity. This leads to a “multi-regime continuous feedback MFQ” in the sense that within each regime, there is a CFMFQ that satisfies the continuity assumptions (i)-(ii). The whole system can be described by describing CFMFQs in each regime and writing additional boundary conditions for the regime boundaries.

By assumptions (iii)-(iv), the time that the joint process moves from (x_1, i) to (x_2, i) without a state transition can be expressed as $\int_{x_1}^{x_2} dx / r_i(x)$ and is finite. Also

notice that the sets S_- and S_+ are fixed sets that do not depend on the buffer level due to assumption (iii).

Assumption (v) is a natural consequence of the boundedness of the system and the continuity assumptions.

Assumption (vi) is a sufficient condition for the irreducibility of the system. The assumption (vi)a ensures that any state (x_2, j) , $j \in S_-$, is accessible from any state (B, i) , $i \in S_+$, and any state (x_2, j) , $j \in S_+$, is accessible from any state $(0, i)$, $i \in S_-$. Then, the assumption (vi)b ensures that any state (B, j) , $j \in S_-$, is accessible from any state (B, i) , $i \in S_+$, and any state $(0, j)$, $j \in S_+$, is accessible from any state $(0, i)$, $i \in S_-$. Together, these two assumptions mean that any joint state (x_2, j) , $0 \leq x_2 \leq B$, $j \in S$, is accessible from (x_1, i) , $0 \leq x_1 \leq B$, $i \in S$. Scheinhardt et al. [11] emphasize that this is a sufficient irreducibility condition, and their derivations remain intact if the CFMFQ is irreducible regardless of whether it satisfies assumption (vi) or not.

Lastly, assumption (vii) yields the equality $\partial_x \partial_t F_i(x, t) = \partial_t \partial_x F_i(x, t)$, which is used in the derivations by Scheinhardt et al. [11].

Under this set of assumptions, and the following definitions for the distributions and the probability masses

$$\begin{aligned}
F_i(x, t) &= \Pr \{X(t) \leq x, Z(t) = i\} & 1 \leq i \leq N, \quad t \geq 0 \\
F(x, t) &= \begin{bmatrix} F_1(x, t) & \cdots & F_N(x, t) \end{bmatrix} \\
F(x) &= \lim_{t \rightarrow \infty} F(x, t) \\
f(x) &= \frac{d}{dx} F(x) \\
c_i^{(0)} &= \Pr \{X(t) = 0, Z(t) = i\} \\
c^{(0)} &= \begin{bmatrix} c_1^{(0)} & \cdots & c_N^{(0)} \end{bmatrix} \\
c_i^{(B)} &= \Pr \{X(t) = B, Z(t) = i\} \\
c^{(B)} &= \begin{bmatrix} c_1^{(B)} & \cdots & c_N^{(B)} \end{bmatrix},
\end{aligned}$$

the steady-state joint pdf vector of the buffer level satisfies the differential equation

$$\frac{d}{dx} (f(x)R(x)) = f(x)Q(x), \quad 0 < x < B, \quad (2.26)$$

along with the boundary conditions

$$\begin{aligned}
c_i^{(0)} &= 0 & \forall i \in S_+ \\
c_i^{(B)} &= 0 & \forall i \in S_- \\
f(0+)R(0+) &= c^{(0)}Q(0) \\
f(B-)R(B-) &= -c^{(B)}Q(B) \\
\left(c^{(0)} + \int_0^B f(x)dx + c^{(B)} \right) \mathbf{1} &= 1.
\end{aligned}$$

Scheinhardt et al. [11] give the explicit solution for a two-state system and recommend numerical methods for more complex systems. Boxma et al. [84] study a similar two-state system with infinite buffer. German et al. [13] attack the problem as a boundary value problem and employ three different numerical discretization methods to solve (2.26), whereas Gribaudo and Telek [14] extend this study to accommodate sign changes in the drift matrix $R(x)$ which leads to potential probability masses at points of sign change. In this thesis, we propose approximating CFMFQs with MRMFQs, as the solution framework for MRMFQs is readily available.

For this purpose, the buffer space is divided into a number of regimes in any of which the parameters of the background process are held constant. To specify, for a finite buffer of capacity B , the buffer space is divided into K uniform regimes, i.e.

$$T^{(k)} = \frac{k}{K} B, \quad 0 \leq k \leq K.$$

In each regime k , the infinitesimal generator of the background process and the drift matrix are chosen to be the matrices $Q(x)$ and $R(x)$ evaluated at the lower boundary point:

$$\begin{aligned}
\tilde{Q}^{(k)} &= Q\left(T^{(k)}\right), & 0 \leq k \leq K \\
Q^{(k)} &= \tilde{Q}^{(k-1)}, & 1 \leq k \leq K \\
\tilde{R}^{(k)} &= R\left(T^{(k)}\right), & 0 \leq k \leq K \\
R^{(k)} &= \tilde{R}^{(k-1)}, & 1 \leq k \leq K.
\end{aligned}$$

These expressions for $\tilde{Q}^{(k)}$, $Q^{(k)}$, $\tilde{R}^{(k)}$, $R^{(k)}$ and $T^{(k)}$ completely describe the MRMFQ approximation to the CFMFQ. Afterwards, the methods to solve MRMFQs described in section 2.3 can be utilized to obtain the solution to the CFMFQ.

In the case of infinite buffer capacity, as $B = T^{(K)} = \infty$, a method for picking a suitable value for $T^{(K-1)}$ is needed. This value should be such that we can safely assume $Q(x)$ and $R(x)$ matrices are approximately held constant beyond it. To this end, we propose selecting $T^{(K-1)}$ the minimum value satisfying

$$\max \left\{ \left| Q(T^{(K-1)}) - \lim_{x \rightarrow \infty} Q(x) \right|_{\infty}, \left| R(T^{(K-1)}) - \lim_{x \rightarrow \infty} R(x) \right|_{\infty} \right\} \leq \epsilon \quad (2.27)$$

for a given $\epsilon > 0$. After selecting a suitable $T^{(K-1)}$, we can uniformly choose the boundary points as

$$T^{(k)} = \frac{k}{K-1} T^{(K-1)}, \quad 0 \leq k \leq K-1.$$

Lastly, we present the condition for the existence of a steady-state distribution for the infinite buffer case. we consider the most general scenario in which $Q(x)$ and $R(x)$ both depend on the buffer level x . Assuming convergence, we define $\lim_{x \rightarrow \infty} Q(x) = \bar{Q}$ and $\lim_{x \rightarrow \infty} R(x) = \bar{R}$. Let $\bar{\pi}$ satisfy $\bar{\pi} \bar{Q} = \mathbf{0}$, $\bar{\pi} \mathbf{1} = 1$. Then, the distribution exists if $\bar{\pi} \bar{R} \mathbf{1} < 0$. For further exploration of this topic, see Brockwell et al. [85].

We close this chapter with the observation that as more and more regimes are used, the accuracy of the MRMFQ approximation increases. However, this does not cause any problems as we have already given a method based on block-tridagonal LU decomposition to solve the boundary conditions in linear time with respect to the number of regimes

Chapter 3

The Workload-Bounded MAP/PH/1 Queue

3.1 Introduction

The workload-bounded MAP/PH/1 queue can be described as follows:

- The job arrival process is modeled by a workload-dependent Markovian Arrival Process (MAP). The matrix parameters of a workload-dependent MAP are allowed to vary with the instantaneous buffer level.
- The workload brought by an individual job has a phase type (PH-type) distribution. The parameters of this distribution is fixed.
- The queue is drained at a rate $c(x)$ when the queue occupancy takes the value $x > 0$. This means that two jobs with exactly the same amount of workload may see different service times, depending on the buffer level at the instants they arrive.
- The queue service discipline is FIFO (first-in-first-out).
- The buffer capacity can be finite or infinite. In the finite buffer case, we define the buffer capacity in terms of the overall workload that the buffer can hold,

instead of the maximum number of jobs allowed in the system as usually preferred by most finite queue capacity models.

- We investigate two different policies as to what to reject when the workload limit gets to be exceeded.
 - *Partial Rejection Policy*: If the current workload plus the job size of an arriving job exceeds the workload capacity B , then the workload is increased up to B , and the remaining part of the arriving job is rejected.
 - *Complete Rejection Policy*: The job is completely rejected in the same situation.

We achieve the following with our model:

- We obtain the steady-state distribution of the system workload in the infinite and finite buffer capacity scenarios, and for both partial and complete rejection policies in the finite buffer capacity case.
- In the finite buffer capacity case with partial rejection policy, workload loss probability, defined as the ratio of the rejected amount of workload to the total amount of workload that arrived in steady-state, is obtained.
- In the finite buffer capacity case with complete rejection policy, both job and workload loss probabilities are obtained.
- All results are verified through simulations.

The following main steps make up the framework for this purpose:

- (i) Using sample path arguments, we model the workload-dependent MAP/PH/1 queue for the infinite buffer case, and for the finite buffer case under partial and complete rejection policies, as a CFMFQ. This involves deriving the transition rates for different classes of states in terms of the workload-bounded MAP parameters, the job size distribution and the buffer capacity.
- (ii) The resulting CFMFQ is approximated by an MRMFQ using discretization.

- (iii) The boundary conditions for this MRMFQ are solved using block-tridiagonal LU factorization in order to obtain the steady-state distribution of the buffer level.

The main contributions of are as follows:

- We extend the problem addressed in [20] and [22] to a more general setting with workload-dependent MAP arrivals as opposed to Poisson arrivals.
- We provide a numerical solution to the finite-capacity workload-dependent MAP/PH/1 queue with complete rejection, which has not been addressed before in the literature. Complete rejection policy is essential in computer and communication systems and networks in which jobs need to be processed in their entirety.
- We employ the block-tridiagonal LU factorization to solve the linear system of equations that arise in the solution of the MRMFQs. The entries of the block-banded matrix is obtained by using Schur decomposition and Sylvester equations as in [10]. We also show by numerical examples that the computational complexity of the proposed algorithm depends linearly on the number of regimes used for discretizing the CFMFQ.

3.2 Stochastic Model for the Workload-dependent MAP/PH/1 Queue

We start by defining the Markovian arrival process (MAP) and the phase-type (PH-type) distribution. The MAP is described by the count process $\{N(t)\}$ and the phase process $\{J(t)\}$, assuming values in $\{0, 1, \dots\}$ and $\{1, \dots, \ell\}$, respectively. The two-dimensional Markov process $\{N(t), J(t)\}$ is then modeled as a Markov process on the state-space $\{(i, j) : i \in \mathcal{Z}, 1 \leq j \leq \ell\}$ whose infinitesimal generator matrix can be

represented in block form as

$$\begin{bmatrix} D_0 & D_1 & & \cdots \\ & D_0 & D_1 & \cdots \\ & & D_0 & D_1 & \cdots \\ & & & \ddots \end{bmatrix}. \quad (3.1)$$

In the above generator, D_0 and D_1 are $\ell \times \ell$ matrices, D_0 has negative diagonal elements and non-negative off-diagonal elements, D_1 is non-negative, and $D = D_0 + D_1$ is an irreducible infinitesimal generator. Note that an upward jump in the count process $\{N(t)\}$ above corresponds to a new packet arrival. When the generator is of the form (3.1), the underlying process is called a MAP which is characterized with the matrix pair (D_0, D_1) .

The PH-type distribution is defined as the distribution of time till absorption in a finite state continuous-time Markov chain [86]. For this purpose, we define a Markov process on the states $\{1, 2, \dots, h, h+1\}$ with initial probability vector $\begin{bmatrix} \alpha_1 & \cdots & \alpha_h & 0 \end{bmatrix}$ and infinitesimal generator

$$Q = \begin{bmatrix} T & T^0 \\ \mathbf{0} & 0 \end{bmatrix},$$

where T is an $h \times h$ matrix, T^0 is $h \times 1$, and $T\mathbf{1} + T^0 = \mathbf{0}$. The matrix T is assumed to be nonsingular, which ensures that the absorbing state can be reached from any other state [87]. The last state $h+1$ is called the absorbing state, and the time till absorption into this absorbing state is a random variable which is said to have a PH-type distribution with representation (α, T) , where $\alpha = \begin{bmatrix} \alpha_1 & \cdots & \alpha_h \end{bmatrix}$.

We study three different versions of the workload-dependent MAP/PH/1 queue; the Infinite Buffer (IB) case, then the case of Finite Buffer with Partial Rejection (FB-PR), and finally the Finite Buffer with Complete Rejection (FB-CR) policies. In all the three versions, the following hold:

- The arrival process is a MAP characterized with the pair $(D_0(x), D_1(x))$ with ℓ phases. The matrices $D_0(x)$ and $D_1(x)$ are $\ell \times \ell$, $D_0(x)$ has negative diagonal elements and non-negative off-diagonal elements, $D_1(x)$ is non-negative,

and $D(x) = D_0(x) + D_1(x)$ is an irreducible infinitesimal generator. The matrix $D_0(x)$ ($D_1(x)$) governs transitions among phases of the MAP without (with) arrivals.

- The job size, represented by the random variable U , has a PH-type distribution characterized with the pair (α, T) with h phases. α , the initial probability vector, is a row vector of length h , and T is an $h \times h$ matrix with negative diagonal elements and non-negative off-diagonal elements. With the definition $T^0 = -T\mathbf{1}$, the matrix $\begin{bmatrix} T & T^0 \\ 0 & 0 \end{bmatrix}$ defines the CTMC governing the PH-type process where state $h+1$ is the absorbing state. The random variable U is defined as the time till absorption into the absorbing state $h+1$.
- The workload depletion rate $c(x)$ is a function of the instantaneous buffer level, x .

Next, we describe the specific stochastic models for the three cases: the infinite buffer, the finite buffer with partial rejection and the finite buffer with complete rejection.

3.2.1 Infinite Buffer (IB)

Consider the operation of the infinite buffer. When the buffer is depleted, it stays empty until the next job arrival. When an arrival occurs, the buffer level increases abruptly by an amount equal to the arriving job size. Replacing these jumps by durations of linear increase with unity slope, we can obtain a transformed process, denoted by $X(t)$, from the buffer level process of the MAP/PH/1 queue, denoted by $Y(t)$. The process $X(t)$ can be modeled as a Markov fluid queue. Since a sample path of $Y(t)$ can be obtained from the sample path of $X(t)$ by deleting the time segments in which $X(t)$ is increasing, solving for the steady-state distribution of $X(t)$ will suffice as far as the steady-state distribution of $Y(t)$ is concerned. Therefore, workload-dependent infinite buffers can be analyzed using the paradigm of Markov fluid queues.

Next, we define the CFMFQ associated with the workload-dependent MAP/PH/1

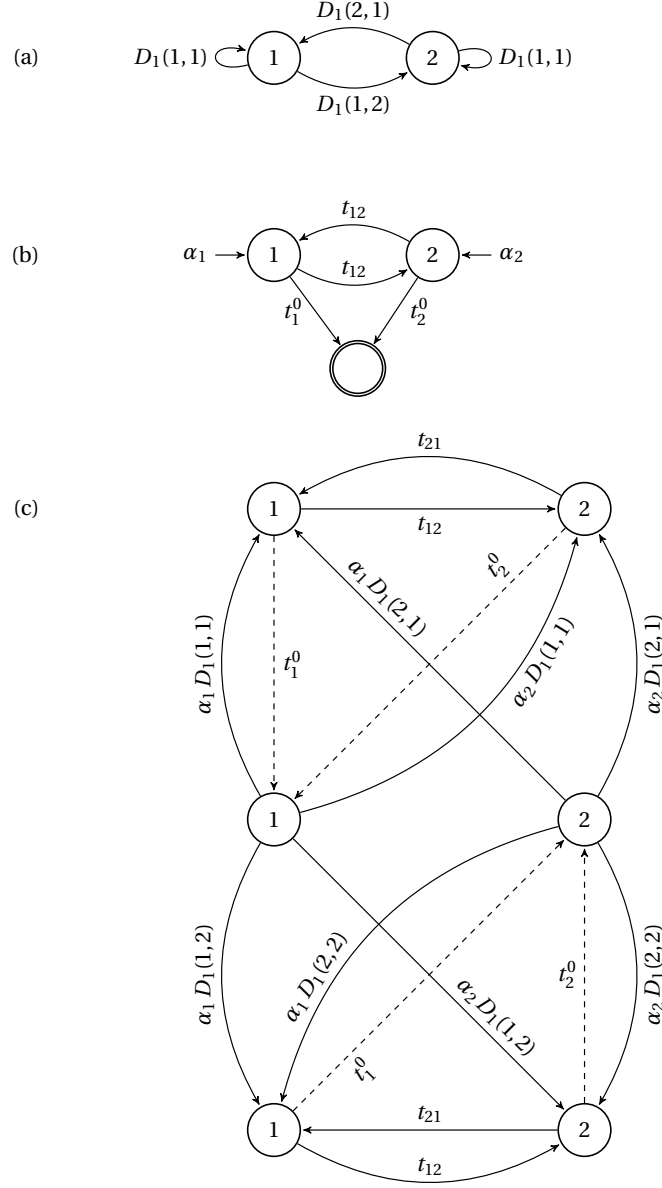


Figure 3.1: An example for the background process of the CFMFQ that models the workload-dependent MAP/PH/1 queue. (a) The MAP has 2 states. Transition rates between states without arrivals (D_0) are omitted. (b) The job size distribution is PH-type with two phases. The absorbing state is denoted with the doubly-encircled state. (c) The CTMC corresponding to this workload-dependent MAP/PH/1 queue. The pair of states at the top are the job size phases that return to MAP state 1 upon absorption, whereas the pair of states at the bottom are the job size phases that return to MAP state 2 upon absorption. The two states in the middle are the MAP states. Transition rates between states without arrivals (D_0) are omitted.

queue for which $Y(t)$ denotes the buffer level at time t . Let $X(t)$ be the process obtained from $Y(t)$ via the transformation described, and $Z(t)$ be its background process. The states of $Z(t)$ will be made up of the phases of the arrival process and those of the job size.

As an example, consider the system demonstrated in Figure 3.1. The MAP, shown in Figure 3.1(a), has 2 states, and the job size distribution, shown in Figure 3.1(b), is PH-type with two phases. When the MAP is in state 1, a job arrival without state change occurs with rate $D_1(1, 1)$, and a job arrival with a transition into state 2 occurs with rate $D_1(1, 2)$. Similarly, when the MAP is in state 2, a job arrival without state change occurs with rate $D_1(2, 2)$, and a job arrival with a transition into state 1 occurs with rate $D_1(2, 1)$. Upon an arrival, the CTMC will go into one of the phases for the job size distribution. Assume that the MAP is in state 1, and an arrival with a state transition into MAP state 2 occurs. As the initial probability vector for the job size distribution is $\alpha = [\alpha_1 \quad \alpha_2]$, the background process will transit into phase 1 with rate $\alpha_1 D_1(1, 2)$, and into phase 2 with rate $\alpha_2 D_1(1, 2)$. Then, the background process will wander within the two phases with the transition rates given with T . When absorption occurs, the background process must return to MAP state 2, as the arrival would cause a state transition in the MAP. For this purpose, we need to have as many replicas of the job size phases as the number of MAP states. Each set of replicas will return a specific MAP state, and upon an arrival, the background process should transit into the appropriate replica. In Figure 3.1(c), the pair of states at the top represents the replica that return to MAP state 1 upon absorption, whereas the pair of states at the bottom return to MAP state 2. The two states in the middle represent the MAP states.

When the background process is in one of the MAP states, the buffer is being depleted with rate $c(x)$. On the other hand, when the background process is in one of the phases belonging to the job size replicas, the buffer fills with a unit rate. Consequently, the process $X(t)$ can be described by a CFMFQ with $h\ell + \ell$ states with the infinitesimal generator $Q(x)$ and the rate matrix $R(x)$ given as follows:

$$Q(x) = \begin{bmatrix} I_\ell \otimes T & I_\ell \otimes T^0 \\ D_1(x) \otimes \alpha & D_0(x) \end{bmatrix}, \quad R(x) = \begin{bmatrix} I_{h\ell} & \\ & -c(x)I_\ell \end{bmatrix}, \quad 0 \leq x < \infty. \quad (3.2)$$

The first $h\ell$ states represent the PH-type replicas, in which the drift is +1, and the

states $h\ell + 1$ through $h\ell + \ell$ represent the MAP phases with drifts $-c(x)$. Note that the MAP states could also have been chosen to be the first ℓ states instead of the last, which would require minor modifications in the definitions of $Q(x)$ and $R(x)$.

Let $f(x) = [f_1(x) \ \cdots \ f_{h\ell+\ell}(x)]$ and $F(x) = [F_1(x) \ \cdots \ F_{h\ell+\ell}(x)]$ denote the steady-state joint pdf and cdf vectors for this CFMFQ, assuming that the steady-state distribution exists. In order to find the steady-state distribution of the buffer level, the MRMFQ approximation described in section 2.4 is carried out, along with the procedure to pick a suitable $T^{(K-1)}$ value.

Next, we examine the boundary conditions for the infinite buffer. Notice that, in each state of the background process, the sign of the service speed remains the same for all regimes, meaning that all states are emitting throughout the buffer space. This allows us to simplify the set of boundary conditions given in (2.13)-(2.22) to obtain:

$$c_m^{(0)} = 0, \forall m \in S_+^{(1)} \quad (3.3)$$

$$c_m^{(k)} = 0, \forall m \in \left(S_+^{(k)} \cap S_+^{(k+1)} \right) \cup \left(S_-^{(k)} \cap S_-^{(k+1)} \right), 1 \leq k < K \quad (3.4)$$

$$c_m^{(K)} = 0, \forall m \in S_-^{(K)} \quad (3.5)$$

$$f^{(1)}(0+)R^{(1)} = c^{(0)}\tilde{Q}^{(0)} \quad (3.6)$$

$$f^{(k+1)}(T^{(k)}+)R^{(k+1)} - f^{(k)}(T^{(k)}-)R^{(k)} = c^{(k)}\tilde{Q}^{(k)}, 1 \leq k < K \quad (3.7)$$

$$f^{(K)}(B-)R^{(K)} = -c^{(K)}\tilde{Q}^{(K)} \quad (3.8)$$

$$\left(\sum_{k=1}^K \int_{T^{(k-1)}+}^{T^{(k)}-} f^{(k)}(x)dx + \sum_{k=0}^K c^{(k)} \right) \mathbf{1} = 1. \quad (3.9)$$

We have to solve $c^{(k)}, 0 \leq k \leq K$, and $a^{(k)}, 1 \leq k \leq K$. We immediately obtain $c^{(k)} = \mathbf{0}$, $1 \leq k \leq K$, and $a_0^{(K)} = 0$, $a_+^{(K)} = \mathbf{0}$. Defining

$$c_-^{(0)} = \begin{bmatrix} c_{h\ell+1}^{(0)} & \cdots & c_{h\ell+\ell}^{(0)} \end{bmatrix} \quad (3.10)$$

and expressing the steady-state joint pdf vector in regime k as

$$f^{(k)}(x) = \begin{bmatrix} a_0^{(k)} & a_-^{(k)} & a_+^{(k)} \end{bmatrix} \begin{bmatrix} L_0^{(k)} \\ e^{A_-^{(k)}(x-T^{(k-1)})} L_-^{(k)} \\ e^{-A_+^{(k)}(T^{(k)}-x)} L_+^{(k)} \end{bmatrix} = a^{(k)} V^{(k)}(x), \quad (3.11)$$

we obtain the linear equation $zH = \mathbf{0}$ where

$$z = \begin{bmatrix} c_-^{(0)} & a^{(1)} & \dots & a^{(K-1)} & a_-^{(K)} \end{bmatrix}, \quad (3.12)$$

$$H = \begin{bmatrix} W^{(0)} & & & & \\ -V^{(1)}(0) & V^{(1)}(T^{(1)}) & & & \\ & -V^{(2)}(T^{(1)}) & V^{(2)}(T^{(2)}) & & \\ & & \ddots & \ddots & \\ & & & V^{(K-1)}(T^{(K-1)}) & \\ & & & & -W^{(K)} \end{bmatrix}, \quad (3.13)$$

$$W^{(0)} = \begin{bmatrix} 0_{\ell \times h\ell} & I_\ell \end{bmatrix} \tilde{Q}^{(0)} (R^{(1)})^{-1}, W^{(K)} = L_-^{(K)}, \quad (3.14)$$

and the matrix $V^{(k)}(x)$, $1 \leq k \leq K$, is defined in (3.11). The unknowns $\begin{bmatrix} c_-^{(0)} & a^{(1)} & \dots & a^{(K-1)} & a_-^{(K)} \end{bmatrix}$ can be solved through the block-tridiagonal LU decomposition algorithm described in section 2.3.2 and the normalization condition (3.9). In the block-tridiagonal LU decomposition algorithm, all the blocks can be chosen to be $(h\ell + \ell) \times (h\ell + \ell)$.

After we obtain the steady-state pdf vector for the associated MRMFQ, denoted by $f(x)$, we can find the steady-state joint pdf vector of the buffer level denoted by $g(x) = \begin{bmatrix} g_1(x) & \dots & g_\ell(x) \end{bmatrix}$ where

$$g_i(x) = \frac{d}{dx} \lim_{t \rightarrow \infty} P\{J(t) = i, Y(t) \leq x\}, \quad x > 0, \quad x \neq T^{(k)}, \quad 0 \leq k \leq K, \quad (3.15)$$

$J(t)$ being the phase process for the underlying MAP and $Y(t)$ being the buffer level process. By conditioning on the components of $f(\cdot)$ which correspond to the states in which the buffer is being depleted, we obtain

$$g(x) = \frac{\begin{bmatrix} f_{h\ell+1}(x) & \dots & f_{h\ell+\ell}(x) \end{bmatrix}}{F(\infty) \begin{bmatrix} 0_{1 \times h\ell} & 1_{1 \times \ell} \end{bmatrix}^T}. \quad (3.16)$$

Note that this step corresponds to deletion of the segments in the sample paths of $X(t)$ in which $X(t)$ is increasing.

Lastly, we present the condition for the existence of a steady-state distribution for IB. Assume convergence, and define $\lim_{x \rightarrow \infty} D_0(x) = \bar{D}_0$, $\lim_{x \rightarrow \infty} D_1(x) = \bar{D}_1$ and $\lim_{x \rightarrow \infty} c(x) = \bar{c}$. Let $\bar{\pi}$ satisfy $\bar{\pi}(\bar{D}_0 + \bar{D}_1) = \mathbf{0}$, $\bar{\pi}\mathbf{1} = 1$. Then, the distribution exists if $\beta\bar{\pi}\bar{D}_1\mathbf{1}/\bar{c} < 1$, where $\beta = -\alpha T^{-1}\mathbf{1}$ is the mean job size.

3.2.2 Finite Buffer with Partial Rejection (FB-PR)

Now, assume that the buffer capacity B is finite, and whenever the job size of an arrival causes the buffer to overflow, the portion of the arriving job that fits the available buffer space is accepted into the buffer, and the remaining part is lost. To solve this model, we employ the same transformation used for IB. This time, the linear increases will occasionally be accompanied by flat regions in which the buffer level stays constant at B . These regions correspond to the overflows caused by arrivals that do not fit into the available buffer space.

Using the same states and enumeration as in IB, we obtain the same $Q(x)$ and $R(x)$ matrices given in (3.2). Since the buffer capacity is finite, we discretize the CFMFQ into a MRMFQ using K regimes with boundaries $T^{(k)} = Bk/K$, $0 \leq k \leq K$. Then, the matrices characterizing the MRMFQ are given by

$$\tilde{Q}^{(K)} = Q(B), \quad Q^{(k)} = \tilde{Q}^{(k-1)} = Q(T^{(k-1)}), \quad 1 \leq k \leq K \quad (3.17)$$

$$R^{(k)} = R(T^{(k-1)}), \quad 1 \leq k \leq K \quad (3.18)$$

$$\tilde{R}^{(K)} = R(B)^-, \quad \tilde{R}^{(k)} = R(T^{(k)}), \quad 0 \leq k \leq K-1 \quad (3.19)$$

where $R(B)^-$ denotes the matrix which is equal to the $R(B)$ matrix except for the positive elements of $R(B)$, which are set to 0.

The boundary conditions that hold for FB-PR are (3.3)-(3.9) from which we obtain again the equation $zH = \mathbf{0}$ where this time

$$z = \begin{bmatrix} c_-^{(0)} & a^{(1)} & \dots & a^{(K)} & c_+^{(K)} \end{bmatrix} \quad (3.20)$$

$$H = \begin{bmatrix} W^{(0)} & & & & \\ -V^{(1)}(0) & V^{(1)}(T^{(1)}) & & & \\ & -V^{(2)}(T^{(1)}) & V^{(2)}(T^{(2)}) & & \\ & & \ddots & \ddots & \\ & & & V^{(K)}(T^{(K)}) & \\ & & & & -W^{(K)} \end{bmatrix}, \quad (3.21)$$

$$c_+^{(K)} = [c_1^{(K)} \dots c_{h\ell}^{(K)}],$$

$$W^{(K)} = - \begin{bmatrix} I_{h\ell} & 0_{h\ell \times \ell} \end{bmatrix} \tilde{Q}^{(K)} (R^{(K)})^{-1},$$

and $c_-^{(0)}$, $W^{(0)}$, and $V^{(k)}(x)$ are given by (3.10), (3.14), and (3.11), respectively. Again, the equation $zH = \mathbf{0}$ can be solved using block-tridiagonal LU factorization.

The steady-state joint pdf vector of the buffer level, $g(x) = [g_1(x) \ \cdots \ g_\ell(x)]$, is defined in the same way as in IB; and similar to (3.16), is given by

$$g(x) = \frac{\begin{bmatrix} f_{h\ell+1}(x) & \cdots & f_{h\ell+\ell}(x) \end{bmatrix}}{F(B) \begin{bmatrix} 0_{1 \times h\ell} & 1_{1 \times \ell} \end{bmatrix}^T}. \quad (3.22)$$

For FB-PR, one can define the workload loss probability, denoted by $P_{w,\text{FB-PR}}$, as the ratio of the workload lost to the overall amount of workload that has arrived at the buffer. In terms of $g(x)$, $P_{w,\text{FB-PR}}$ is expressed as:

$$P_{w,\text{FB-PR}} = \frac{1}{-\alpha T^{-1} \mathbf{1}} \int_0^B \int_{B-y}^\infty (w - B + y) g(y) D_1(y) \mathbf{1} \alpha e^{Tw} T^0 dw dy. \quad (3.23)$$

3.2.3 Finite Buffer with Complete Rejection (FB-CR)

In this model, the buffer capacity B is finite, and arrivals with job sizes exceeding the available buffer space are rejected entirely. We will use the same approach as in IB and FB-PR; however, due to the complete rejection policy, the underlying CFMFQ will be different. We will now show that the characterizing matrices of the associated CFMFQ for FB-CR are given as:

$$Q(x) = \begin{bmatrix} I_\ell \otimes \tilde{T}(x) & I_\ell \otimes \tilde{T}^0(x) \\ D_1(x) \otimes \tilde{\alpha}(x) & \tilde{D}_0(x) \end{bmatrix}, \quad 0 \leq x \leq B, \quad (3.24)$$

$$R(x) = \begin{bmatrix} I_{h\ell} \\ -c(x) I_\ell \end{bmatrix}, \quad 0 \leq x < B, \quad (3.25)$$

$$R(B) = \begin{bmatrix} 0_{h\ell \times h\ell} \\ -c(B) I_\ell \end{bmatrix},$$

where

$$\tilde{T}^0(x) = [\tilde{t}_i^0(x)], \quad \tilde{t}_i^0(x) = \frac{t_i^0}{1 - \nu^{(i)} e^{T(B-x)} \mathbf{1}}, \quad 1 \leq i \leq h, \quad (3.26)$$

$$\tilde{T}(x) = [\tilde{t}_{ij}(x)], \quad \tilde{t}_{ij}(x) = t_{ij} \frac{1 - \nu^{(j)} e^{T(B-x)} \mathbf{1}}{1 - \nu^{(i)} e^{T(B-x)} \mathbf{1}}, \quad 1 \leq i, j \leq h, i \neq j, \quad (3.27)$$

$$\tilde{t}_{ii}(x) = - \sum_{j=1, j \neq i}^h \tilde{t}_{ij}(x) - \tilde{t}_i^0(x), \quad 1 \leq i \leq h, \quad (3.28)$$

$$\tilde{\alpha}(x) = [\tilde{\alpha}_i(x)], \quad \tilde{\alpha}_i(x) = \alpha_i \left(1 - \nu^{(i)} e^{T(B-x)} \mathbf{1} \right), \quad 1 \leq i \leq h, \quad (3.29)$$

$$\tilde{D}_0(x) = D_0(x) + D_1(x) \alpha e^{T(B-x)} \mathbf{1}, \quad (3.30)$$

t_{ij} , $1 \leq i, j \leq h$, t_i^0 , $1 \leq i \leq h$, and α_i , $1 \leq i \leq h$, denote the entries of T , T^0 , and α , respectively, and $\nu^{(i)}$ denotes a row vector of zeros except for a value of one at position i .

To see this, let us first assume the associated CFMFQ is in a state at time τ during which the buffer level is increasing. Denoting the phase for the job size at time τ by $s(\tau) = i$, the buffer level with $X(\tau)$, we already know the remaining job size denoted by $u(\tau)$ satisfies $u(\tau) < B - X(\tau)$. Hence, as $\Delta\tau \rightarrow 0$, the quantity $\Pr\{s(\tau + \Delta\tau) = h + 1 \mid s(\tau) = i, u(\tau) < B - X(\tau)\}$

$$\begin{aligned} &= \frac{\Pr\{s(\tau + \Delta\tau) = h + 1, u(\tau) < B - X(\tau) \mid s(\tau) = i\}}{\Pr\{u(\tau) < B - X(\tau) \mid s(\tau) = i\}} \\ &= \frac{t_i^0 \Delta\tau}{1 - \nu^{(i)} e^{T(B-X(\tau))} \mathbf{1}}. \end{aligned} \quad (3.31)$$

Similarly, the probability $\Pr\{s(\tau + \Delta\tau) = j \mid s(\tau) = i, u(\tau) < B - X(\tau)\}$ where $1 \leq j \leq h$ reduces as $\Delta\tau \rightarrow 0$ to

$$\begin{aligned} &= \frac{\Pr\{u(\tau) < B - X(\tau), s(\tau + \Delta\tau) = j \mid s(\tau) = i\}}{\Pr\{u(\tau) < B - X(\tau) \mid s(\tau) = i\}} \\ &= \frac{\Pr\{u(\tau + \Delta\tau) < B - X(\tau + \Delta\tau) \mid s(\tau + \Delta\tau) = j\} t_{ij} \Delta\tau}{\Pr\{u(\tau) < B - X(\tau) \mid s(\tau) = i\}} \\ &= \frac{1 - \nu^{(j)} e^{T(B-X(\tau+\Delta\tau))} \mathbf{1}}{1 - \nu^{(i)} e^{T(B-X(\tau))} \mathbf{1}} t_{ij} \Delta\tau \\ &= \frac{1 - \nu^{(j)} \left[\sum_{n=0}^{\infty} (T(B - X(\tau) + c(\tau)\Delta\tau))^n / n! \right] \mathbf{1}}{1 - \nu^{(i)} e^{T(B-X(\tau))} \mathbf{1}} t_{ij} \Delta\tau \\ &= \frac{1 - \nu^{(j)} \left[\sum_{n=0}^{\infty} (T(B - X(\tau)))^n / n! + O(\Delta\tau) \right] \mathbf{1}}{1 - \nu^{(i)} e^{T(B-X(\tau))} \mathbf{1}} t_{ij} \Delta\tau \\ &= \frac{1 - \nu^{(j)} e^{T(B-X(\tau))} \mathbf{1}}{1 - \nu^{(i)} e^{T(B-X(\tau))} \mathbf{1}} t_{ij} \Delta\tau + O(\Delta\tau^2). \end{aligned} \quad (3.32)$$

The north-east and north-west blocks of the characterizing matrix $Q(x)$ in (3.24)

follow from (3.31) and (3.32), respectively. Let us now look into epochs of new job arrivals when the buffer level takes the value x . A new job is rejected if its size exceeds the available buffer space $B - x$, which occurs with probability $\alpha e^{T(B-x)} \mathbf{1}$. Therefore, transitions associated with rejected arrivals should contribute to transitions without arrivals. This observation is reflected in the south-east corner of $Q(x)$ given in (3.24). On the other hand, using Bayes' rule, an accepted job's service will start at phase i with probability $\tilde{\alpha}_i(x) = \alpha_i (1 - \nu^{(i)} e^{T(B-x)} \mathbf{1})$ which is the probability of being initially at phase i given that the PH-type random variable is less than $B - x$ from which the south-west corner of $Q(x)$ in (3.24) follows.

After the CFMFQ is defined, its MRMFQ approximation with discretization is carried out exactly as in FB-PR except for a slight modification. Notice that $\tilde{T}$ and $\tilde{T}^0$ approach infinity as x approaches B . Therefore, we can not use (3.17) towards $\tilde{Q}^{(K)}$. We need to use a $\tilde{Q}^{(K)}$ matrix with finite entries in order to be able to carry out numerical calculations. Moreover, arbitrarily large choices as the entries of $\tilde{Q}^{(K)}$ have the potential to lead to an ill-conditioned system of equations. Therefore, we opted for using $\tilde{Q}^{(K)} = Q(B - B/2K)$, which amounts to computing $\tilde{Q}^{(K)}$ by evaluating $Q(x)$ at the midpoint of regime K . We obtained reasonably well results with this midpoint strategy, so we stick to this method throughout the study.

At this point, we have everything needed for solving the buffer level distribution $g(x)$, which is defined in (3.15). The solution procedure is exactly the same as in FB-PR, and (3.22) still holds for $g(x)$. Now that we have the buffer level distribution, the job loss probability $P_{\text{FB-CR}}$ can be expressed as

$$P_{\text{FB-CR}} = \frac{\int_0^B g(y) D_1(y) \mathbf{1} \alpha e^{T(B-y)} \mathbf{1} dy}{\int_0^B g(y) D_1(y) \mathbf{1} dy}. \quad (3.33)$$

On the other hand, the workload loss probability $P_{w,\text{FB-CR}}$ can be written as

$$P_{w,\text{FB-CR}} = \frac{1}{-\alpha T^{-1} \mathbf{1}} \int_0^B \int_{B-y}^\infty w g(y) D_1(y) \mathbf{1} \alpha e^{Tw} T^0 dw dy. \quad (3.34)$$

Note that (3.34) differs from (3.23) only by the term w inside the integral instead of the term $w - (B - y)$, which reflects different rejection policies of these two schemes.

Lastly, we would like to point out that all three models can be extended to accommodate multiple traffic classes that have distinct arrival processes and job size distributions. Here, we give the stochastic model of FB-CR with two traffic classes. Let the arrivals processes of class A and B be MAP_A and MAP_B with parameters $(D_{0,A}(x), D_{1,A}(x))$ and $(D_{0,B}(x), D_{1,B}(x))$ respectively. We denote the number of states in MAP_A and MAP_B with ℓ_A and ℓ_B respectively. The job size distributions PH_A and PH_B have parameters (α_A, T_A) and (α_B, T_B) with number of phases h_A and h_B respectively. Then, the infinitesimal generator of the background process can be written as

$$Q(x) = \begin{bmatrix} I_{\ell_A \ell_B} \otimes \tilde{T}_A(x) & & I_{\ell_A \ell_B} \otimes \tilde{T}_A^0(x) \\ & I_{\ell_A \ell_B} \otimes \tilde{T}_B(x) & I_{\ell_A \ell_B} \otimes \tilde{T}_B^0(x) \\ I_{\ell_B} \otimes (D_{1,A}(x) \otimes \tilde{\alpha}_A) & D_{1,B}(x) \otimes (I_{\ell_A} \otimes \tilde{\alpha}_B) & D_0(x) \end{bmatrix}$$

where

$$D_0(x) = (I_{\ell_B} \otimes \tilde{D}_{0,A}(x)) + (\tilde{D}_{0,B}(x) \otimes I_{\ell_A}),$$

and (3.26)-(3.30) can be used to compute $\tilde{D}_{0,A}(x)$, $\tilde{D}_{0,B}(x)$, $\tilde{T}_A(x)$, $\tilde{T}_A^0(x)$, $\tilde{T}_B(x)$, $\tilde{T}_B^0(x)$, $\tilde{\alpha}_A(x)$, and $\tilde{\alpha}_B(x)$. The drift matrix is given by

$$R(x) = \begin{bmatrix} I_{\ell_A \ell_B (h_A + h_B)} & \\ & -c(x) I_{\ell_A \ell_B} \end{bmatrix}.$$

The solution method is exactly the same as the single class case, the only difference being that the composite MAP consists of $\ell_A \ell_B$ states, and the number of states to be censored out to obtain $g(x)$ is $\ell_A \ell_B (h_A + h_B)$.

Note that the choice of the order of the states as well as the order of the MAP states from each class in the composite MAP is arbitrary. Then, $Q(x)$ and $R(x)$ matrices can be formed accordingly.

3.3 Numerical Examples

In this section, we present numerical examples to validate the proposed approach. The analysis and simulations were carried out in Matlab. In order to simulate

non-homogeneous processes such workload-dependent MAP, we preferred a time-slot based simulation instead of an event-based approach. Assume that at an instant t in the simulation, the workload-dependent MAP, whose parameters are $(D_0(x), D_1(x))$, is in state i , and the buffer level at t is x . For the time slot $(t, t + \Delta t)$, we generate a transition out of state i with probability $-D_{0,(i,i)}(x)\Delta t$. If a transition is actually to take place, we determine the next state using the vector $\begin{bmatrix} D_{0,(i,1)}(x) & \cdots & D_{0,(i,N)}(x) & D_{1,(i,1)}(x) & \cdots & D_{1,(i,N)}(x) \end{bmatrix}$, which is the i -th row of the matrix $\begin{bmatrix} D_0(x) & D_1(x) \end{bmatrix}$. Note that given a transition occurs, the probability of transmitting into state j , $j \neq i$, from i without an arrival is $-D_{0,(i,j)}(x)/D_{0,(i,i)}(x)$, and the probability of transmitting into state j from i with an arrival is $-D_{1,(i,j)}(x)/D_{0,(i,i)}(x)$. The simulations are run for 10^6 units of simulated time, and we use $\Delta t = 5 \times 10^{-3}$ throughout this chapter.

The first example addresses the FB-CR case for which we fix $B = 10$ and $c(x) = 1 - \frac{1}{2} \sin(\frac{2\pi}{B}x)$ which takes values in $[0.5, 1.5]$ in a single period within $[0, B]$. This function is selected since it drives the buffer level towards the middle of the buffer space $[0, B]$. The job size distribution is of phase-type characterized with the pair $([1 \ 0], \begin{bmatrix} -2 & 2 \\ 0 & -3 \end{bmatrix})$. We define the following function $\mathcal{D}_{\gamma,B}(x) = \frac{1-e^{\gamma x/B}}{1-e^\gamma}$, which takes values in $[0, 1]$ for $x \in [0, B]$. The function $\mathcal{D}_{\gamma,B}(x)$ is an increasing function of x in $[0, B]$, and it is convex (concave) in x when $\gamma > 0$ ($\gamma < 0$). We use $\mathcal{D}_{\gamma,B}(x)$ to experiment with general non-linear continuous functions which can be increasing or decreasing, and convex or concave. For this example, we set

$$\begin{aligned} D_0(x) &= \begin{bmatrix} -d_{11}(x) & \mathcal{D}_{2,10}(x) \\ 1 + \mathcal{D}_{-1,10}(x) & -d_{22}(x) \end{bmatrix}, \\ D_1(x) &= \begin{bmatrix} 2 - \mathcal{D}_{-4,10}(x) & 1 - \mathcal{D}_{-3,10}(x) \\ 1 - \mathcal{D}_{1,10}(x) & \frac{1}{2}(1 - \mathcal{D}_{1,10}(x)) \end{bmatrix}, \end{aligned} \tag{3.35}$$

where $d_{11}(x)$ and $d_{22}(x)$ are selected such that $D_0(x) + D_1(x)$ is stochastic. Note that there are all possible combinations of increasing/decreasing and convex/concave functions in (3.35). The steady-state density of the buffer level seen at an arbitrary time, which we denote by $\mathbf{g}(x) = g(x)\mathbf{1}$, is plotted for varying number of regimes K in Figure 3.2 along with the simulation results. It is clear that the analysis results converge to simulation results as K is increased with the difference between the two vanishing for $K \geq 1024$. Throughout the rest of the numerical examples, we fix

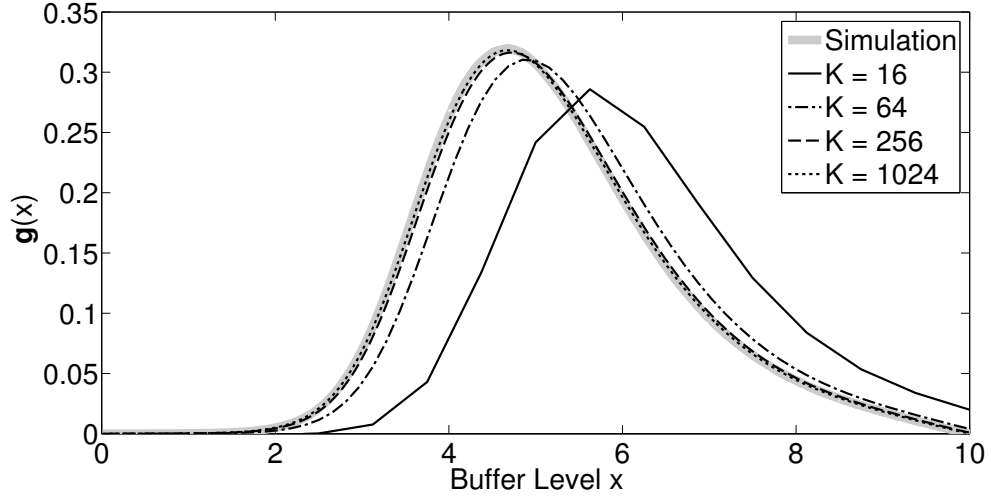


Figure 3.2: Steady-state buffer level pdf for varying number of regimes.

$K = 1024$.

We show the structure and the sparsity of the matrix H , as defined in (2.24), in Figure 3.3 for this example using 4 regimes. Notice that the matrix can be partitioned with square blocks of size 6, which is equal to the number of states.

In the second example, for the two finite buffer models FB-PR and FB-CR, we present three scenarios in which the job size distribution is exponential, Erlangian (E_{10}), and Hyper-exponential (H_2) with a coefficient of variation of 10 with balanced means [88], all having mean job sizes of 1. We test the proposed approach in two different loading scenarios. For this purpose, the MAP in this example is characterized by the matrices in (3.35) for the low loading case, and $D_1(x)$ is doubled for the high loading case along with the corresponding modifications in $d_{ii}(x)$, $i = 1, 2$. The service speed and the buffer capacity are chosen as in the previous example.

The steady-state buffer level density is given in figures 3.4 and 3.5 for low and high loading scenarios, respectively, for FB-CR and FB-PR rejection policies. The proposed approach perfectly captures the pdf of the buffer level for all the tested job size distributions and for both loading scenarios without any numerical stability problems. Moreover, as opposed to FB-PR, the pdf of the buffer level for FB-CR vanishes as the buffer level approaches B due to the complete rejection policy. The

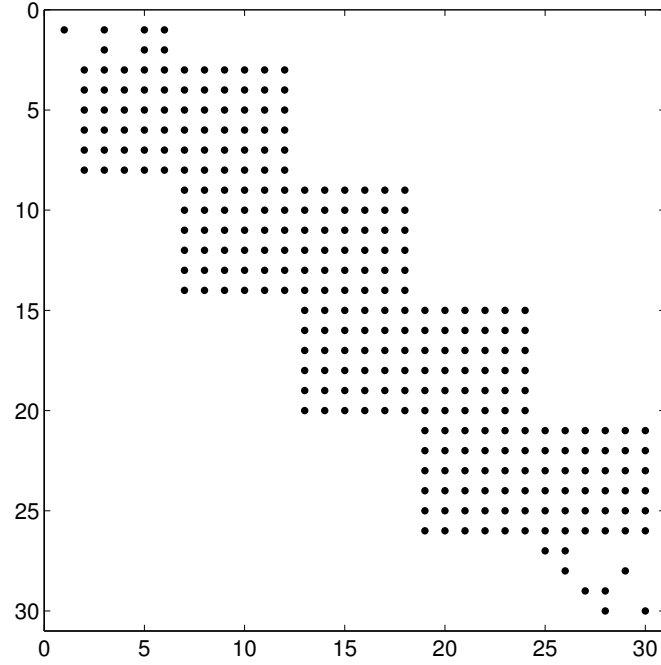


Figure 3.3: Structure of the 30×30 matrix H for $K = 4$. There are six states, and the matrix H can be partitioned with square blocks of size 6.

workload loss probabilities produced by FB-PR (FB-CR) and the corresponding simulation results are given in Table 3.1 (Table 3.2), the latter table also presenting the job loss probability. Note that with FB-CR, larger jobs are more likely to be rejected. This situation is magnified with increased service time variability. If a rejection policy favors smaller jobs as opposed to larger jobs, this would have adverse effect on workload loss probability but may lead to improved overall job loss probability. For example, in Table 2, in low-loading case, the job loss rate is not even monotonically increasing with the service time variability for FB-CR.

For validating the approach for the IB scenario, we use the previous example but with a MAP arrival process characterized by

$$D_0(x) = \begin{bmatrix} -d_{11}(x) & 2 - e^{-x} \\ 3 - e^{-x/2} & -d_{22}(x) \end{bmatrix},$$

$$D_1(x) = \begin{bmatrix} e^{-x} & 2e^{-x/2} \\ \frac{1}{2}e^{-x/10} & e^{-x/5} \end{bmatrix},$$

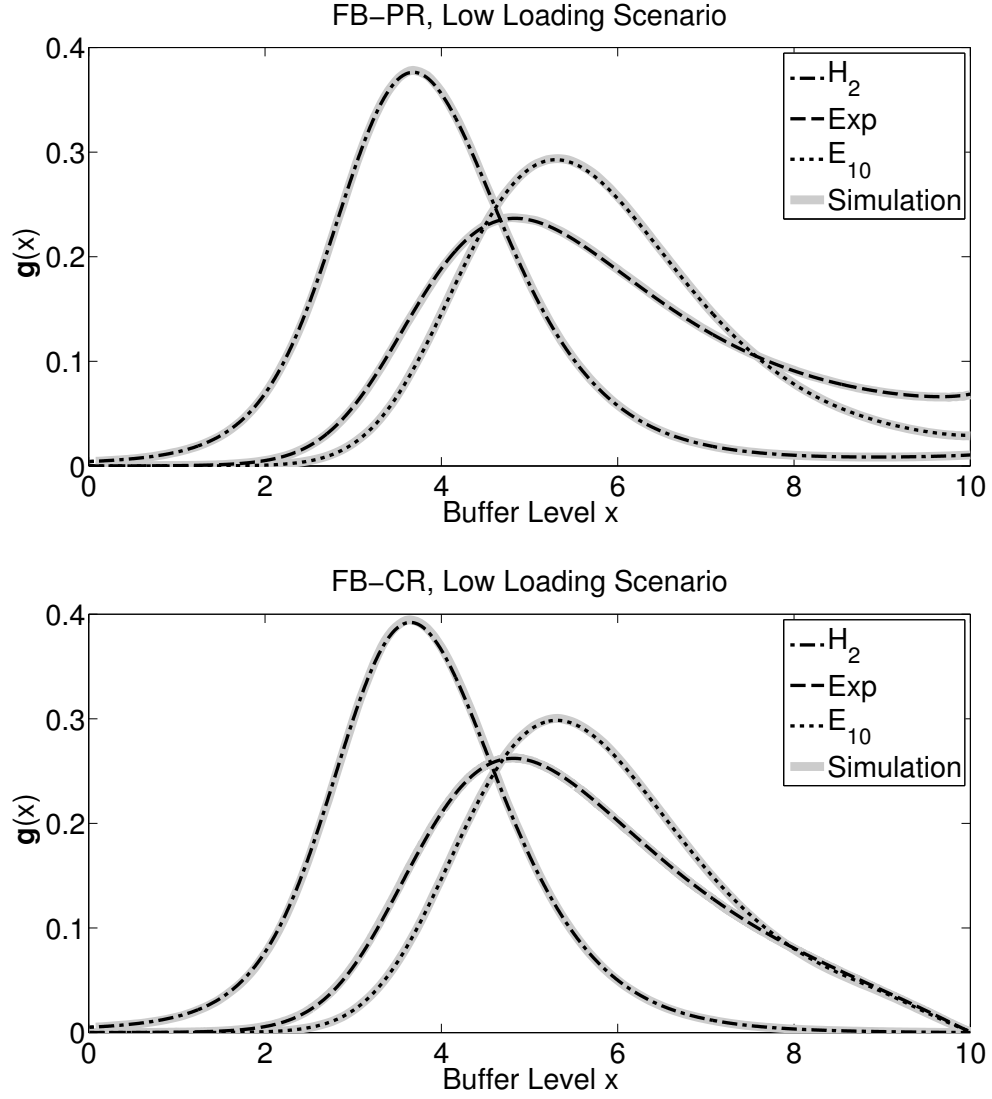


Figure 3.4: Steady-state buffer level pdf for FB-PR and FB-CR under low loading.

		$P_{w, \text{FB-PR}}$	Simulation
High Loading	H_2	4.9714×10^{-1}	$4.9124 \times 10^{-1} \pm 3.1493 \times 10^{-3}$
	Exp	2.9519×10^{-1}	$2.9185 \times 10^{-1} \pm 7.0134 \times 10^{-4}$
	E_{10}	2.6532×10^{-1}	$2.6206 \times 10^{-1} \pm 5.3100 \times 10^{-4}$
Low Loading	H_2	4.7235×10^{-1}	$4.6966 \times 10^{-1} \pm 4.0205 \times 10^{-3}$
	Exp	5.9215×10^{-2}	$5.8706 \times 10^{-2} \pm 2.7172 \times 10^{-4}$
	E_{10}	1.3103×10^{-2}	$1.2856 \times 10^{-2} \pm 1.3227 \times 10^{-4}$

Table 3.1: Workload loss probabilities for FB-PR: simulation results represent 10 runs with 10^6 of simulated time and 99% confidence intervals.

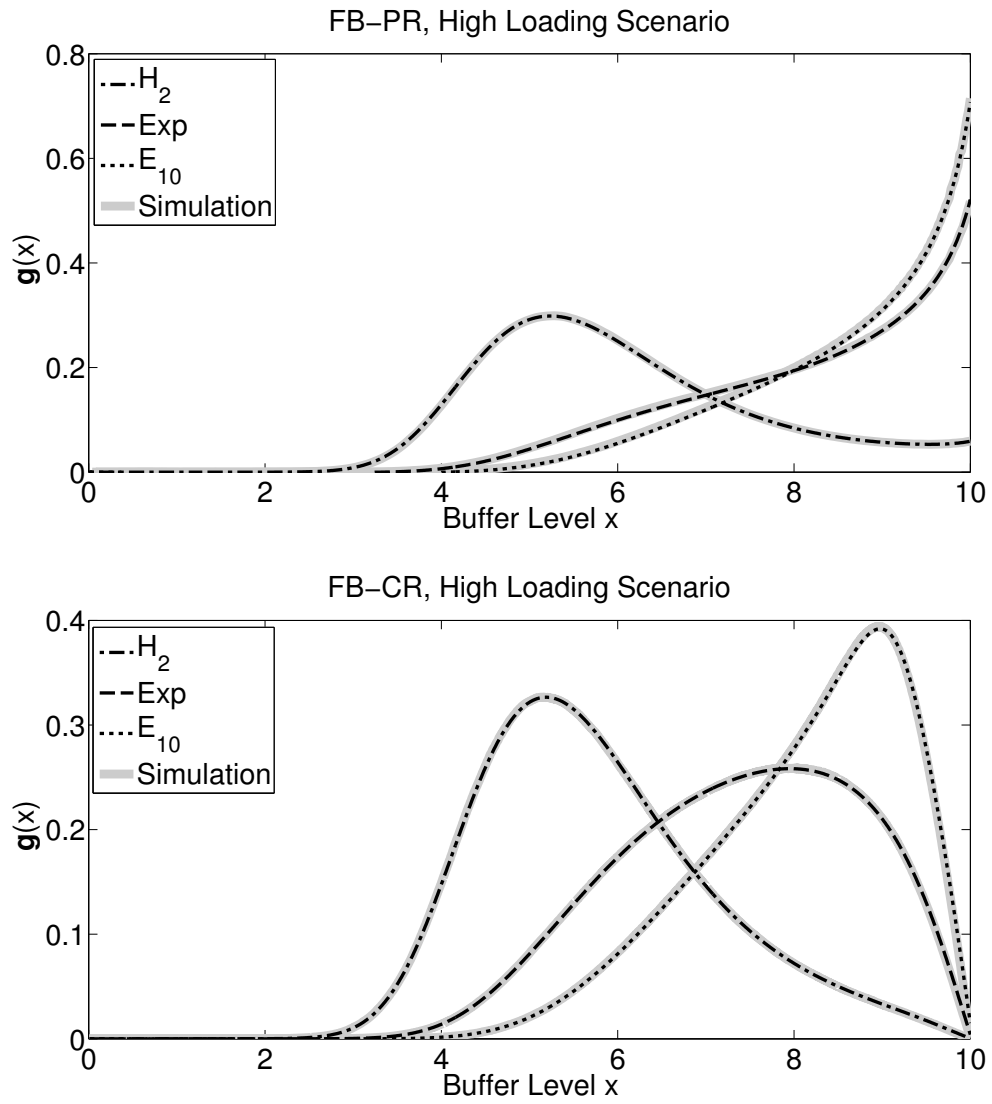


Figure 3.5: Steady-state buffer level pdf for FB-PR and FB-CR under high loading.

		$P_{w, \text{FB-CR}}$	Simulation
High Loading	H ₂	5.1012×10^{-1}	$5.0963 \times 10^{-1} \pm 4.6680 \times 10^{-3}$
	Exp	3.2352×10^{-1}	$3.2191 \times 10^{-1} \pm 5.1730 \times 10^{-4}$
	E ₁₀	2.6643×10^{-1}	$2.6353 \times 10^{-1} \pm 4.4079 \times 10^{-4}$
Low Loading	H ₂	4.9944×10^{-1}	$4.9837 \times 10^{-1} \pm 4.5928 \times 10^{-3}$
	Exp	1.0427×10^{-1}	$1.0374 \times 10^{-1} \pm 4.7205 \times 10^{-4}$
	E ₁₀	1.9091×10^{-2}	$1.8369 \times 10^{-2} \pm 9.7872 \times 10^{-5}$
		$P_{\text{FB-CR}}$	Simulation
High Loading	H ₂	1.1453×10^{-2}	$1.1333 \times 10^{-2} \pm 6.4506 \times 10^{-5}$
	Exp	1.3568×10^{-1}	$1.3457 \times 10^{-1} \pm 3.6109 \times 10^{-4}$
	E ₁₀	2.3274×10^{-1}	$2.2964 \times 10^{-1} \pm 4.4478 \times 10^{-4}$
Low Loading	H ₂	4.8591×10^{-3}	$4.8388 \times 10^{-3} \pm 4.8460 \times 10^{-5}$
	Exp	3.2901×10^{-2}	$3.2606 \times 10^{-2} \pm 1.2858 \times 10^{-4}$
	E ₁₀	1.5654×10^{-2}	$1.5245 \times 10^{-2} \pm 8.3769 \times 10^{-5}$

Table 3.2: Workload and job loss probabilities for FB-CR: simulation results represent 10 runs with 10^6 of simulated time and 99% confidence intervals.

where $d_{11}(x)$ and $d_{22}(x)$ are again selected such that $D_0(x) + D_1(x)$ is stochastic. For the high loading case, we again doubled $D_1(x)$. The service speed is $c(x) = 3 - \frac{1}{2}e^{-x} \cos(2\pi x)$, which has no particular significance other than having a limit as $x \rightarrow \infty$ and being complex enough to produce non-trivial buffer level distributions. The resulting steady-state buffer level pdf obtained by analysis and simulation is depicted in Figure 3.6 which clearly demonstrates the agreement between the two. The behaviors of the systems having different job size distributions are quite different, which is reflected in the pdf plots.

In the third example, we present a workload-bounded buffer example with FB-CR policy for which the number of states of the MAP is varied, but its parameters are not workload-dependent. Note that, this problem can again be modeled by a CFMFQ because of the workload-bounded nature of the system. In this example, the incoming MAP is assumed to be the superposition of N independent IPPs (Interrupted Poisson Process). Each of the N sources goes through alternating on and off intervals which are exponentially distributed with parameters $\lambda_{on} = 0.1$ and $\lambda_{off} = 0.15$, respectively. When a source is on, it generates Poisson arrivals with rate

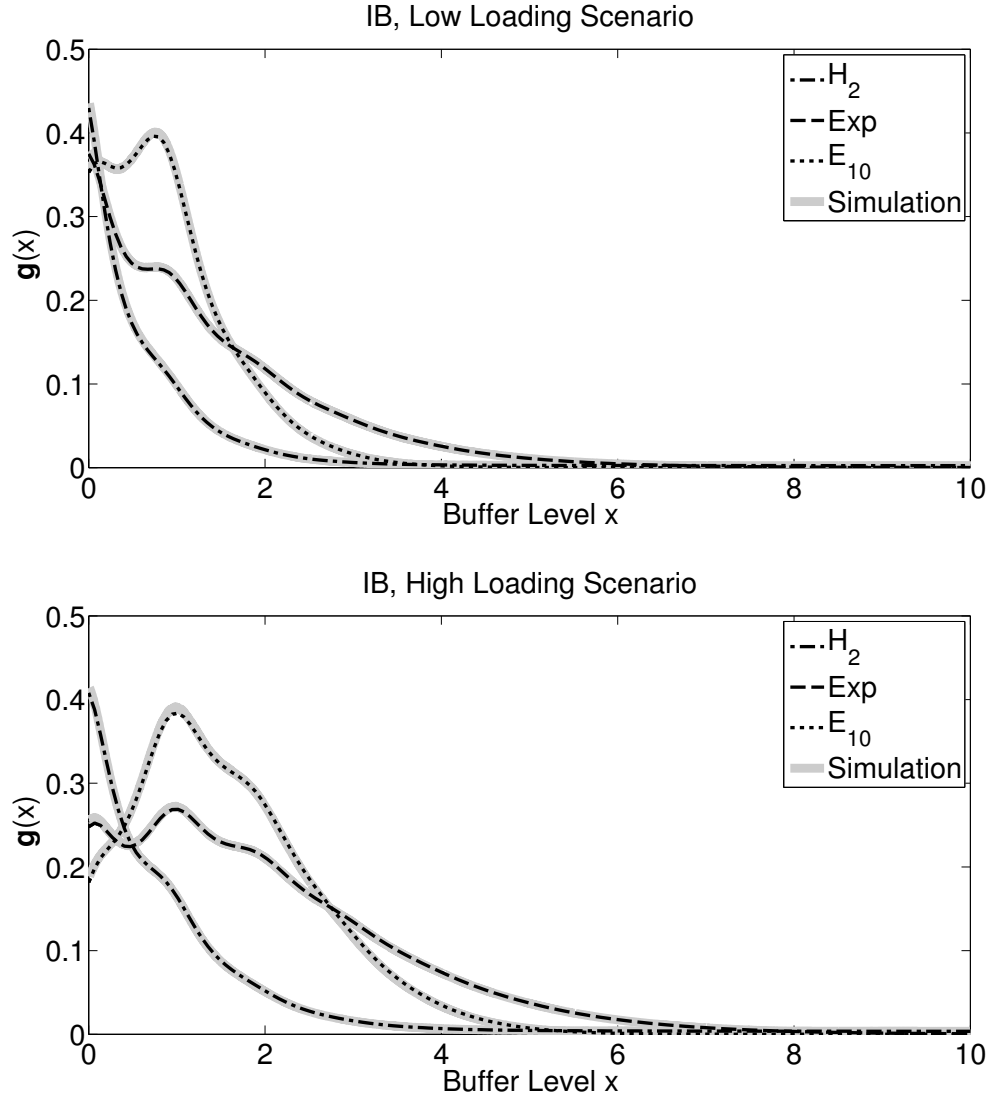


Figure 3.6: Steady-state buffer level pdf for IB under low and high loading. The $T^{(K-1)}$ values found via (2.27) for low and high loading scenarios are 62.2859 and 69.1978, respectively, for $\epsilon = 10^{-3}$. The plots are truncated as the pdf vanishes.

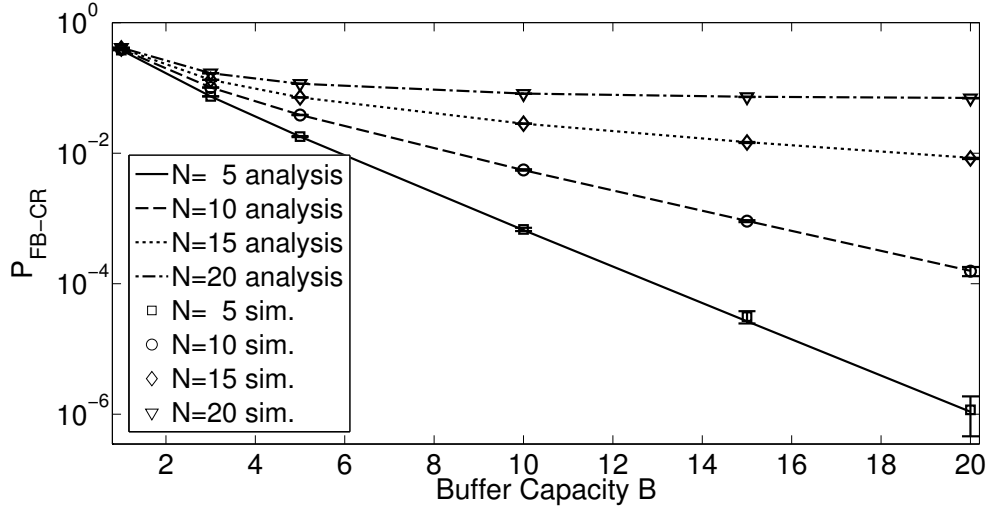


Figure 3.7: Job loss probability for the FB-CR model under various loads (given by $\rho = 0.06N$) and buffer capacities. Simulation results represent 10 runs with 10^6 of simulated time and 99% confidence intervals.

$\theta = 0.1$, and there are no arrivals when the source is off. The job size distribution is exponentially distributed with parameter $\mu = 1$, and the service rate is $c = 1$. The offered load to the buffer is

$$\rho = \frac{N\theta\lambda_{off}/\lambda_{on}}{1 + (\lambda_{off}/\lambda_{on})} \frac{1}{\mu},$$

which equals $0.06N$ for the values we chose. We plot the job loss probability for various values of N and B in Figure 3.7. The simulation data is obtained with 10 runs of 10^6 seconds for each point, and 99% confidence intervals are provided on the figure. This example demonstrates that our method is able to accurately solve workload-dependent buffers with MAP arrivals of order larger than two, again without numerical stability problems.

As for the fourth example, we analyze a finite buffer system with complete rejection whose arrivals occur according to an MMPP with N states. The job size has H_2 distribution with mean 5 and coefficient of variation 10 with balanced means. Enumerating the states of the MMPP from 1 to N , a state i transits into all other states with rate i , and leads to an arrival with rate $(\frac{1}{B} - \frac{2}{B} \frac{i-1}{N-1})x + \frac{i-1}{N-1}$ within $0 \leq x \leq B$, where B is the buffer capacity, and selected as 10 for this example. In other words, the arrival rate in state 1 linearly goes from 0 up to 1 within $[0, B]$, the arrival rate in state N goes linearly from 1 down to 0 within $[0, B]$, and the rates of the rest of the

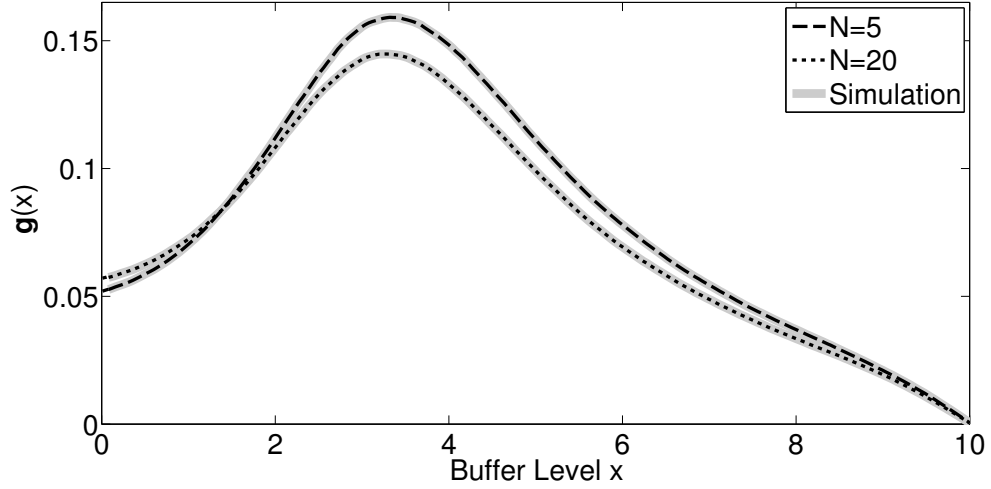


Figure 3.8: Steady-state buffer level pdf for the system with MMPP arrivals for $N = 5$ and 20.

states have equally spaced slopes that fall within $1/B$ and $-1/B$ while all rate functions intersect at the point $(B/2, 1/2)$. The service speed is taken $c(x) = 1 - \frac{1}{2} \sin(\frac{2\pi}{B}x)$ as before. The resulting pdfs for two values of N are given in Figure 3.8.

Besides demonstrating that our method is able to accurately solve workload-dependent buffers with a large number of states, we also provide a computational run time analysis with this example. Run times for various values of N and K are presented in Table 3.3 which demonstrates the linear dependence of the computational complexity of the proposed approach on the number of regimes. The run times for each case is provided in terms of overall run time, and the three major steps that constitute the proposed method: Constructing the block-tridiagonal matrix from the boundary conditions (Matrix Fill), the block-LU factorization, and the normalization of the solution stemming from (3.9). It is clear that the most computation-intensive step is the Matrix Fill, which consists of a Schur decomposition and a Sylvester equation for each regime in addition to the calculation of $V^{(k)}(x)$ matrices, a procedure involving matrix exponentiation. We refer the reader to [15] for the computational complexity and stability of the associated numerical algorithms.

N	K	Matrix Fill	Block LU	Normalization	Overall
5	256	0.6245	0.0318	0.2284	0.8918
	512	1.2618	0.0668	0.4593	1.7948
	768	1.8902	0.1063	0.6799	2.6842
	1024	2.5127	0.1472	0.8951	3.5635
10	256	0.8897	0.0622	0.2722	1.2327
	512	1.7840	0.1279	0.5401	2.4631
	768	2.6942	0.1979	0.8059	3.7114
	1024	3.6196	0.2721	1.0772	4.9853
15	256	1.2297	0.1100	0.3612	1.7133
	512	2.4828	0.2278	0.7021	3.4304
	768	3.7789	0.3559	1.0391	5.1969
	1024	5.0898	0.4861	1.3930	6.9971
20	256	1.6526	0.1654	0.4426	2.2772
	512	3.3454	0.3406	0.8695	4.5815
	768	5.0736	0.5262	1.2887	6.9237
	1024	6.8673	0.7234	1.7191	9.3542

Table 3.3: Run times in units of seconds for various values of N and K obtained on a PC with Intel Core i7, 2.20 GHz processor and 8 GB RAM.

We solve a two class system in FB-CR setting in the last example. The buffer capacity is 10, and the parameters are as follows:

$$\begin{aligned}
D_{0,A} &= \begin{bmatrix} -0.6 & 0.4 \\ 0.2 & -0.8 \end{bmatrix}, & D_{1,A} &= \begin{bmatrix} 0.2 & 0 \\ 0.4 & 0.2 \end{bmatrix}, \\
D_{0,B} &= \begin{bmatrix} -1.6 & 0.4 & 0.6 \\ 0.2 & -1.8 & 0.4 \\ 0.2 & 0.2 & -2 \end{bmatrix}, & D_{1,B} &= \begin{bmatrix} 0.2 & 0 & 0.4 \\ 0.4 & 0.2 & 0.6 \\ 0.2 & 1 & 0.4 \end{bmatrix}, \\
\alpha_A &= [0.87457 \quad 0.12543], & T_A &= \begin{bmatrix} -3 & 1 \\ 2.5 & -3.5 \end{bmatrix}, \\
\alpha_B &= [1 \quad 0 \quad 0], & T_B &= \begin{bmatrix} -10 & 10 & 0 \\ 0 & -10 & 10 \\ 0 & 0 & -10 \end{bmatrix}.
\end{aligned}$$

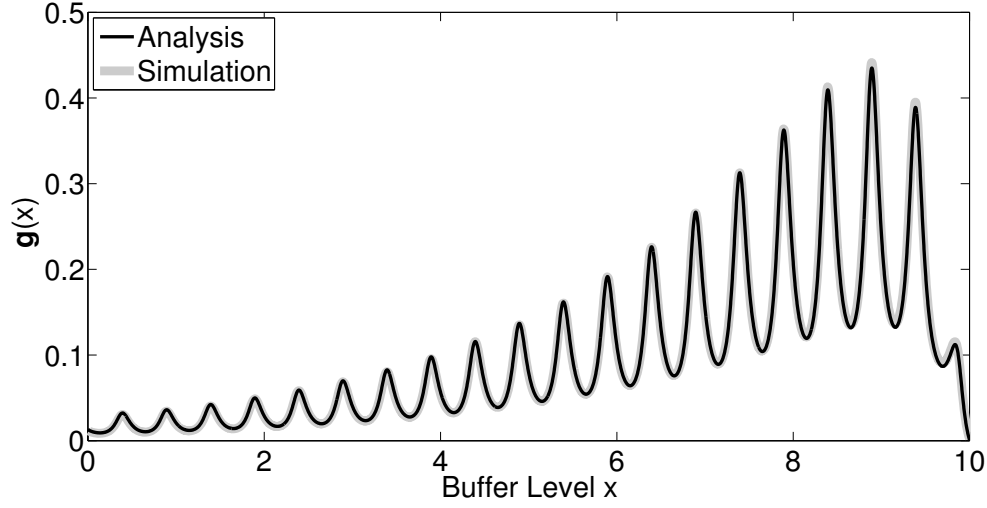


Figure 3.9: Steady-state buffer level pdf for the two class system in FB-CR setting.

We selected the service speed as $c(x) = 1.5 + 0.8 \sin(4\pi x)$, which is a rapidly oscillating function, in order to stress test our method. The steady-state buffer level pdf is plotted in Figure 3.9, which shows an agreement between the analysis and the simulation result. Therefore, we conclude that our method works also with multiple classes and is able to capture the behaviors of systems with rapidly varying parameters. Note that even though none of the parameters except for one depend on the buffer level, the system is still a CFMFQ due to the dependence of the service speed on the buffer level. Moreover, even without this dependence, the system would still be a CFMFQ due to the complete rejection policy.

3.4 Conclusion

In this part of the study, using sample path arguments, we reduce the steady-state analysis of a workload-dependent buffer with MAP arrivals and PH-type distributed job sizes to that of a CFMFQ. Then, we appropriately approximate the CFMFQ using uniform discretization by a K -regime MRMFQ and use the existing boundary conditions to solve for the MRMFQ. Moreover, while solving the boundary equations

of the MRMFQ, we take advantage of the block-tridiagonal structure of the equations in order to reduce the computational complexity to $O(K)$. With this method, large number of regimes (in the order of thousands) can be used relatively easily for discretization. we have demonstrated perfect match with simulation results in all scenarios we tested provided that K is large enough.

Future work will be directed towards applications of the proposed method for the analysis of real-world systems in which controlled queues play a major role. Other directions of future research are the exploration of different discretization techniques used for reducing CFMFQs to MRMFQs, and the comparison of our method to existing CFMFQ solvers in terms of accuracy, numerical stability and computational complexity. In addition, as we have shown that the matrix exponentiation operation constitutes a significant portion of the overall run time for the method we propose, there seems to be room for improvement by replacing the procedure Matlab prefers to compute matrix exponentiation with approximate, but faster numerical procedures [89, 90].

Chapter 4

Analysis of Horizon-Based Reservation in OBS Networks

4.1 Introduction

Optical Burst Switching (OBS) [26] offers a compromise between optical packet switching that has yet to be implemented due to lack of optical buffers, and the relatively IP-unfriendly optical circuit switching. In the OBS paradigm, a number of packets are merged into a single payload, called a burst. A control packet, called the Burst Control Packet (BCP), is sent in the electronic domain in advance of the burst with relevant information in order to reserve resources for the burst. Optical nodes receiving the BCP configure themselves to accommodate the burst to arrive, if possible. The burst is then sent in the optical domain after an amount of time, called the offset time.

There are different reservation mechanisms for node configuration. A sample scenario with three reservation mechanisms, Just-in-Time (JIT) [27], Horizon [29] and Just-Enough-Time (JET) [28, 26], was given in Figure 1.6 (page 12).

The simplest is the Just-in-Time (JIT) mechanism. If the optical node is free upon the arrival of a BCP, JIT reserves the node for the incoming burst. Then, the

node becomes “reserved”. During the period of time a node is reserved, all arrivals are blocked. After the transmission of the burst that reserved the node is completed, the node becomes free again. In the example in Figure 1.6(a), the burst B1 arrives at time 0 and the node becomes reserved for 6 time units. Within this period, two BCPs arrive both of which are blocked.

Horizon-based reservation is a delayed reservation method, in which parameter called the horizon is used to keep track of the remaining time the node is reserved. Upon an arrival of a burst, its offset time is compared to the horizon parameter. If the offset time is larger than the horizon, that means that the burst will arrive after the node will become free. In this case, the burst is accepted, and the horizon is updated to the sum of the offset time and the burst length in time units, since this quantity is now the new time that the optical node will become free after it serves the newly accepted burst. In the example in Figure 1.6(b), the burst B2 is accepted since the offset time, 5, is larger than the horizon value at the time of its arrival, which is 3. On the other hand, the burst B3 is blocked since its offset time, 2, is less than the horizon value at the time of its arrival, which is 8.

Notice that burst B3 in this scenario can fit into the idle period, called the “void”, between B1 and B2. However, horizon-based reservation has no mechanism to make use of the voids. In contrast, Just-Enough-Time (JET) supports void-filling, for which an example is given in Figure 1.6(c). Horizon is easier to implement than JET since it does not deal with voids, and we will focus on the analysis of horizon-based reservation in this thesis.

Another feature of OBS networks is fiber delay lines (FDL), which are basically coils of fiber that induce a fixed amount of delay on a burst that traverses it. Using FDLs, contention can be resolved among multiple bursts that arrive at a node within each other’s duration. One of the contending bursts is chosen to be transmitted right away (or is already on the channel), and the rest is “stored” within FDLs, meaning that they are forwarded to FDLs of appropriate sizes that can assure that each burst, upon leaving the FDL it had been fed to, finds the channel available. A sample scenario with the horizon-based reservation mechanism in the presence of FDLs can be seen in Figure 1.8 (page 15). Bursts that are to arrive before the node

becomes idle may be accepted if there are FDLs with sufficient delays that would store the burst until the node is freed. On the other hand, if the horizon is greater than the offset time of a burst even after the maximum delay achievable with the FDLs, then that burst is blocked.

For better performance, a number of FDLs can be available at an optical node. These FDLs can be configured to provide degenerate buffering or non-degenerate buffering. In degenerate buffering, each FDL provides an integer multiple of a fixed delay. In other words, if there are N FDLs, the delay line i , $1 \leq i \leq N$, provides a delay of $i\Delta$, where Δ is a fixed quantity and called the granularity parameter. On the other hand, in non-degenerate buffering, the delays each FDL provides can be arbitrary.

In case of contention, the choice of the FDL impacts the overall delay performance of the system. Obviously, the delay caused by the chosen FDL should be sufficient to guarantee a free channel to the burst that is fed into the FDL. On the other hand, among the FDLs satisfying this condition, the FDL with the least amount of delay should be chosen in order not to unnecessarily increase the overall delay the burst experiences.

For instance, consider a scenario in which horizon-based reservation is employed along with degenerate buffering with N FDLs. Assume that a BCP arrives indicating that a burst is to arrive with offset time $T_O < T_H$, where T_H denotes the horizon parameter. Then, the burst should be fed to FDL i , where $i = \lceil (T_H - T_O) / \Delta \rceil$ and Δ is the granularity parameter. In this case, the horizon is updated to be $T_O + i\Delta + T_B$, where T_B denotes the burst length in units of time.

In this study, we present a two-part analysis of the horizon-based reservation scheme on a single-channel OBS system. First, the system is modeled without FDLs. In this model, the offset times are allowed to have general distributions, while the burst arrivals occur according to a Markovian Arrival Process (MAP) and the burst lengths possess a phase-type (PH-type) distribution. Moreover, this analysis is extended to multiple traffic classes with different arrival processes, and burst length and offset time distributions. Using the results of this analysis, we conclude that deterministic offset times perform better than stochastic offset times in terms of overall blocking probability, and in two-class scenarios, class differentiation .

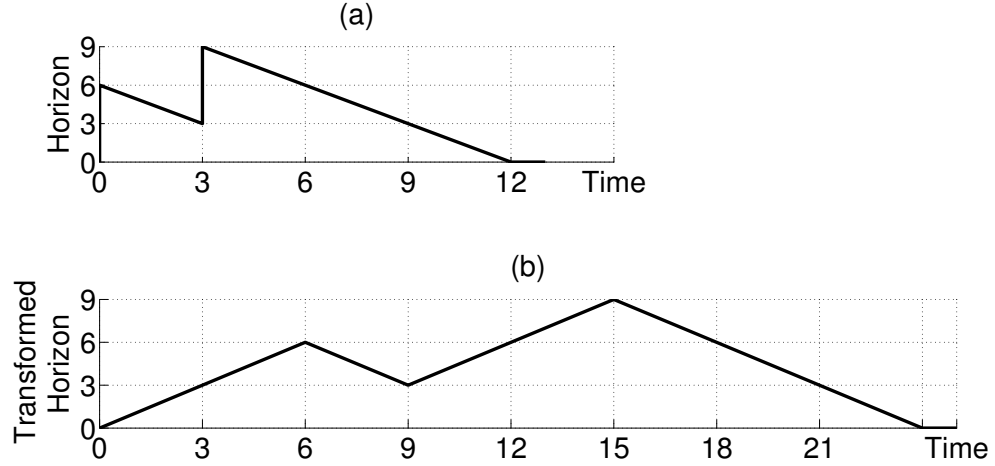


Figure 4.1: The sample path of (a) the horizon parameter and (b) its transformed counterpart. Scenario explained in Figure 1.6 (page 12).

In the second part, we model the horizon-based reservation scheme in the presence of FDLs. This model is also extended to multiple traffic classes. Due to our observation about deterministic offset times, we model the offset times deterministic, or having a distribution that can be expressed as a sum of impulses.

4.2 Stochastic Model for Horizon-based Reservation with No Fiber Delay Lines

When a BCP arrives at an optical node, the corresponding burst is accepted if the horizon of the node is smaller than the offset time of the burst, since the node will not be busy when the burst arrives. In this case, the horizon of the optical node is updated to be equal to the sum of the offset time and the burst length. Otherwise, the burst is said to be blocked.

In order to clarify the stochastic model, we revisit the example given in Figure 1.6 (page 12). In Figure 4.1(a), we plot the evolution of the horizon parameter. As explained in section 1.1 (see Figure 1.4), we replace the abrupt jumps in the sample path of the horizon with linear ascents to obtain the transformed horizon plotted in Figure 4.1(b). The transformed horizon can be modeled with a CFMFQ.

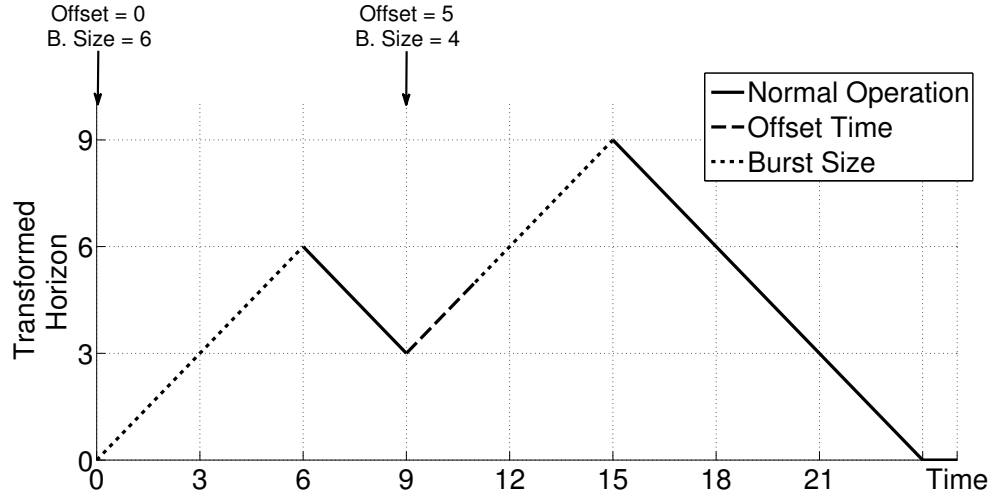


Figure 4.2: The sample path of the horizon parameter and its transformed counterpart.

To construct the CFMFQ modeling the transformed horizon, we first need to understand the behavior of the system. In normal operation, the transformed horizon is decreased with unity rate as long as it is nonzero. Upon a BCP arrival, if the offset time exceeds the instantaneous transformed horizon value, the burst is accepted. At this point, the transformed horizon starts increasing with unity rate until it reaches the offset time. Then, it continues to increase for an amount equal to the burst length. The breakdown of the evolution of the transformed horizon is depicted in Figure 4.2.

Therefore, there will be three subsets of states in the background process of the CFMFQ:

- (i) The states that govern the normal operation, which is the horizon decreasing with unity rate. These states will be made of from the states of the BCP arrival process, and will have -1 drifts.
- (ii) The states that take the instantaneous horizon value to the offset time of the burst. The drift in these states will be $+1$.
- (iii) The states that add the burst length. These states will be made of from the states of the PH-type process that defines the distribution of the burst length,

and will have +1 drifts.

Let $\theta(x)$ and $\Theta(x)$ denote the pdf and the cdf of the offset time distribution. If a burst corresponding to an arriving BCP is accepted, it means that the offset time, T_O , is greater than the horizon. We can write the probability of the offset time expiring in an infinitesimal period between the horizon values x and $x + dx$, given that the offset time “survives” the horizon value x as

$$\lim_{dx \rightarrow 0} \frac{\Pr\{x < T_O \leq x + dx\}}{\Pr\{T_O > x\}} = \frac{\theta(x)}{1 - \Theta(x)} dx.$$

Therefore, the rate out of the hazard state is

$$H(x) = \frac{\theta(x)}{1 - \Theta(x)}. \quad (4.1)$$

Incidentally, $H(x)$, as defined in (4.1), is called the *hazard rate* (or failure rate) for distribution $\theta(x)$.

Now, we can fully describe the CFMFQ modeling the horizon. We assume that the bursts arrive according to a MAP with parameters D_0 , D_1 , which are $\ell \times \ell$. The burst length has a PH-type distribution with parameters α , T , having h phases (excluding the absorbing state). Then, the infinitesimal generator of the background process is

$$Q(x) = \begin{bmatrix} I_\ell \otimes T & 0 & I_\ell \otimes T^0 \\ H(x)I_\ell \otimes \alpha & -H(x)I_\ell & 0 \\ 0 & D_1(x) & D_0(x) \end{bmatrix}, \quad (4.2)$$

where I_m denotes the $m \times m$ identity matrix, and

$$T^0 = -T\mathbf{1}, \quad (4.3)$$

$$D_1(x) = D_1(1 - \Theta(x)),$$

$$D_0(x) = D_0 + D_1\Theta(x),$$

and the diagonal rate matrix is given by

$$R(x) = R = \begin{bmatrix} I_{h\ell+\ell} \\ -I_\ell \end{bmatrix}.$$

Notice that the $Q(x)$ matrix varies continuously with x , resulting in a CFMFQ. This feedback arises due to the system being a blocking system.

In this model, the first $h\ell$ states correspond to the ℓ replicas of the PH-type distribution of the burst length, and the next ℓ states represent the offset time. Recall that the need for the PH-type replicas was explained in section 3.2.1 (see Figure 3.1 on page 60).

Note that $\theta(x)$ is assumed to be smooth and of infinite support in this formulation. In addition, we note the following cases:

- $\theta(x)$ is composed of a number of impulses.

In this case, regime boundaries are defined at the points where there is an impulse in $\theta(x)$. One hazard state for each impulse is defined. Upon an arrival, the background process enters one of these according to their corresponding probabilities. Then, the background process does not leave this state until it hits the corresponding regime boundary. At this boundary, it transits into the subset of burst length states. This will be formulated in more detail later in the analysis with FDLs. Note that a special subcase of this where the offset times are deterministic was investigated by Kankaya and Akar [43].

- $\theta(x)$ is of finite support.

Two solutions can be employed in this case. The first is to just use the hazard rate as defined in (4.1). The problem with this is that the hazard rate become infinite to the right of the support of $\theta(x)$. For instance, assume $\theta(x)$ is the uniform distribution over $[0, 1]$. Then, when $0 \leq x < 1$, we have $H(x) = 1/(1 - x)$. Notice that $H(x)$ approaches infinity as x approaches 1. When $x > 1$, $H(x)$ becomes infinity. Then, a large number instead of infinity can be used as $H(x)$. We used this method in the examples to come as it is easier.

In the second method, a regime boundary is defined at the upper limit of the support of $\theta(x)$. When x is less than this value, $H(x)$ as given in (4.1), which produces $H(x) = 1/(1 - x)$ is used. If the horizon makes it to this regime boundary, it can be made to get stuck there by defining the drift at this regime boundary to be 0. Then, while the horizon is kept at the

regime boundary, the background process can transit into the subset of burst length states.

- $\theta(x)$ is composed of a mixture of these two.

In this case, a combination of the methods mentioned can be used.

After the formulation of the CFMFQ, the solution is obtained by the methods described in sections 2.3.1, 2.3.3, and 2.4. Once the steady-state pdf vector of the horizon, denoted by $g(x)$, is found, the burst blocking probability can be computed via

$$p_B = \frac{\int_0^\infty g(x) D_1 \mathbf{1} \Theta(x) dx}{c^{(0)} D_1 \mathbf{1} + \int_0^\infty g(x) D_1 \mathbf{1} dx}. \quad (4.4)$$

In addition, the blocking probability conditioned on the offset time can also be computed easily. When the offset time is known to be equal to a known value, say t_o , the burst will be blocked if the horizon is larger than this value. Hence, the conditional blocking probability is equal to the complementary cumulative distribution function of the horizon:

$$p_{B|T_O=t_o} = \int_{t_o}^\infty g(x) \mathbf{1} dx.$$

It is also possible to model multiple classes that have different arrival processes, offset time and burst length distributions with this method. Here, we formulate a system with two classes. Formulating systems with more than two classes is not much difficult, although they might lack closed forms. In that case, enumerating the states and building the $Q(x)$ and R matrices algorithmically should be the preferred method.

For classes A and B, let the MAP matrix pairs be denoted by (D_{0A}, D_{1A}) and (D_{0B}, D_{1B}) with respective sizes ℓ_A and ℓ_B , the PH-type parameters by (α_A, T_A) and (α_B, T_B) with respective number of phases h_A and h_B , and the offset time pdf and cdf's by $\theta_A(x)$, $\Theta_A(x)$ and $\theta_B(x)$, $\Theta_B(x)$ respectively. Then, the matrices describing

the CFMFQ are given by

$$Q(x) = \begin{bmatrix} I_{\ell_A \ell_B} \otimes T_A & 0 & 0 & 0 & I_{\ell_A \ell_B} \otimes T_A^0 \\ 0 & I_{\ell_A \ell_B} \otimes T_B & 0 & 0 & I_{\ell_A \ell_B} \otimes T_B^0 \\ H_A(x) I_{\ell_A \ell_B} \otimes \alpha_A & 0 & -H_A(x) I_{\ell_A \ell_B} & 0 & 0 \\ 0 & H_B(x) I_{\ell_A \ell_B} \otimes \alpha_B & 0 & -H_B(x) I_{\ell_A \ell_B} & 0 \\ 0 & 0 & I_{\ell_B} \otimes D_{1A}(x) & D_{1B}(x) \otimes I_{\ell_A} & D_0(x) \end{bmatrix}$$

$$R = \begin{bmatrix} I_{(h_A+h_B+2)\ell_A \ell_B} \\ -I_{\ell_A \ell_B} \end{bmatrix},$$

where

$$\begin{aligned} T_A^0 &= -T_A \mathbf{1}, & T_B^0 &= -T_B \mathbf{1}, \\ H_A(x) &= \frac{\theta_A(x)}{1 - \Theta_A(x)}, & H_B(x) &= \frac{\theta_B(x)}{1 - \Theta_B(x)}, \\ D_{1A}(x) &= D_{1A}(1 - \Theta_A(x)), & D_{1B}(x) &= D_{1B}(1 - \Theta_B(x)), \\ D_{0A}(x) &= D_{0A} + D_{1A} \Theta_A(x), & D_{0B}(x) &= D_{0B} + D_{1B} \Theta_B(x), \\ D_0(x) &= I_{\ell_B} \otimes D_{0A}(x) + D_{0B}(x) \otimes I_{\ell_A}. \end{aligned}$$

To compute the blocking probability, individual distributions corresponding to the separate classes should be computed from the composite distribution. Note that in the enumeration we used, the states from 1 upto $\ell_A \ell_B$ correspond to the state pairs $(1, 1), (2, 1), \dots, (\ell_A, 1), (1, 2), \dots, (\ell_A, \ell_B)$, respectively. Defining

$$g_A(x) = \begin{bmatrix} g_{1,A}(x) & \cdots & g_{\ell_A,A}(x) \end{bmatrix}, \quad (4.5)$$

$$g_B(x) = \begin{bmatrix} g_{1,B}(x) & \cdots & g_{\ell_B,B}(x) \end{bmatrix}, \quad (4.6)$$

these individual distributions can be found from $g(x)$ via

$$g_{i,A}(x) = \sum_{j \in S_i^A} g_j(x), \quad 1 \leq i \leq \ell_A, \quad (4.7)$$

$$g_{i,B}(x) = \sum_{j \in S_i^B} g_j(x), \quad 1 \leq i \leq \ell_B, \quad (4.8)$$

where S_i^A (S_i^B) is the set of states of the composite MAP in which the MAP governing class A (B) is in state i . A similar procedure can be defined for the probability masses

at 0, $c_A^{(0)}$ and $c_B^{(0)}$. Then, the blocking probability for each individual class and the overall blocking probability can be expressed as

$$\begin{aligned} p_{B(A)} &= \frac{\int_0^\infty g_A(x) D_{1A} \mathbf{1} \Theta_A(x) dx}{c_A^{(0)} D_{1A} \mathbf{1} + \int_0^\infty g_A(x) D_{1A} \mathbf{1} dx}, \\ p_{B(B)} &= \frac{\int_0^\infty g_B(x) D_{1B} \mathbf{1} \Theta_B(x) dx}{c_B^{(0)} D_{1B} \mathbf{1} + \int_0^\infty g_B(x) D_{1B} \mathbf{1} dx}, \\ p_B &= \frac{\int_0^\infty g_A(x) D_{1A} \mathbf{1} \Theta_A(x) dx + \int_0^\infty g_B(x) D_{1B} \mathbf{1} \Theta_B(x) dx}{c_A^{(0)} D_{1A} \mathbf{1} + \int_0^\infty g_A(x) D_{1A} \mathbf{1} dx + c_B^{(0)} D_{1B} \mathbf{1} + \int_0^\infty g_B(x) D_{1B} \mathbf{1} dx}. \end{aligned} \quad (4.9)$$

As a final note, we point out that for a system with N classes, the total number of states of the background process will be $(1 + N + \sum_{i=1}^N h_i) \prod_{i=1}^N \ell_i$ with $\prod_{i=1}^N \ell_i$ of them having -1 drifts.

4.3 Numerical Examples for Horizon-based Reservation with No Fiber Delay Lines

First, we start with a single class example in which the arrivals are MAP, the burst lengths are PH-type distributed and the offset times are uniform on $[0, 3]$. The matrices defining the MAP are

$$D_0 = \begin{bmatrix} -2\rho & 0 \\ 2\rho & -2\rho \end{bmatrix}, \quad D_1 = \begin{bmatrix} 0 & 2\rho \\ 0 & 0 \end{bmatrix},$$

and the parameters of the PH-type distribution are

$$\alpha = \begin{bmatrix} 0.8 & 0.2 \end{bmatrix}, \quad T = \begin{bmatrix} -2 & 1 \\ 0 & -1 \end{bmatrix}.$$

The mean of the burst length distribution is 1, which means that the load is ρ . We obtained two sets of results for (i) load values between 0.02 and 0.2 with increments of 0.02 (low loads), and (ii) load values between 0.2 and 2 with increments of 0.2 (moderate to high loads). The pdf plots for the loads 0.2, 0.8 and 2 are given in Figure 4.3. The agreement between the analytical and simulated results is obvious, thus validating our method.

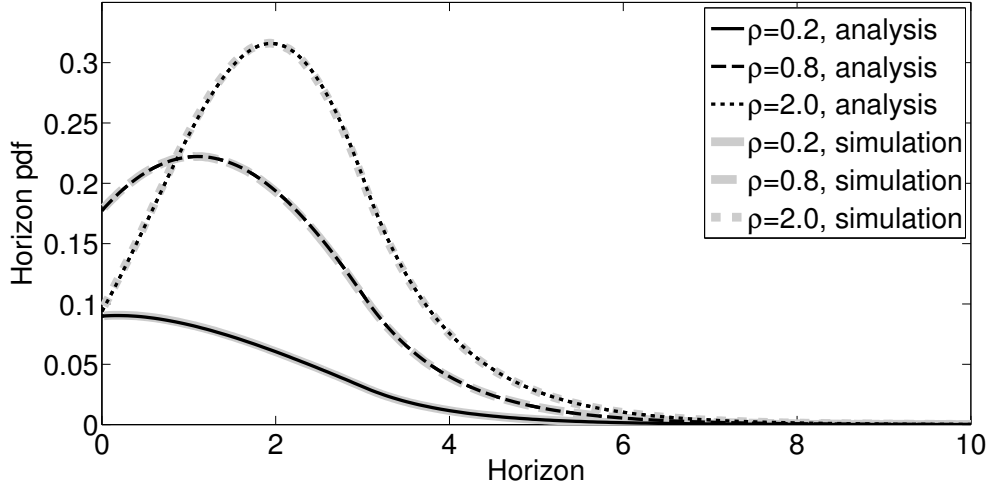


Figure 4.3: Horizon pdf plots for 3 different load values.

Within this scenario, we also would like to present the burst blocking probability against load. First, we present in Figure 4.4 the burst blocking probability for $\rho = 0.02$ computed by our method with various number of regimes. As seen, the blocking probability converges as the number of regimes grow. It is possible to use number of regimes as high as 2^{15} due to the fact that the block-tridiagonal LU decomposition has linear time complexity in the number of regimes. As a result, one can pick a suitable value for the number of regimes in line with their required level of accuracy.

We give the burst blocking probability against load in figures 4.5 and 4.6, for low and moderate to high loads respectively. Figures given in Figure 4.5 were obtained with $K = 2^{14}$ regimes and 10^8 seconds of simulated time, whereas figures given in Figure 4.6 were obtained with $K = 2^{10}$ regimes and 10^6 seconds of simulated time. $T^{(K-1)}$ value was selected in accordance with (2.27) in all examples to follow, although the plots are truncated.

We also give the burst blocking probability conditioned on the offset time for $\rho = 2$ in Figure 4.7. Note that obtaining this plot via simulation would require exceptionally long simulations, which we did not pursue.

Next, we investigate the effect of shifting the offset time distribution on the blocking probability. In this example, the arrivals are Poisson with rate 0.1, the burst

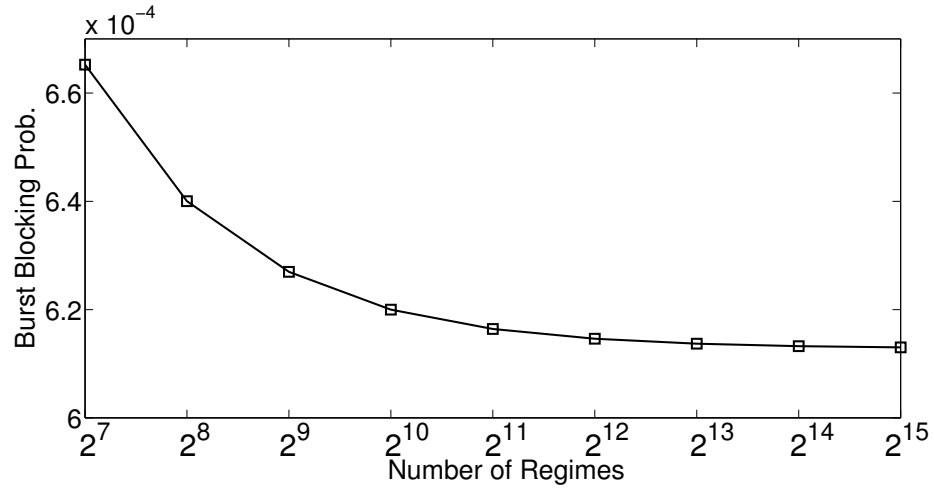


Figure 4.4: Burst blocking probability for $\rho = 0.02$ found via analysis with different number of regimes.

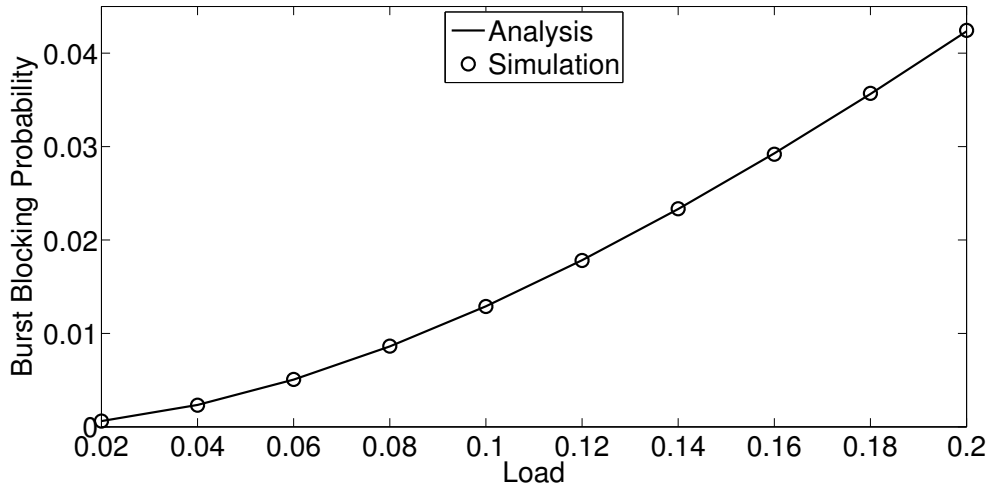


Figure 4.5: Burst blocking probabilities for different load values: Low loads. $K = 2^{14}$ regimes are used for the analysis and 10^8 seconds were simulated.

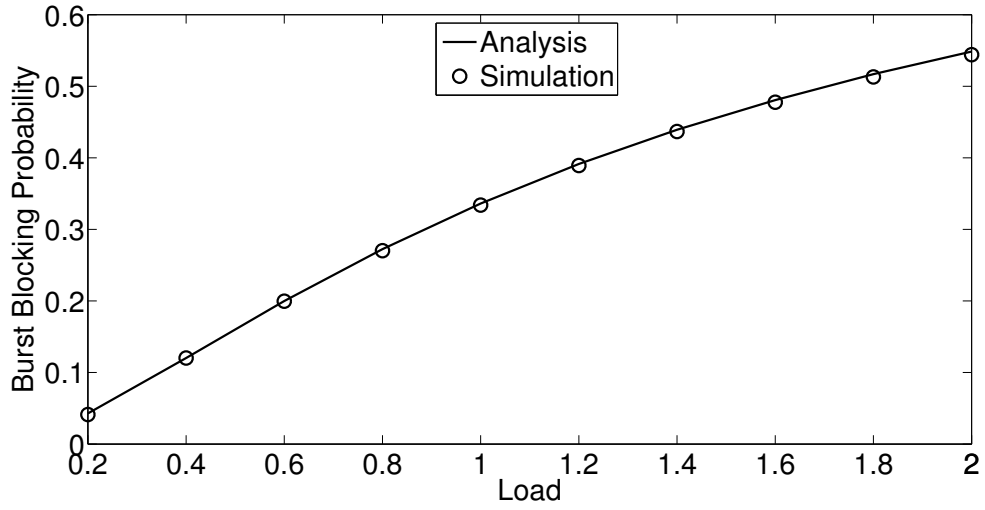


Figure 4.6: Burst blocking probabilities for different load values: Moderate to high loads. $K = 2^{10}$ regimes are used for the analysis and 10^6 seconds were simulated.

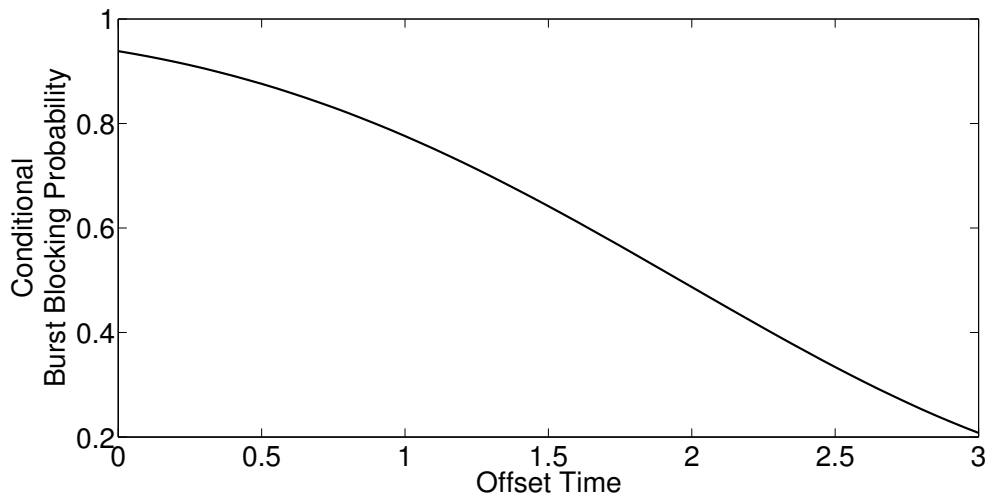


Figure 4.7: Burst blocking probability conditioned on the offset time for $\rho = 2$.

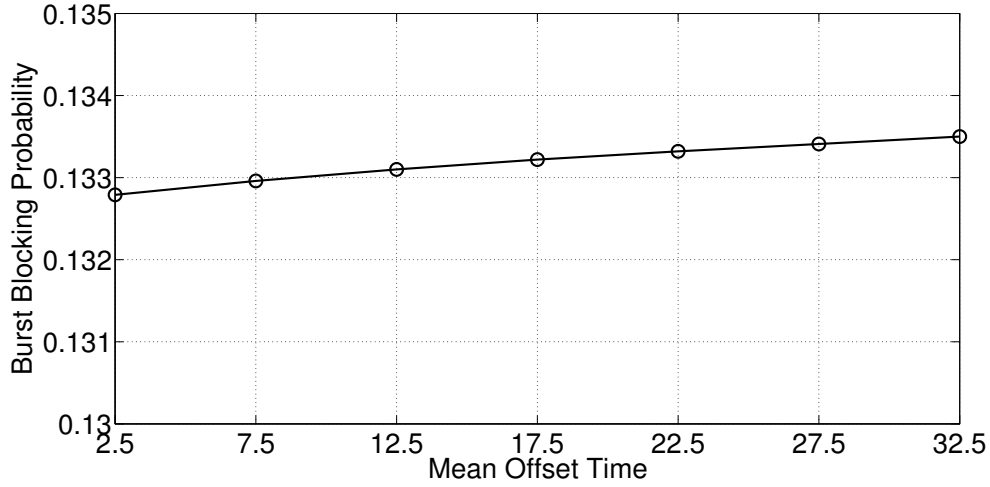


Figure 4.8: The effect of shifting the offset time distribution on the blocking probability.

lengths are exponentially distributed with mean 1 and the offset times are uniformly distributed on intervals of width 5. We vary the mean of the offset times between 2.5 and 32.5 with increments of 5. The blocking probabilities, which are plotted in Figure 4.8, barely change, so we conclude that shifting the offset time distribution without changing its shape has only a marginal effect on the blocking probability.

After observing that shifting has a marginal effect, we investigate the effect of the offset time variation. We use the same arrival process and burst length distribution as the previous example. The offset times are uniformly distributed in the interval $[10 - u/2, 10 + u/2]$, so the mean offset time is fixed at 10, but the variance increases with the parameter u . In Figure 4.9, the blocking probability is plotted against u , showing that increased variation in offset time hurts the performance in terms of blocking probability, i.e. deterministic offset times are preferable over random ones. This result was also obtained in [39], but through simulation.

Lastly, we give a two-class example. The arrivals of both classes are Poisson and the overall rate is varied between 0.2 and 1 with increments of 0.2. Three scenarios in which class 1, the low priority class, brings 25%, 50% and 75% of the overall traffic are solved. Burst lengths are exponentially distributed with mean 1 for both classes. In this example, we compare two cases.

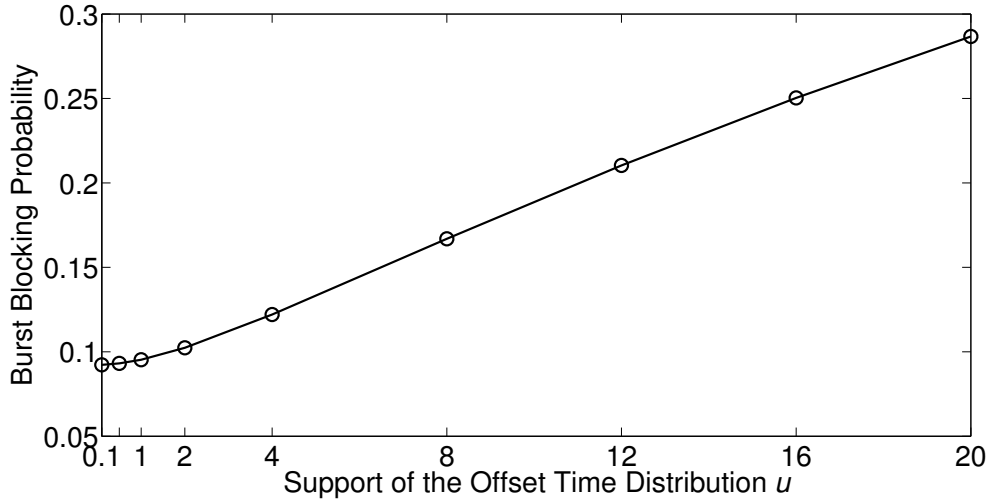


Figure 4.9: The effect of the offset time variation.

In the first one, the offset times of the two classes are taken to be deterministic, and equal to 0 and 5 for classes 1 and 2 respectively. In the second case, we replace the offset time distribution of class 2 with an exponential distribution with mean 5. We had showed that keeping the mean constant, offset time distributions with higher variances lead to increased blocking probabilities. Therefore, an increase in the overall blocking probability is to be expected, which is seen in Figure 4.10. However, with this example, we intend to investigate the effect of the offset time distribution on the quality of service (QoS) with respect to burst blocking in terms of class separation. For this purpose, we plotted the ratio of the blocking probability of the lower priority class to the blocking probability of the higher priority class in Figure 4.11. We see that not only the exponentially distributed offset time for the higher priority class increases overall blocking probability, it also reduces the separation between the classes. Therefore, we conclude that in addition to achieving lower blocking probabilities, deterministic offset times also lead to better class separation when QoS is sought.

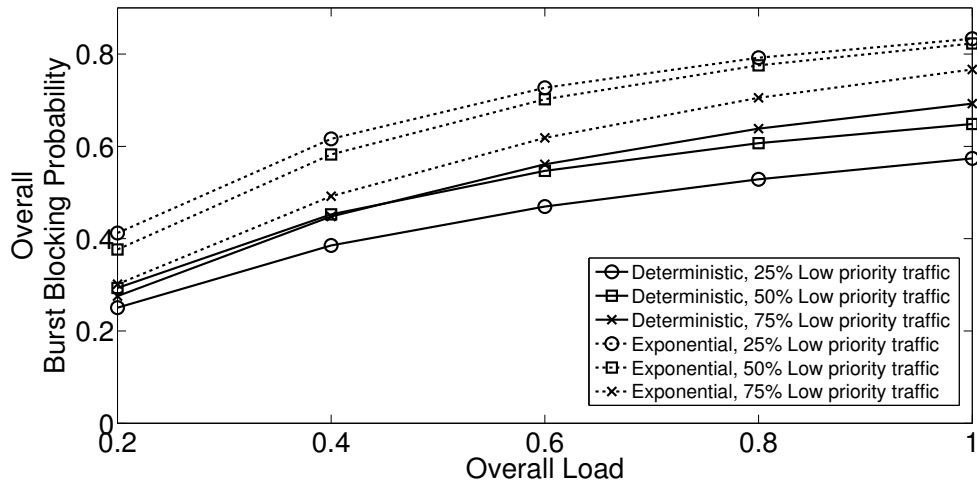


Figure 4.10: Overall blocking probability with two classes.

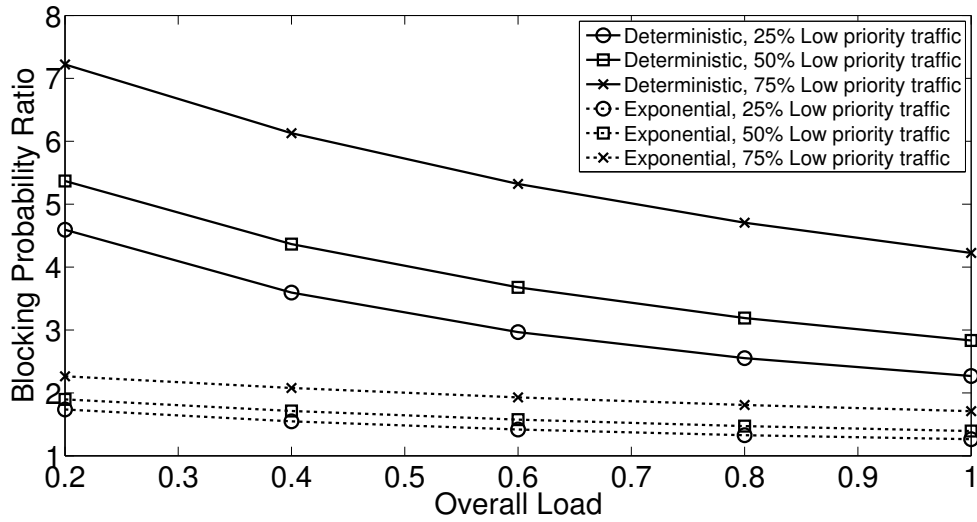


Figure 4.11: Ratios of the blocking probabilities of the two classes.

4.4 Stochastic Model for Horizon-based Reservation in the Presence of Fiber Delay Lines

In this section, we will present the stochastic model of the horizon-based reservation scheme in the presence of FDLs. We will base this model on the model presented in section 4.2 with appropriate modifications that reflects the behavior due to the existence of FDLs.

First of all, in this section, we assume that the distribution of the offset times is composed of only impulses. This is justified as we have established in the previous section that deterministic offset times lead to better performance in terms of blocking compared to random offset times. Therefore, it is natural for the optical ingress nodes to pick deterministic values for the offset times of the bursts they form. On the other hand, BCPs arriving at a core optical node may have different offset times depending on the paths they have travelled and the nodes they originate. Therefore, we used a distribution that is composed of a number of impulses to model the offset times. Even so, some of the numerical examples we present will have purely deterministic offsets resulting in a single impulse.

We also note that incorporating general continuous distributions for offset times into the model is not difficult. The only modification necessary would be adding states that model the continuous distribution of the offset times, and deriving the rates of transition out of these states by taking into account the FDLs.

Upon an arrival of a BCP, the corresponding burst is accepted if one of the following is true:

- (i) The offset time of the burst is greater than the horizon. In this case, no FDL is used.
- (ii) The offset time of the burst is less than the horizon, but there exists an FDL in the system that has an amount of delay which, when added to the offset time, exceeds the horizon. If this is the case, we assume that the FDL with the least amount of delay is used among those that satisfy the condition.

Let Δ_i denote the delay FDL i , $1 \leq i \leq N$, produces, where N is the total number of FDLs. Remember that if degenerate buffering is used, $\Delta_i = i\Delta$, where Δ is the granularity parameter. Assume that the FDLs are enumerated such that $\Delta_i < \Delta_j$ when $i < j$, $1 \leq i, j \leq N$. Then, a burst with offset time T_O whose BCP arrives when the horizon is equal to x

- is accepted without being fed into any FDLs if $T_O \geq x$,
- is accepted and is fed into FDL i if $T_O + \Delta_i \geq x > T_O + \Delta_{i-1}$,
- is blocked if $x > T_O + \Delta_N$.

Let the offset time distribution be

$$\theta(x) = \sum_{j=1}^m p_j \delta(x - T_j).$$

For each value T_j that the offset time can take, the burst is accepted if $x < T_j + \Delta_N$. Remember that we had defined states that take the horizon value to the offset time value. Similarly, in the presence of FDLs, we need to define states that take the horizon value to the values $T_j + \Delta_i$ for $1 \leq i \leq N$. Note that the horizon should be taken to $T_j + \Delta_i$ whenever the horizon is between $T_j + \Delta_{i-1}$ and $T_j + \Delta_i$. Therefore, a single state is sufficient to update the horizon to $T_j + \Delta_i$, $1 \leq i \leq N$.

Let s_j , $1 \leq j \leq m$, denote the state that takes the horizon value to $T_j + \Delta_i$, $1 \leq i \leq N$. When the background process is in state s_j , it cannot change state unless the horizon takes one of the values $T_j + \Delta_i$, $1 \leq i \leq N$. Therefore, the rate out of state s_{ij} is 0 everywhere except when $x = T_j + \Delta_i$, and the drift in this state is +1. Consequently, regime boundaries at points $T_j + \Delta_i$, $1 \leq j \leq m$, $1 \leq i \leq N$ should be defined.

When the background process is in the subset of states that constitute the MAP, it leaves that state with a rate given by the corresponding entry in the matrix D_1 , provided that the burst is accepted. The state it transits into is one of the states that updates the horizon with the offset time, which were described in the paragraph above. The offset time is going to be one of the values T_j , $1 \leq j \leq m$. Therefore,

the next state, s_j , is determined by the offset time. Note that given state transition occurs, the next state is s_j with probability p_j .

In addition to these two subset of states, we also have the states that add the burst length to the horizon. As before, these consist of the states that describe the PH-type distribution with parameters α and T . Moreover, we will have ℓ replicas of the PH-type states, and the offset states, s_j , as before to remember the MAP state to return to after the burst length is added to the horizon.

Combining all these, the system can be modeled with an MRMFQ, and the infinitesimal generator of the background process can be written as

$$Q^{(k)} = \begin{bmatrix} D_0^{(k)} & 0 & \tilde{p}^{(k)} \otimes D_1 \\ I_\ell \otimes T^0 & I_\ell \otimes T & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad 1 \leq k \leq K,$$

$$\tilde{Q}^{(k)} = \begin{bmatrix} D_0^{(k)} & 0 & \tilde{p}^{(k)} \otimes D_1 \\ I_\ell \otimes T^0 & I_\ell \otimes T & 0 \\ 0 & \tilde{J}_2^{(k)} & \tilde{J}_3^{(k)} \end{bmatrix}, \quad 0 \leq k \leq K,$$

where

$$p = [p_1 \quad \cdots \quad p_m],$$

$$\tilde{p}_j^{(k)} = p_j \mathbf{1}_{\{T^{(k)} \leq T_j + \Delta_N\}},$$

$$\tilde{p}^{(k)} = [\tilde{p}_1^{(k)} \quad \cdots \quad \tilde{p}_m^{(k)}],$$

$$D_0^{(k)} = D_0 + D_1 (p - \tilde{p}^{(k)}) \mathbf{1},$$

$$\tilde{J}_2^{(k)} = \begin{bmatrix} \mathbf{1}_{\{T^{(k)} = T_1 + \Delta_i \text{ for some } i\}} \\ \vdots \\ \mathbf{1}_{\{T^{(k)} = T_m + \Delta_i \text{ for some } i\}} \end{bmatrix} \otimes (I_\ell \otimes \alpha),$$

$$\tilde{J}_3^{(k)} = \text{diag} \begin{bmatrix} \mathbf{1}_{\{T^{(k)} = T_1 + \Delta_i \text{ for some } i\}} \\ \vdots \\ \mathbf{1}_{\{T^{(k)} = T_m + \Delta_i \text{ for some } i\}} \end{bmatrix} \otimes -I_\ell,$$

and $\text{diag}[\cdot]$ denotes a diagonal matrix whose diagonal consists of the entries of its input. Notice that the total number of states of the background process is $(1 + h + m)\ell$. On the other hand, the number of regimes, K , is the number of unique elements in

the multiset $\{T|T = T_j + \Delta_i, 1 \leq i \leq N, 1 \leq j \leq m\}$. Also, the drift matrices are given by

$$\begin{aligned} R^{(k)} &= \begin{bmatrix} -I_\ell & & \\ & I_{(h+m)\ell} & \\ & & \end{bmatrix}, & 1 \leq k \leq K-1, \\ R^{(K)} &= \begin{bmatrix} -I_\ell & & \\ & I_{h\ell} & \\ & & -I_{m\ell} \end{bmatrix}, \\ \tilde{R}^{(k)} &= \begin{bmatrix} -I_\ell & & \\ & I_{h\ell} & \\ & & 0_{m\ell} \end{bmatrix}, & 0 \leq k \leq K. \end{aligned}$$

The reason for $R^{(K)}$ being different from $R^{(k)}$, $1 \leq k \leq K-1$, is to keep the background process from entering regime K when the background process is in one of the states that update the horizon, as the horizon cannot be increasing beyond $T^{(K-1)} = T_m + \Delta_N$ due to offset time and FDLs.

Note that when solving the distribution of the transformed horizon, $f(x)$, boundary conditions should be examined carefully. In contrast to the models presented upto this point, there will be probability masses at the regime boundaries due to the states that have zero drifts at the boundaries. Also, the boundary condition (2.20) will contribute. As a result, the corresponding linear system of equations result in a block-banded matrix that does not readily lend itself to the block-tridiagonal LU decomposition algorithm with blocks that have sizes equal to the number of the states. As mentioned in section 2.3.2, extra care must be taken in order to fully cover the matrix when partitioning for the block-tridiagonal LU decomposition algorithm.

Also notice that the state ordering in this model differs from the models we have presented so far in the thesis. This means that the distribution of the real horizon, $g(x)$, can be found from $f(x)$ via

$$g(x) = \frac{\begin{bmatrix} f_1(x) & \cdots & f_\ell(x) \end{bmatrix}}{\begin{bmatrix} F_1(x) & \cdots & F_\ell(x) \end{bmatrix} \mathbf{1}}.$$

The model for the two-class system can be found in a similar way. For classes A

and B, we denote the MAP parameters with (D_{0A}, D_{1A}) and (D_{0B}, D_{1B}) with sizes ℓ_A and ℓ_B , the PH-type distribution parameters with (α_A, T_A) and (α_B, T_B) with sizes h_A and h_B , and the offset time distributions with

$$\theta_A(x) = \sum_{j=1}^{m_A} p_j \delta(x - T_{jA}),$$

$$\theta_B(x) = \sum_{j=1}^{m_B} p_j \delta(x - T_{jB}),$$

respectively. Also, let the set of FDLs available to class A and B be $(\Delta_1, \dots, \Delta_{N_A})$ and $(\Delta_1, \dots, \Delta_{N_B})$ respectively.

Then, by forming the composite MAP from the MAP parameters of classes A and B, we can write the infinitesimal generator of the background process as

$$Q^{(k)} = \begin{bmatrix} D_0^{(k)} & 0 & 0 & \tilde{p}_A^{(k)} \otimes (I_{\ell_B} \otimes D_{1A}) & p_B^{(k)} \otimes (D_{1B} \otimes I_{\ell_A}) \\ I_{\ell_A \ell_B} \otimes T_A^0 & I_{\ell_A \ell_B} \otimes T_A & 0 & 0 & 0 \\ I_{\ell_A \ell_B} \otimes T_B^0 & 0 & I_{\ell_A \ell_B} \otimes T_B & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$1 \leq k \leq K,$$

$$\tilde{Q}^{(k)} = \begin{bmatrix} D_0^{(k)} & 0 & 0 & \tilde{p}_A^{(k)} \otimes (I_{\ell_B} \otimes D_{1A}) & p_B^{(k)} \otimes (D_{1B} \otimes I_{\ell_A}) \\ I_{\ell_A \ell_B} \otimes T_A^0 & I_{\ell_A \ell_B} \otimes T_A & 0 & 0 & 0 \\ I_{\ell_A \ell_B} \otimes T_B^0 & 0 & I_{\ell_A \ell_B} \otimes T_B & 0 & 0 \\ 0 & \tilde{f}_{A2}^{(k)} & 0 & \tilde{f}_{A4}^{(k)} & 0 \\ 0 & 0 & \tilde{f}_{B3}^{(k)} & 0 & \tilde{f}_{B5}^{(k)} \end{bmatrix},$$

$$0 \leq k \leq K,$$

where

$$p_A = \begin{bmatrix} p_{1A} & \cdots & p_{m_A} \end{bmatrix},$$

$$p_B = \begin{bmatrix} p_{1B} & \cdots & p_{m_B} \end{bmatrix},$$

$$\tilde{p}_{j,A}^{(k)} = p_{j,A} \mathbf{1}_{\{T^{(k)} \leq T_{jA} + \Delta_{N_A}\}},$$

$$\tilde{p}_A^{(k)} = \begin{bmatrix} \tilde{p}_{1,A}^{(k)} & \cdots & \tilde{p}_{m_A,A}^{(k)} \end{bmatrix},$$

$$\tilde{p}_{j,B}^{(k)} = p_{j,B} \mathbf{1}_{\{T^{(k)} \leq T_{jB} + \Delta_{N_B}\}},$$

$$\begin{aligned}
\tilde{p}_B^{(k)} &= \begin{bmatrix} \tilde{p}_{1,B}^{(k)} & \cdots & \tilde{p}_{m_B,B}^{(k)} \end{bmatrix}, \\
D_0^{(k)} &= I_{\ell_B} \otimes \left[D_{0A} + D_{1A} \left(p_A - \tilde{p}_A^{(k)} \right) \mathbf{1} \right] + \left[D_{0B} + D_{1B} \left(p_B - \tilde{p}_B^{(k)} \right) \mathbf{1} \right] \otimes I_{\ell_A}, \\
\tilde{J}_{A2}^{(k)} &= \begin{bmatrix} \mathbf{1}_{\{T^{(k)}=T_1+\Delta_i \text{ for some } i\}} \\ \vdots \\ \mathbf{1}_{\{T^{(k)}=T_{m_A}+\Delta_i \text{ for some } i\}} \end{bmatrix} \otimes (I_{\ell_A \ell_B} \otimes \alpha_A), \\
\tilde{J}_{A4}^{(k)} &= \text{diag} \begin{bmatrix} \mathbf{1}_{\{T^{(k)}=T_1+\Delta_i \text{ for some } i\}} \\ \vdots \\ \mathbf{1}_{\{T^{(k)}=T_{m_A}+\Delta_i \text{ for some } i\}} \end{bmatrix} \otimes -I_{\ell_A \ell_B}, \\
\tilde{J}_{B3}^{(k)} &= \begin{bmatrix} \mathbf{1}_{\{T^{(k)}=T_1+\Delta_i \text{ for some } i\}} \\ \vdots \\ \mathbf{1}_{\{T^{(k)}=T_{m_B}+\Delta_i \text{ for some } i\}} \end{bmatrix} \otimes (I_{\ell_A \ell_B} \otimes \alpha_B), \\
\tilde{J}_{B5}^{(k)} &= \text{diag} \begin{bmatrix} \mathbf{1}_{\{T^{(k)}=T_1+\Delta_i \text{ for some } i\}} \\ \vdots \\ \mathbf{1}_{\{T^{(k)}=T_{m_B}+\Delta_i \text{ for some } i\}} \end{bmatrix} \otimes -I_{\ell_A \ell_B}.
\end{aligned}$$

The total number of states of the background process is $(1+h_A+h_B+m_A+m_B)\ell_A\ell_B$.

The drift matrices are

$$\begin{aligned}
R^{(k)} &= \begin{bmatrix} -I_{\ell_A \ell_B} & & \\ & I_{(h_A+h_B+m_A+m_B)\ell_A \ell_B} & \\ & & \end{bmatrix}, & 1 \leq k \leq K-1, \\
R^{(K)} &= \begin{bmatrix} -I_{\ell_A \ell_B} & & \\ & I_{(h_A+h_B)\ell_A \ell_B} & \\ & & -I_{(m_A+m_B)\ell_A \ell_B} \end{bmatrix}, \\
\tilde{R}^{(k)} &= \begin{bmatrix} -I_{\ell} & & \\ & I_{(h_A+h_B)\ell_A \ell_B} & \\ & & 0_{(m_A+m_B)\ell_A \ell_B} \end{bmatrix}, & 0 \leq k \leq K.
\end{aligned}$$

Finally, similar to (4.4) and (4.9), the blocking probability in the single class case is given by

$$p_B = \frac{\int_0^\infty g(x) D_1 \mathbf{1} \Theta(x - \Delta_N) dx}{c^{(0)} D_1 \mathbf{1} + \int_0^\infty g(x) D_1 \mathbf{1} dx},$$

and in two classes case by

$$p_{B(A)} = \frac{\int_0^\infty g_A(x) D_{1A} \mathbf{1} \Theta_A(x - \Delta_{N_A}) dx}{c_A^{(0)} D_{1A} \mathbf{1} + \int_0^\infty g_A(x) D_{1A} \mathbf{1} dx},$$

$$p_{B(B)} = \frac{\int_0^\infty g_B(x) D_{1B} \mathbf{1} \Theta_B(x - \Delta_{N_B}) dx}{c_B^{(0)} D_{1B} \mathbf{1} + \int_0^\infty g_B(x) D_{1B} \mathbf{1} dx},$$

$$p_B = \frac{\int_0^\infty g_A(x) D_{1A} \mathbf{1} \Theta_A(x - \Delta_{N_A}) dx + \int_0^\infty g_B(x) D_{1B} \mathbf{1} \Theta_B(x - \Delta_{N_B}) dx}{c_A^{(0)} D_{1A} \mathbf{1} + \int_0^\infty g_A(x) D_{1A} \mathbf{1} dx + c_B^{(0)} D_{1B} \mathbf{1} + \int_0^\infty g_B(x) D_{1B} \mathbf{1} dx}.$$

Note that since we assume discrete offset times in this section, the distributions $\Theta(x)$, $\Theta_A(x)$, and $\Theta_B(x)$ are piecewise constant, and the integrals appearing in the loss expressions above can be computed easily without resorting to numerical integration.

4.5 Numerical Examples for Horizon-based Reservation in the Presence of Fiber Delay Lines

We start with demonstrating the benefit of FDLs in terms of blocking probability. We have a single class with Poisson arrivals of rate 0.8, exponentially distributed burst lengths with mean 1, and no offsets. The number of FDLs are varied from 0 upto 20. Under this set of parameters, a granularity parameter of 0.5 is a good choice as can be inferred from [91] (see Figure 6(a) on page 538). The burst blocking probability against the number of FDLs used is plotted in Figure 4.12. The benefit of using FDLs is obvious.

We also show the structure and the sparsity of the matrix H , as defined in (2.24), in Figure 4.13 for this example using 8 FDLs. This means there are 9 regimes, and the number of states is 3. Notice that this matrix can not be partitioned with square blocks of size 3 as some elements would be left out.

With the rest of the examples, we focus on the two class model. First, we demonstrate the accuracy of our model with an example with the following parameters:

$$D_{0A} = \begin{bmatrix} -0.8 & 0.8 \\ 0 & -0.8 \end{bmatrix}, \quad D_{1A} = \begin{bmatrix} 0 & 0 \\ 0.8 & 0 \end{bmatrix},$$

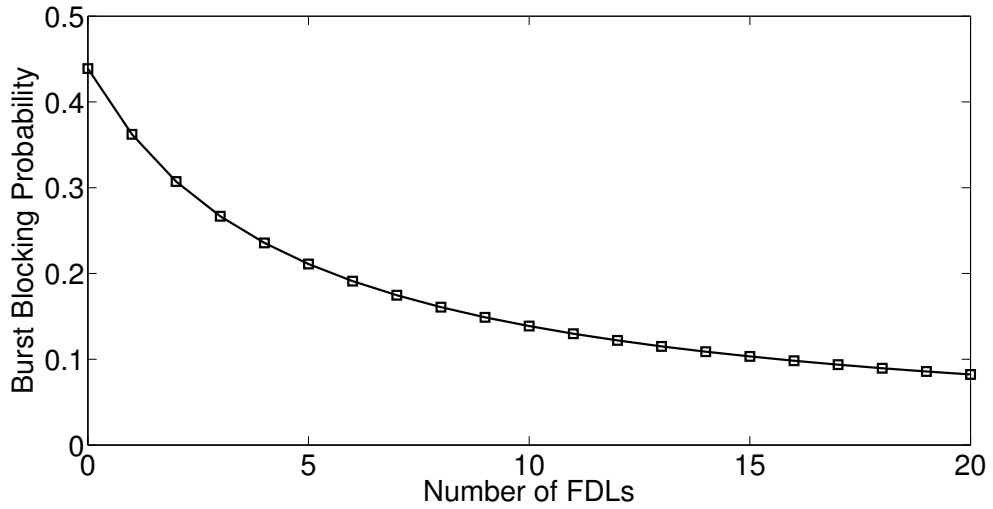


Figure 4.12: The burst blocking probability with varying number of FDLs demonstrating the merits of FDLs.

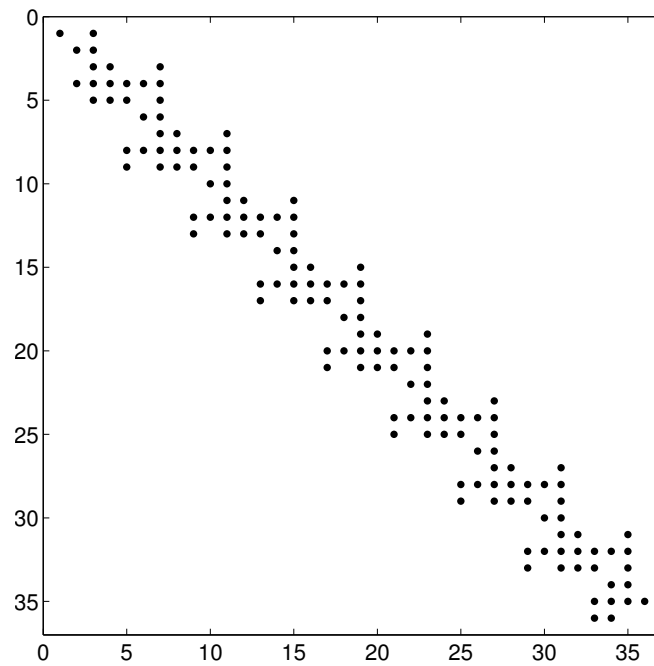


Figure 4.13: Structure of the 36×36 matrix H with 8 FDLs. There are 3 states, but the matrix H can not be partitioned with square blocks of size 3. Compare with Figure 3.3 on page 71.

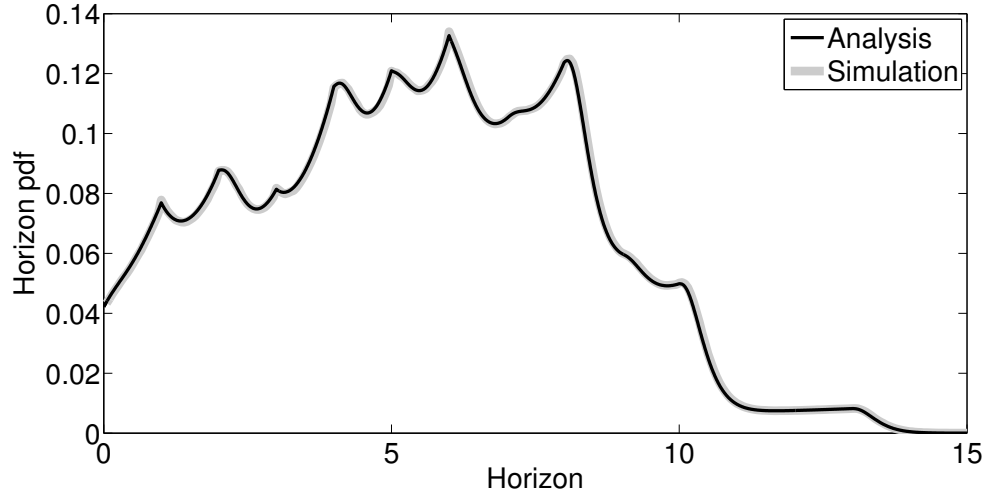


Figure 4.14: The pdf of the horizon for a scenario with a variety of general cases.

$$D_{0B} = \begin{bmatrix} -0.5 & 0 & 0 \\ 0 & -0.2 & 0 \\ 0 & 0 & -0.8 \end{bmatrix},$$

$$\alpha_A = \begin{bmatrix} 0.8 & 0.2 \end{bmatrix},$$

$$\alpha_B = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix},$$

$$\theta_A(x) = \delta(x-1),$$

$$\theta_B(x) = 0.5\delta(x) + 0.3\delta(x-2) + 0.2\delta(x-5),$$

$$D_{1B} = \begin{bmatrix} 0 & 0.25 & 0.25 \\ 0.1 & 0 & 0.1 \\ 0.4 & 0.4 & 0 \end{bmatrix},$$

$$T_A = \begin{bmatrix} -1.5 & 0 \\ 0 & -2 \end{bmatrix},$$

$$T_B = \begin{bmatrix} -6 & 6 & 0 \\ 0 & -6 & 6 \\ 0 & 0 & -6 \end{bmatrix},$$

$$\Delta_A = 0.5, \quad N_A = 20,$$

$$\Delta_B = 0.4, \quad N_B = 10.$$

This set of parameters include a variety of general cases, including even completely different sets of FDLs for the two classes. Although this does not make a lot of sense for a physical system, we preferred such a setting to demonstrate the capability of the model. The pdf of the horizon is plotted in Figure 4.14, demonstrating perfect match with the simulation result.

With the third example, we present three scenarios. In all three, both classes have Poisson arrivals with rate 0.4 and exponentially distributed burst lengths with mean 1. There are 20 FDLs with granularity 0.5.

	Class A	Class B	Overall
Scenario i	0.01057	0.23753	0.12405
Simulation	0.01110	0.24042	0.12568
Scenario ii	0.01001	0.22489	0.11745
Simulation	0.00994	0.22669	0.11852
Scenario iii	0.00197	0.53871	0.27034
Simulation	0.00192	0.53918	0.27074

Table 4.1: Per class and overall blocking probabilities for the three scenarios.

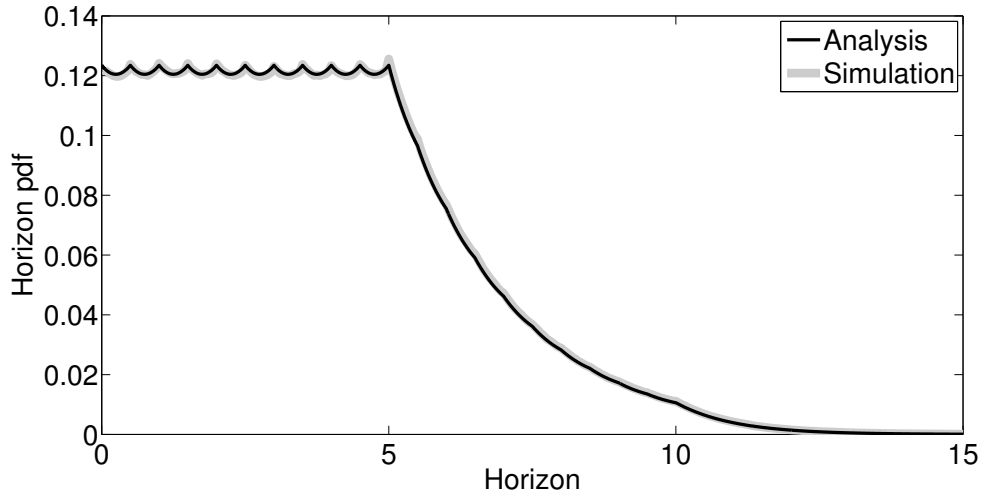


Figure 4.15: The pdf of the horizon for scenario i.

Scenario i. Both classes have 0 offsets. Class A can access all the FDLs, whereas class B access only the first 10, producing delays from 0.5 upto 5. The horizon pdf for this scenario is given in Figure 4.15.

Scenario ii. Both classes have access to all FDLs. Class A has a deterministic offset of 5, whereas class B has no offset. The horizon pdf for this scenario is given in Figure 4.16.

Scenario iii. Both classes have access to all FDLs. Class A has a deterministic offset of 10, whereas class B has no offset. The horizon pdf for this scenario is given in Figure 4.17.

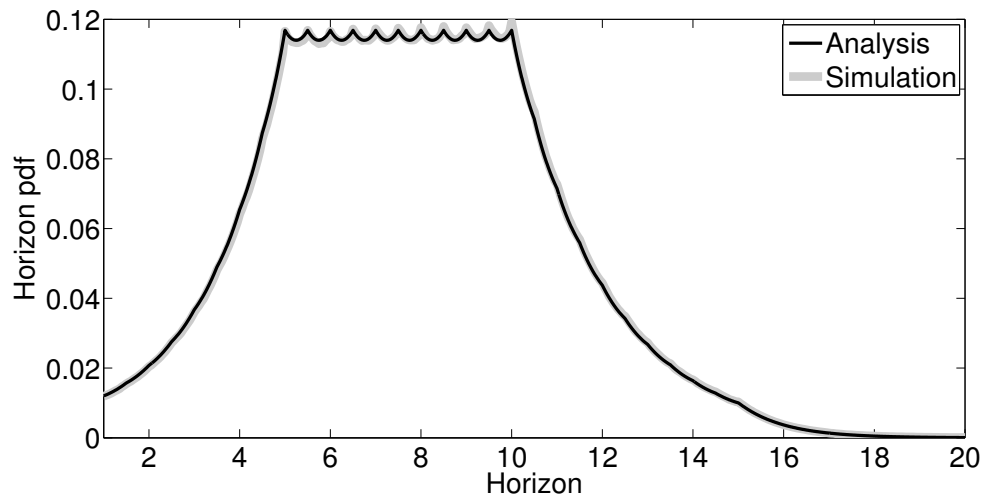


Figure 4.16: The pdf of the horizon for scenario ii.

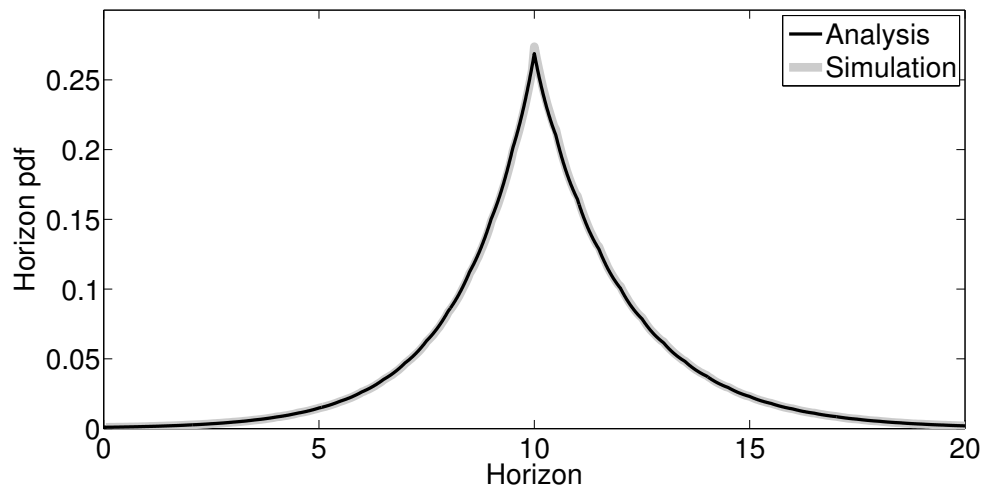


Figure 4.17: The pdf of the horizon for scenario iii.

The Figures 4.15, 4.16 and 4.17 along with Table 4.1 demonstrate that our model perfectly captures the behavior of the system. As demonstrated with these three scenarios, an analysis of QoS differentiation in terms of burst blocking probability and provisioning is possible using our model.

As the last example, we investigate QoS differentiation in terms of burst blocking probability using FDL access limitation. Arrivals are Poisson, and burst lengths are exponentially distributed with mean 1. There are 20 FDLs with granularity 0.5, and both classes have no offsets. Class A (high priority) has access to all FDLs, while the number of FDLs class B (low priority) is varied between 0 and 20. The separation of classes as well as overall blocking performance is investigated for two different load values under three different traffic share scenarios:

- i. 25% high priority traffic - 75% high priority traffic,
- ii. 50% high priority traffic - 50% high priority traffic,
- iii. 75% high priority traffic - 25% high priority traffic.

The per class loss and overall probabilities for each scenario is given in Figure 4.18. It can be inferred from the plots that class separation is achievable using FDL access limitation as long as the number of FDLs that the low priority class have access to does not get too close to the full set of FDLs. Another observation is that when the low priority class is dominant, if the number of FDLs the low priority class can access is selected too small, unnecessary performance degradation can occur as seen in Figure 4.18(a) and 4.18(d). So, we conclude that moderate levels of FDL limitation on the the low priority class should be the preferred solution.

4.6 Conclusion

We model the horizon-based delayed reservation scheme on a single-channel OBS system, and give an exact solution to the model for single user and two-user scenarios. We assume no specific circumstances such as low load. In the most general

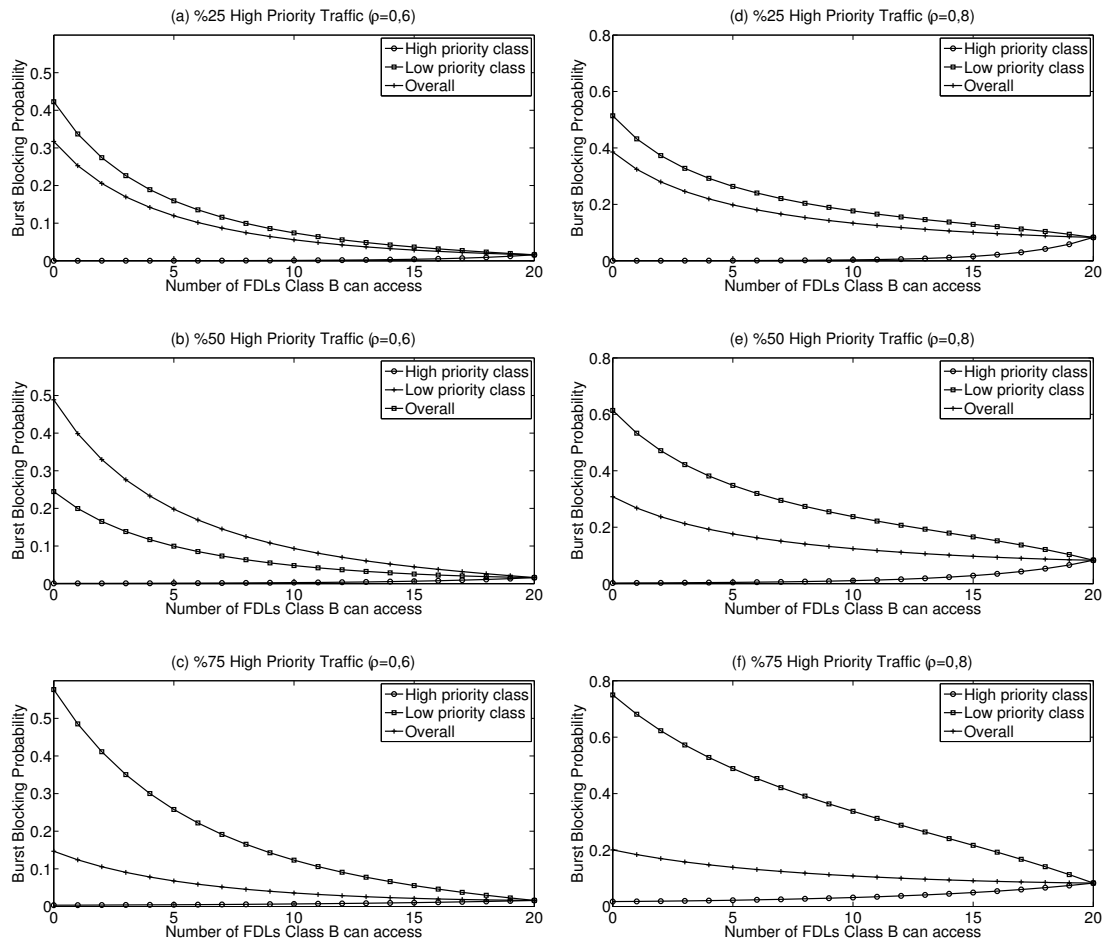


Figure 4.18: QoS differentiation in terms of burst blocking probability using FDL access limitation.

case, the bursts arrive according to a MAP, burst lengths are PH-type distributed and the offset time is generally distributed.

We base our model on the continuous feedback Markov fluid queue (CFMFQ) theory. We describe how the system maps to a CFMFQ by formulating the matrices that describe the CFMFQ. Then, by discretizing the CFMFQ into a MRMFQ, we obtain a much simpler system that can be solved using existing methods. We provide numerical examples validating our model, and also investigate the effect of the offset time distribution on the burst blocking probability.

In addition, we model the same system in the presence of FDLs. The numerical examples we present demonstrate that our model perfectly agrees with simulations in a wide variety of scenarios. Using our model, we investigate QoS differentiation by means of limiting the number of FDLs one of the classes can access. The results show that FDL access limitation is a promising method that can be employed for the purposes of QoS differentiation.

As our foremost aim with the numerical examples in this chapter was the demonstration of the capability of the stochastic model we based on CFMFQs, we did not pursue a very detailed analysis to solve actual practical problems related to OBS networks. The study by Kankaya and Akar [91] investigates a number of problems in this setting such as the value of the optimal granularity parameter in terms of the number of FDLs or the total fiber length. Their results reveal that finding answers to such questions is not a trivial task, and requires exhaustive experimentation.

We have significantly improved over the model of Kankaya and Akar [91] as well as their study on offset-based differentiation [43] by

- (i) allowing impulsive distributions for offset times instead of purely deterministic ones,
- (ii) modeling multiple traffic classes, and
- (iii) combining the analysis of offset-based differentiation and FDLs.

Building on this framework, more detailed provisioning analyses may be carried out, which we leave as future work. In addition, other aspects of QoS such as delay performance can be taken into account to provide a more encompassing understanding of the horizon-based single channel systems. Another research direction is the investigation of multi-channel systems.

Chapter 5

Achieving Air-time Fairness in Multi-rate IEEE802.11 WLANs

5.1 Introduction

The IEEE 802.11 suite of protocols is the most widely used set for Wireless Local Area Networks (WLAN). On the Medium Access Control (MAC) side, IEEE 802.11 employs a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) MAC protocol with binary exponential back-off, known as Distributed Coordination Function (DCF) [44]. DCF defines a mandatory basic access mechanism and an optional Request-To-Send/Clear-To-Send (RTS/CTS) mechanism. The focus of this study is on the basic access mechanism, which is more frequently used in practice.

In DCF basic access mechanism, an 802.11 node with a frame to transmit listens to the channel first to detect an idle period of length at least equal to the Distributed Inter-Frame Space (DIFS). The node then sets its back-off timer value to an integer that is uniformly chosen in the interval $[0, CW - 1]$, where CW is set to CW_{min} (minimum contention window size) at the first transmission attempt. The back-off timer is then decremented at each slot as long as the channel is idle whereas it is stopped when a transmission is detected on the channel. Re-activation of the timer

occurs after the node senses the channel idle after this transmission for at least another DIFS. The back-off timer hitting zero triggers the frame's first transmission. Once the destination host successfully receives the frame, it transmits an acknowledgment frame (ACK) after a short inter-frame space (SIFS) time. If the transmitting node does not receive an ACK within a specified ACK timeout for the transmitted frame, a collision is said to have taken place. Upon each collision, CW is doubled until a maximum contention window size CW_{max} value is reached and the above back-off mechanism is repeatedly applied at each unsuccessful transmission.

IEEE 802.11 standards support multiple raw data rates and hence such networks are called multi-rate WLANs. Moreover, the 802.11 standards support link adaptation by which a host selects one of the available transmission rates at a given transmission opportunity based on channel conditions and/or application traffic type. Although link adaptation appears to be a powerful means to enhance throughput in multi-rate WLANs, its effective use in multi-user 802.11 WLANs has been shown to be limited due to the performance anomaly problem [52].

Consider a scenario of multiple hosts with a higher raw bit rate in addition to a single host with a lower bit rate as used in [52] with all frame sizes assumed to be the same. Since the CSMA/CA algorithm of DCF provides the same equal channel access probability to all hosts (throughput-fair), the throughput of high rate hosts will be the same as the slow host. Therefore, DCF penalizes fast hosts and instead favors the slow host. This artifact is known as the performance anomaly problem of 802.11 DCF which impedes a direct relationship between the raw data rate and the actual throughput in scenarios with multiple users with different data rates [52].

Time-based fairness is proposed in [53],[54],[55] as an alternative to throughput fairness to cope with the performance anomaly problem. With time-based fairness, each competing node receives an equal share of the wireless channel occupancy time, i.e., air-time. A system achieving time-based fairness is called air-time fair. When air-time fair mechanisms are employed, the throughput of an individual node becomes strictly proportional with its raw bit rate and therefore high rate nodes will no longer be dragged down by slower ones, which leads to significantly higher cumulative throughputs [54].

We propose a novel and relatively simple to implement method for achieving air-time fairness in IEEE 802.11 WLANs. In this method, a competing node runs multiple instances of the standard back-off algorithm at each node. Equivalently, a competing node behaves as a collection of multiple virtual nodes where each virtual node has its own DCF instant. When the back-off timer of a virtual node hits zero, then its controlling physical node decides to transmit the awaiting frame on behalf of the virtual node. The multiplicity of back-off algorithm instances acts as an instrument to deliver air-time fairness.

Consider a competing node i that runs N_i instances of the basic back-off algorithm. Assume that each node i requires an average air-time $E[A_i]$ at each of its transmission opportunities. Let $A_{max} \geq E[A_j], \forall j$ be a value known to all nodes. We propose that the parameter N_i is set to $N_i = \frac{A_{max}}{E[A_i]}$ which can be done in a distributed manner once all nodes have the value of A_{max} .

Note that the parameter N_i need not be an integer. In that case, we propose a novel mechanism that appropriately switches between $N_{i-} = \lfloor N_i \rfloor$ and $N_{i+} = \lceil N_i \rceil$ back-off algorithms. We also provide a Markov chain-based analytical model to validate this switching mechanism actually delivers air-time fairness. Moreover, through extensive simulations, we show that the method achieves air-time fairness at the expense of an acceptable reduction in channel utilization. The proposed method can also be used in conjunction with frame aggregation to substantially mitigate this utilization reduction.

5.2 Existing Methods for Air-time Fairness

The first approach is based on the use of contention window parameter CW_{min} as an instrument to achieve air-time fairness. It has been shown that under certain assumptions, the nodal throughput is inversely proportional with the CW_{min} value of the node [56, 57]. Therefore, it is natural to conclude that air-time fairness can be achieved if the initial contention window size CW_{min} is chosen to be inversely proportional with the raw bit rate.

The main advantage of the CW_{min} -approach to deliver air-time fairness is in its simplicity of implementation and the preservation of the DCF mechanism. However, several drawbacks of this approach within the scope of air-time based fairness can be listed:

- The relationship between CW_{min} and the nodal throughput is valid only for regimes where the collision probabilities are small. Actually, the relationship between CW_{min} and the nodal throughput is sensitive to system parameters such as number of nodes, choice of initial congestion windows, etc. For example, a simulation study by Tinnirello et al. [58] demonstrates that the throughput ratio between two classes of nodes with a fixed CW_{min} ratio is slightly sensitive to the number of nodes in each class. Similar results also appear in [61].
- When the ratio between the lowest and highest raw bit rates is relatively large, the nodes with highest raw bit rates need to use large initial contention windows. This can lead to a considerable under-utilization of the channel [63].
- In actual implementations, CW_{min} needs to be a power of two [59]. Therefore, perfect air-time fairness between two nodes can not be achieved if the raw bit rate ratio is not a power of two.

A number of solutions have been proposed for tackling these problems. However, the proposals include tracking WLAN population [63] that is hard to achieve in a distributed way, and CW_{min} adaptation based on an estimation of the number of idle slots, which widely deviates from the standard.

Packet fragmentation is another approach to achieve air-time fairness. In this approach, nodes with high bit rates use a frame size equal to the maximum transmission unit (MTU) whereas slow nodes fragment their packets. The obvious drawback of the fragmentation-based approach is the increase in overhead. This disadvantage is even magnified when most nodes are slow. Implementation complexity can be mentioned as another drawback.

Another category of solutions is the frame aggregation approach, which is proposed in the IEEE 802.11e standard in which a transmission opportunity (TXOP), also referred to as the maximum channel occupation time, is broadcasted by the base station to each contending node. Consequently, nodes can aggregate their awaiting frames for transmission as long as the channel occupancy time does not exceed TXOP. Frame aggregation is also a crucial component of 802.11n due to the benefits it offers due to the significant reduction of overhead [48].

Frame aggregation can be used as a means of achieving air-time fairness and nodes with better channel conditions are allowed to send multiple frames at a transmission opportunity as opposed to low bit rate nodes that do not perform aggregation. Although TXOP can be used as an effective means of providing air-time fairness, the following drawbacks are identified:

- Frames are typically of variable size and further mechanisms including fragmentation are needed to transmit a number of frames within TXOP.
- Frame aggregation is generally used as a means of reducing overhead and thus enhancing cumulative throughput. If this method is used for air-time fairness, then slow nodes would not benefit from aggregation as much in case they dominate the user type.
- Let us assume all frames to be of the same length for the sake of simplicity. In the TXOP approach, to deliver air-time fairness, the TXOP may be defined to be the time required for the slowest node to transmit a single frame. Let us now assume a 802.11b WLAN occupied by two nodes with 11 Mbps raw bit rates. In this case, when a node has channel access, it will transmit 11 back-to-back frames. Clearly, such a scheme leads to unfairness between these two nodes in the short term.
- Frame aggregation may lead to relatively poor delay performance as shown in the paper by Lin et al. [62].

Other efforts for air-time fairness involve either cross-layer techniques, or relatively complex methods that involve a mixture of the mentioned schemes. In

comparison, our proposed method is simple, fully distributed and avoids the listed drawbacks.

5.3 Proposed MAC and Its Analytical Modeling

We assume a WLAN with K saturated users (nodes, stations) which always have frames to send. We also assume that link adaptation at each node $i, 1 \leq i \leq K$, is done in a way that frame error probabilities can be neglected. The case of non-zero frame error probabilities are left for future research. In order to include the possibility of frame aggregation, we define a burst to be a number of back-to-back frames to be transmitted at a given transmission opportunity. A burst may correspond to a single frame if frame aggregation is not allowed.

Figure 5.1 illustrates a snapshot of the air-time utilization of the WLAN channel in time which consists of alternating idle and busy periods. When the user i transmits successfully, i.e., no collisions take place, the channel is said to be occupied for an air-time of A_i with mean $E[A_i]$. Actually, $A_i = B_i/R_i$ where B_i is the burst size and R_i is the raw bit rate, of user i . When a transmission occurs, this transmission belongs to user i with probability p_i . Transmissions are followed with idle periods whose duration is a random variable denoted by A_I with mean $E[A_I]$. In this generic model, there is flexibility in what constitutes an idle or busy period. If air-time fairness is sought only in terms of air-time required for the transmission of payloads, all header transmissions at the higher and MAC layers may be counted towards the idle period. The random variable A_I has a fixed part which is examined in detail in [52], but also has a varying component stemming from contention periods which may dominate with increasing number of users K . The modeling and analysis of the idle

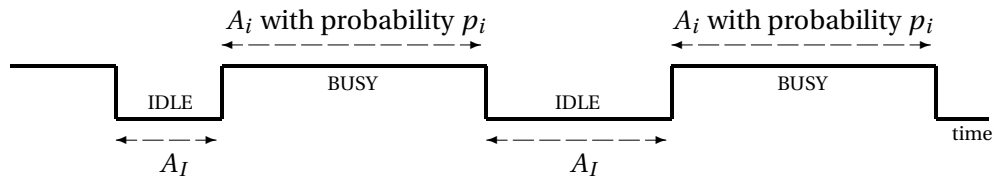


Figure 5.1: A snapshot of the air-time utilization of a random access WLAN.

period A_I has received a lot of attention in the literature [92],[52]. In this study, we will not seek for a stochastic model of A_I but rather focus on the air-time utilization values of individual users, which are crucial for air-time fairness. Note that users with low bit rates tend to occupy the channel longer at each transmission opportunity. A_i can also vary for the same user i in time due to varying frame lengths but we assume that $E[A_i]$ is known to user i .

Let U_i denote the air-time utilization of user i which is defined as the overall air-time consumed by successful transmissions of user i over a large time window divided by the window length. With this definition,

$$U_i = \frac{p_i E[A_i]}{E[A_I] + \sum_{j=1}^K p_j E[A_j]}. \quad (5.1)$$

The overall channel utilization U is then given by $U = \sum_{j=1}^K U_j$. If $U_i = U_j, i \neq j, \forall i, j$, then we say the system is air-time fair. The IEEE 802.11 DCF is known to provide equal channel access probabilities, i.e., $p_i^{(DCF)} = 1/K$, for all i . Denoting the air-time utilization of user i in DCF by $U_i^{(DCF)}$, we have

$$U_i^{(DCF)} = \frac{E[A_i]}{KE[A_I] + \sum_{j=1}^K E[A_j]}. \quad (5.2)$$

It is therefore clear that air-time utilizations of users may be different with DCF due to either different raw bit rates or frame lengths. One possibility is to control $E[A_i]$ by frame aggregation in a way that $E[A_i] = E[A_j], i \neq j, \forall i, j$. Indeed, this is what is done with frame fragmentation/aggregation approaches. Instead, we propose controlling p_i by letting the user i run N_i instances of the DCF. Here, N_i is set to $\frac{A_{max}}{E[A_i]}$, where the quantity A_{max} is such that $A_{max} \geq E[A_j], \forall j$, and is known to each user.

In theory, any value that exceeds all the $E[A_j]$ values is acceptable for the quantity A_{max} . However, in the ideal case, A_{max} should be kept as low as possible to reduce the number of total virtual nodes in the system. This is because the virtual population determines the channel utilization and has an adverse effect on it. Therefore, the best value for A_{max} would be $A_{max} = \max_j E[A_j]$. However, this choice would require the dissemination of the $E[A_j]$ value of each node. Moreover, further communication would be necessary in the case of rate adaptation, or nodes becoming online or going offline. To avoid this complication, we propose that A_{max}

is set to the time to transmit the largest possible frame with the minimum data rate the protocol supports, which then lends itself to a distributed implementation.

With this choice of A_{max} , $N_i \geq 1$ for all i , but N_i need not be an integer for some user i . Let us first assume that N_i is an integer for all i , $1 \leq i \leq K$, in which case user i behaves as a collection of N_i virtual users each running its own DCF. We call this architecture Multiple DCF (MDCF). Consequently, the channel access probability for user i for MDCF denoted by $p_i^{(MDCF)}$ can be written as

$$p_i^{(MDCF)} = \frac{N_i}{\sum_{j=1}^K N_j} = \frac{1}{A_i \sum_{j=1}^K \frac{1}{A_j}}. \quad (5.3)$$

Finally, the air-time utilization of user i in MDCF, denoted by $U_i^{(MDCF)}$, is written as

$$U_i^{(MDCF)} = \frac{1}{K + \bar{A}_I \sum_{j=1}^K \frac{1}{A_j}}. \quad (5.4)$$

Since, the right hand side of (5.4) does not depend on i , U_i is independent of i and consequently, MDCF is air-time fair for integer N_i .

In the case of non-integer N values, one can attack this problem by scaling up A_{max} . As an example, assume $K = 2$, $N_1 = 1.4$ and $N_2 = 3.1$. One can achieve air-time fairness by running 14 (31) instances of the DCF for user 1 (2), but at the expense of lowered cumulative throughputs stemming from significant increases in the overall number of DCFs in the system and therefore $E[A_I]$. Instead, we propose an all-distributed approach still based on MDCF in the case of non-integer N_i . For the sake of simplicity, consider the particular case $K = 2$, N_2 is an integer but N_1 is not. Let $N_{1-} = \lfloor N_1 \rfloor$ and $N_{1+} = \lceil N_1 \rceil$. User 1 runs N_{1-} instances of the DCF for a geometrically distributed period of mean B_{1-} transmissions belonging to user 1, and then switches to N_{1+} instances for again a geometrically distributed period of mean B_{1+} of its own transmissions. For convenience, we set $B = B_{1-} + B_{1+}$ and $B_{1-} = a_1 B$, $B_{1+} = (1 - a_1)B = b_1 B$.

Another way of describing this operation is as follows. Assume that user 1 is running N_{1-} instances of the DCF. When user 1 completes a successful transmission, a new instance of the DCF is added with probability $\frac{1}{a_1 B}$. Similarly, when user 1 is running N_{1+} instances of the DCF and when it completes a successful transmission,

one of the existing DCF instances is dropped with probability $\frac{1}{b_1 B}$. Note that B can be chosen large enough to ensure that these two values correspond to probabilities.

Let T_i , $i = 1, 2$, denote the total air-time consumed by user i over a period of B successful transmissions of user 1. It is obvious that $T_1 = BE[A_1] = BA_{max}/N_1$. It is not difficult to write

$$T_2 = \frac{A_{max}}{N_2} \left(\frac{N_2}{N_{1-}} a_1 B + \frac{N_2}{N_{1+}} (1 - a_1) B \right).$$

In order to ensure air-time fairness, we should have $T_1 = T_2$, which then yields an equation in the unknown a_1 . Solving for a_1 , we obtain

$$\begin{aligned} a_1 &= \frac{N_{1-}}{N_1} (N_{1+} - N_1), \\ b_1 &= 1 - a_1 = \frac{N_{1+}}{N_1} (N_1 - N_{1-}). \end{aligned} \tag{5.5}$$

Since the choice of a_1 or b_1 does not depend on any parameters of user 2, the alternating MDCF policy proposed for user 1 can be implemented in an all-distributed fashion.

The choice of B can have some implications. If B is chosen to be very large, it is clear that short-term deviation from air-time fairness can result. On the other hand, if B is chosen to be small, then transient effects stemming from high frequency insertions and deletions of DCF instances may lead to undesirable behavior.

5.4 Air-time Fairness of MDCF for Non-integer N_i

Although we have shown air-time fairness for $K = 2$ and for only integer N_2 , the choice of the switching parameter a_i for user i , $1 \leq i \leq K$,

$$\begin{aligned} a_i &= \frac{N_{i-}}{N_i} (N_{i+} - N_i), \\ b_i &= 1 - a_i = \frac{N_{i+}}{N_i} (N_i - N_{i-}), \end{aligned} \tag{5.6}$$

as a generalization of (5.5) also leads to air-time fairness for arbitrary K and non-integer N_i , $1 \leq i \leq K$, where $N_{i-} = \lfloor N_i \rfloor$ and $N_{i+} = \lceil N_i \rceil$. Actually, it is true for MDCF

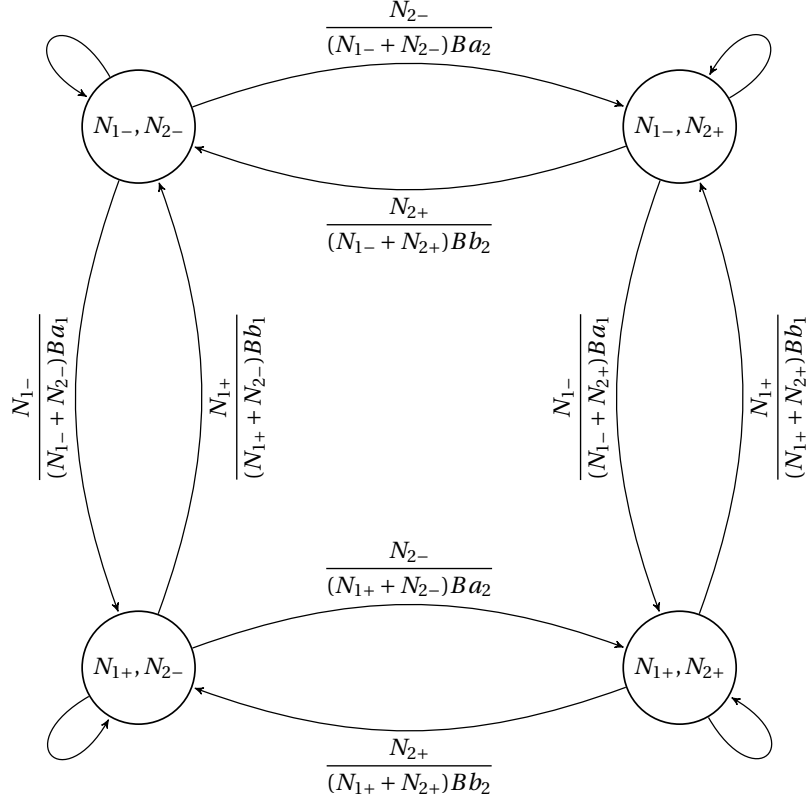


Figure 5.2: The transition diagram of the four-state Markov chain described for the two-user scenario.

that the identity (5.3) holds despite non-integer N_j and the expression for $U_i^{(MDCF)}$ in (5.4) given for integer N_i remains intact for non-integer N_i as well, provided that each user i switches among N_{i-} and N_{i+} instances according to (5.6). In this section, we give the proof for this generalization.

For the sake of convenience, let $K = 2$ first. Note that user 2 now alternates between N_{2-} and N_{2+} instances of the DCF according to the switching parameter a_2 as defined in (5.6). Let $X_i(k)$ denote the number of instances of DCF run by user i just before the k -th transmission in the system. Then the process $X(k) = \{(X_1(k), X_2(k)) : k \geq 1\}$ is a discrete-time Markov chain (DTMC) with four states, namely the states (N_{1-}, N_{2-}) , (N_{1-}, N_{2+}) , (N_{1+}, N_{2-}) , and (N_{1+}, N_{2+}) . The transition diagram of this Markov chain is depicted in Figure 5.2.

Let $\pi(i, j)$, $i = N_{1-}, N_{1+}$, $j = N_{2-}, N_{2+}$, denote the steady-state distribution of this Markov chain. It is not difficult to show that this Markov chain is reversible and its steady-state distribution is explicitly given by

$$\begin{aligned}\pi(N_{1-}, N_{2-}) &= \frac{(N_{1+} - N_1)(N_{2+} - N_2)(N_{1-} + N_{2-})}{(N_1 + N_2)}, \\ \pi(N_{1-}, N_{2+}) &= \frac{(N_{1+} - N_1)(N_2 - N_{2-})(N_{1-} + N_{2+})}{(N_1 + N_2)}, \\ \pi(N_{1+}, N_{2-}) &= \frac{(N_1 - N_{1-})(N_{2+} - N_2)(N_{1+} + N_{2-})}{(N_1 + N_2)}, \\ \pi(N_{1+}, N_{2+}) &= \frac{(N_1 - N_{1-})(N_2 - N_{2-})(N_{1+} + N_{2+})}{(N_1 + N_2)}.\end{aligned}\tag{5.7}$$

To show this, one can substitute the above distribution and verify the detailed balance equations. A packet is transmitted by user 1 at state (N_{1-}, N_{2-}) with probability $N_{1-}/(N_{1-} + N_{2-})$ or at state (N_{1-}, N_{2+}) with probability $N_{1-}/(N_{1-} + N_{2+})$ and so on. Therefore,

$$p_1 = \sum_{i \in \{N_{1-}, N_{1+}\}} \sum_{j \in \{N_{2-}, N_{2+}\}} \pi(i, j) \frac{i}{i + j} = \frac{N_1}{N_1 + N_2}.\tag{5.8}$$

Similarly, $p_2 = N_2/(N_1 + N_2)$. Therefore, we conclude that the expression for $p_i^{(MDCF)}$ in (5.3) remains intact for non-integer N_i when $K = 2$.

For the more general case $K > 2$, the process becomes $X(k) = \{(X_1(k), \dots, X_K(k)) : k \geq 1\}$. We define a state of $X(k)$ as $\mathbf{m} = (m_1, \dots, m_K)$ where $m_i \in \{N_{i-}, N_{i+}\}$, $1 \leq i \leq K$. Clearly, $X(k)$ is a DTMC with 2^K states. We claim that the steady-state probability of $X(k)$ being in state $\mathbf{m}$ is given by

$$\pi_{\mathbf{m}} = \frac{\sum_{i=1}^K m_i}{\sum_{i=1}^K N_i} \prod_{i=1}^K \left(1_{\{m_i = N_{i-}\}} (N_{i+} - N_i) + 1_{\{m_i = N_{i+}\}} (N_i - N_{i-}) \right).\tag{5.9}$$

Observe the agreement between this expression and the steady-state distribution when $K = 2$ given in (5.7).

As $X_i(k)$ is the number of instances of DCF run by user i just before the k -th transmission, and $X_i(k)$ can change only after a transmission by user i , each state $\mathbf{m}$ can transit into K other states. These states consist of the states that differ from $\mathbf{m}$ in only one of their entries.

From the entries of $\mathbf{m}$, we define $\tilde{m}_j$ as the alternate of m_j , that is

$$\tilde{m}_j = 1_{\{m_j = N_{j-}\}} N_{j+} + 1_{\{m_j = N_{j+}\}} N_{j-}.$$

Observe that $1_{\{m_j=N_{j-}\}} = 1_{\{\tilde{m}_j=N_{j+}\}}$. Using this definition, we denote the state that differs from $\mathbf{m}$ in position j by $\tilde{\mathbf{m}}(j)$:

$$\tilde{\mathbf{m}}(j) = (m_1, \dots, \tilde{m}_j, \dots, m_K).$$

Note that the sum of the entries of $\tilde{\mathbf{m}}(j)$ is $\sum_{\substack{i=1 \\ i \neq j}}^K m_i + \tilde{m}_j$.

The states that $\mathbf{m}$ can transit into are $\tilde{\mathbf{m}}(j)$, $1 \leq j \leq K$, and these also constitute the entire set of states that transit into $\mathbf{m}$. The transition probability from state $\mathbf{m}$ to state $\tilde{\mathbf{m}}(j)$, $p(\mathbf{m}, \tilde{\mathbf{m}}(j))$, can be expressed as

$$\begin{aligned} p(\mathbf{m}, \tilde{\mathbf{m}}(j)) &= \frac{m_j}{\sum_{i=1}^K m_i} \frac{1}{B} \left(\frac{1_{\{m_j=N_{j-}\}}}{a_j} + \frac{1_{\{m_j=N_{j+}\}}}{b_j} \right) \\ &= \frac{m_j}{B \sum_{i=1}^K m_i} \left(\frac{1_{\{m_j=N_{j-}\}} N_j}{m_j (N_{j+} - N_j)} + \frac{1_{\{m_j=N_{j+}\}} N_j}{m_j (N_j - N_{j-})} \right) \\ &= \frac{N_j}{B \sum_{i=1}^K m_i} \left(\frac{1_{\{m_j=N_{j-}\}}}{N_{j+} - N_j} + \frac{1_{\{m_j=N_{j+}\}}}{N_j - N_{j-}} \right). \end{aligned}$$

Similarly,

$$\begin{aligned} p(\tilde{\mathbf{m}}(j), \mathbf{m}) &= \frac{\tilde{m}_j}{\sum_{\substack{i=1 \\ i \neq j}}^K m_i + \tilde{m}_j} \frac{1}{B} \left(\frac{1_{\{\tilde{m}_j=N_{j-}\}}}{a_j} + \frac{1_{\{\tilde{m}_j=N_{j+}\}}}{b_j} \right) \\ &= \frac{\tilde{m}_j}{B \sum_{\substack{i=1 \\ i \neq j}}^K m_i + \tilde{m}_j} \left(\frac{1_{\{\tilde{m}_j=N_{j-}\}} N_j}{\tilde{m}_j (N_{j+} - N_j)} + \frac{1_{\{\tilde{m}_j=N_{j+}\}} N_j}{\tilde{m}_j (N_j - N_{j-})} \right) \\ &= \frac{N_j}{B \sum_{\substack{i=1 \\ i \neq j}}^K m_i + \tilde{m}_j} \left(\frac{1_{\{\tilde{m}_j=N_{j-}\}}}{N_{j+} - N_j} + \frac{1_{\{\tilde{m}_j=N_{j+}\}}}{N_j - N_{j-}} \right) \\ &= \frac{N_j}{B \sum_{\substack{i=1 \\ i \neq j}}^K m_i + \tilde{m}_j} \left(\frac{1_{\{m_j=N_{j+}\}}}{N_{j+} - N_j} + \frac{1_{\{m_j=N_{j-}\}}}{N_j - N_{j-}} \right). \end{aligned}$$

Also, the steady-state probability of $X(k)$ being in state $\tilde{\mathbf{m}}(j)$ can be written from

(5.9) as

$$\begin{aligned}
\pi_{\tilde{\mathbf{m}}(j)} &= \frac{\sum_{\substack{i=1 \\ i \neq j}}^K m_i + \tilde{m}_j}{\sum_{i=1}^K N_i} \prod_{\substack{i=1 \\ i \neq j}}^K \left(1_{\{m_i=N_{i-}\}}(N_{i+} - N_i) + 1_{\{m_i=N_{i+}\}}(N_i - N_{i-}) \right) \\
&\quad \times \left(1_{\{\tilde{m}_j=N_{j-}\}}(N_{j+} - N_j) + 1_{\{\tilde{m}_j=N_{j+}\}}(N_j - N_{j-}) \right) \\
&= \frac{\sum_{\substack{i=1 \\ i \neq j}}^K m_i + \tilde{m}_j}{\sum_{i=1}^K N_i} \prod_{\substack{i=1 \\ i \neq j}}^K \left(1_{\{m_i=N_{i-}\}}(N_{i+} - N_i) + 1_{\{m_i=N_{i+}\}}(N_i - N_{i-}) \right) \\
&\quad \times \left(1_{\{m_j=N_{j+}\}}(N_{j+} - N_j) + 1_{\{m_j=N_{j-}\}}(N_j - N_{j-}) \right)
\end{aligned}$$

To prove our claim (5.9), we will verify the detailed balance equations:

$$\pi_{\mathbf{m}} \sum_{j=1}^K p(\mathbf{m}, \tilde{\mathbf{m}}(j)) = \sum_{j=1}^K \pi_{\tilde{\mathbf{m}}(j)} p(\tilde{\mathbf{m}}(j), \mathbf{m}).$$

The left-hand-side equals

$$\begin{aligned}
\pi_{\mathbf{m}} \sum_{j=1}^K p(\mathbf{m}, \tilde{\mathbf{m}}(j)) &= \frac{\sum_{i=1}^K m_i}{\sum_{i=1}^K N_i} \prod_{i=1}^K \left(1_{\{m_i=N_{i-}\}}(N_{i+} - N_i) + 1_{\{m_i=N_{i+}\}}(N_i - N_{i-}) \right) \\
&\quad \times \sum_{j=1}^K \frac{N_j}{B \sum_{i=1}^K m_i} \left(\frac{1_{\{m_j=N_{j-}\}}}{N_{j+} - N_j} + \frac{1_{\{m_j=N_{j+}\}}}{N_j - N_{j-}} \right) \\
&= \frac{1}{B \sum_{i=1}^K N_i} \sum_{j=1}^K N_j \prod_{\substack{i=1 \\ i \neq j}}^K \left(1_{\{m_i=N_{i-}\}}(N_{i+} - N_i) + 1_{\{m_i=N_{i+}\}}(N_i - N_{i-}) \right),
\end{aligned}$$

and the right-hand-side is

$$\begin{aligned}
\sum_{j=1}^K \pi_{\tilde{\mathbf{m}}(j)} p(\tilde{\mathbf{m}}(j), \mathbf{m}) &= \sum_{j=1}^K \frac{\sum_{\substack{i=1 \\ i \neq j}}^K m_i + \tilde{m}_j}{\sum_{i=1}^K N_i} \prod_{\substack{i=1 \\ i \neq j}}^K \left(1_{\{m_i=N_{i-}\}}(N_{i+} - N_i) + 1_{\{m_i=N_{i+}\}}(N_i - N_{i-}) \right) \\
&\quad \times \left(1_{\{m_j=N_{j+}\}}(N_{j+} - N_j) + 1_{\{m_j=N_{j-}\}}(N_j - N_{j-}) \right) \\
&\quad \times \frac{N_j}{B \sum_{\substack{i=1 \\ i \neq j}}^K m_i + \tilde{m}_j} \left(\frac{1_{\{m_j=N_{j+}\}}}{N_{j+} - N_j} + \frac{1_{\{m_j=N_{j-}\}}}{N_j - N_{j-}} \right)
\end{aligned}$$

$$= \frac{1}{B \sum_{i=1}^K N_i} \sum_{j=1}^K N_j \prod_{\substack{i=1 \\ i \neq j}}^K \left(1_{\{m_i=N_{i-}\}}(N_{i+} - N_i) + 1_{\{m_i=N_{i+}\}}(N_i - N_{i-}) \right),$$

since

$$\left(1_{\{m_j=N_{j+}\}}(N_{j+} - N_j) + 1_{\{m_j=N_{j-}\}}(N_j - N_{j-}) \right) \left(\frac{1_{\{m_j=N_{j+}\}}}{N_{j+} - N_j} + \frac{1_{\{m_j=N_{j-}\}}}{N_j - N_{j-}} \right) = 1.$$

As the detailed balance equations are verified, we conclude that the steady-state probability of $X(k)$ being in state $\mathbf{m}$ given in (5.9) is indeed correct.

Now, to prove that MDCF yields air-time fairness with arbitrary $K > 2$, we have to show that for any user i , $1 \leq i \leq K$, we have

$$\sum_{\mathbf{m} \in M} \pi_{\mathbf{m}} \frac{m_i}{\sum_{j=1}^K m_j} = \frac{N_i}{\sum_{j=1}^K N_j},$$

where M is the set of all 2^K states. For this purpose, we will first show that

$$\sum_{\mathbf{m} \in M} \left(\prod_{j=1}^K 1_{\{m_j=N_{j-}\}}(N_{j+} - N_j) + 1_{\{m_j=N_{j+}\}}(N_j - N_{j-}) \right) = 1 \quad (5.10)$$

for any K . For $K = 1$,

$$\begin{aligned} \sum_{\mathbf{m} \in M} \left(\prod_{j=1}^K 1_{\{m_j=N_{j-}\}}(N_{j+} - N_j) + 1_{\{m_j=N_{j+}\}}(N_j - N_{j-}) \right) &= (N_{1+} - N_1) + (N_1 - N_{1-}) \\ &= 1. \end{aligned}$$

Now assume that (5.10) holds for K users. For $K + 1$ users, define

$$M_{(K+1)-} = \{ \mathbf{m} \in M \mid m_{K+1} = N_{(K+1)-} \}, \quad (5.11)$$

$$M_{(K+1)+} = \{ \mathbf{m} \in M \mid m_{K+1} = N_{(K+1)+} \}, \quad (5.12)$$

where it is clear that $M_{(K+1)-} \cup M_{(K+1)+} = M$. Then we can write

$$\begin{aligned} \sum_{\mathbf{m} \in M} \left(\prod_{j=1}^{K+1} 1_{\{m_j=N_{j+}\}}(N_{j+} - N_j) + 1_{\{m_j=N_{j-}\}}(N_j - N_{j-}) \right) \\ = \sum_{\mathbf{m} \in M_{(K+1)-}} (N_{(K+1)+} - N_{(K+1)}) \left(\prod_{j=1}^K 1_{\{m_j=N_{j+}\}}(N_{j+} - N_j) + 1_{\{m_j=N_{j-}\}}(N_j - N_{j-}) \right) + \\ \sum_{\mathbf{m} \in M_{(K+1)+}} (N_{(K+1)} - N_{(K+1)-}) \left(\prod_{j=1}^K 1_{\{m_j=N_{j-}\}}(N_{j+} - N_j) + 1_{\{m_j=N_{j+}\}}(N_j - N_{j-}) \right) \end{aligned}$$

$$\begin{aligned}
&= (N_{(K+1)+} - N_{(K+1)}) + (N_{(K+1)} - N_{(K+1)-}) \\
&= 1.
\end{aligned}$$

Therefore, (5.10) holds by induction. Then, for a user i in a system of K users,

$$\begin{aligned}
&\sum_{\mathbf{m} \in M} \pi_{\mathbf{m}} \frac{m_i}{\sum_{j=1}^K m_j} \\
&= \sum_{\mathbf{m} \in M} \frac{m_i}{\sum_{j=1}^K m_j} \frac{\sum_{j=1}^K m_j}{\sum_{j=1}^K N_j} \prod_{j=1}^K \left(1_{\{m_j=N_{j-}\}}(N_{j+} - N_j) + 1_{\{m_j=N_{j+}\}}(N_j - N_{j-}) \right) \\
&= \frac{1}{\sum_{j=1}^K N_j} \sum_{\mathbf{m} \in M} m_i \prod_{j=1}^K \left(1_{\{m_j=N_{j-}\}}(N_{j+} - N_j) + 1_{\{m_j=N_{j+}\}}(N_j - N_{j-}) \right) \\
&= \frac{1}{\sum_{j=1}^K N_j} \left[\sum_{\mathbf{m} \in M_{i-}} N_{i-} (N_{i+} - N_i) \prod_{\substack{j=1 \\ j \neq i}}^K \left(1_{\{m_j=N_{j-}\}}(N_{j+} - N_j) + 1_{\{m_j=N_{j+}\}}(N_j - N_{j-}) \right) \right. \\
&\quad \left. + \sum_{\mathbf{m} \in M_{i+}} N_{i+} (N_i - N_{i-}) \prod_{\substack{j=1 \\ j \neq i}}^K \left(1_{\{m_j=N_{j-}\}}(N_{j+} - N_j) + 1_{\{m_j=N_{j+}\}}(N_j - N_{j-}) \right) \right] \\
&= \frac{N_{i-} (N_{i+} - N_i) + N_{i+} (N_i - N_{i-})}{\sum_{j=1}^K N_j} \\
&= \frac{N_i}{\sum_{j=1}^K N_j}.
\end{aligned}$$

This concludes the proof of $p_i = N_i / \sum_{j=1}^K N_j$ for $K > 2$, and leads us to the conclusion that air-time fairness is achievable in an all-distributed manner for an arbitrary number of users in the WLAN via the switching mechanism of MDCE.

5.5 Practical Aspects

In this section, we will offer solutions to some issues in practical implementations for MDCE.

First, we describe how a physical node behaves under MDCE operation. Assume that a user running N instances of the DCF, has a frame to transmit. As an example,

let $N = 3$ and $CW_{min} = 32$. Let the initial backoff timer values of the three DCF instances be 3, 12, and 24. Then after DIFS plus three slots (assuming an idle channel), the node will transmit on behalf of the first virtual node. If the transmission queue is empty at the epoch of reception of the acknowledgment of this frame, then all contention windows are reset. Upon an arriving frame, contention windows of all three nodes will (independently) be initialized according to DCF rules.

If there are more frames to transmit, the initial contention window of the first virtual node is chosen uniformly from the set $\{0, \dots, CW_{min} - 1\}$ and the other two will continue to be decremented at each slot as long as the channel is idle.

An interesting situation arises when the backoff timer values of multiple virtual nodes turn out to be identical. As an example, let the initial backoff timer values of the three DCF instances be 12, 12, and 24. If this were a real WLAN, there would have been a collision due to the two virtual nodes. However, since the node associated with these virtual nodes will realize that this is an internal collision, it defers from transmitting. We call this behavior *Internal Collision Prevention* (ICP). In this case, we propose that the virtual nodes causing the internal collision behave like they experienced a real collision, i.e., their contention windows are doubled.

ICP also gives rise to an interesting situation. Consider for example, a scenario with two nodes having one and two virtual nodes respectively. Suppose that all three virtual nodes have the same backoff counter values and therefore are destined for a collision. However, due to ICP, the node with two virtual nodes will defer from transmission and as a result, the other node transmits successfully.

Note that ICP behavior is not included in the mathematical model previously described. However, as the simulation study will demonstrate, ICP has marginal impact on air-time fairness but is obviously beneficiary in terms of overall channel utilization. In all the numerical examples to follow, we employ ICP.

Another practical issue is assigning a value to A_{max} , which needs to be larger than $\max_j E[A_j]$. The ideal value would be $A_{max} = \max_j E[A_j]$, but this would in general require communication between stations to find out and disseminate this value. Instead, we propose to set A_{max} to the ratio of the maximum transmission

unit allowed to the minimum data rate supported. Then, each station can independently calculate A_{max} since the maximum transmission unit and the minimum supported data rate are known through protocol parameters.

The obvious disadvantage of this solution is that stations might run more algorithms than necessary. For instance, consider a scenario where each station runs IEEE 802.11b, has 11 Mbps data rate and transmits the largest frames allowed. In this case, MDCF would ideally work with each station running a single algorithm since all A_j values are equal. However, with the solution described above, each station would want to run 11 algorithms apiece, which will result in degraded channel utilization and overall throughput. We offer limited frame aggregation to remedy this situation, and an example is provided in the next section.

Lastly, we assert that a station knows the mean of the air-time it requires, $E[A_j]$. To this end, we propose that a station can compute the mean payload it transfers at each transmission opportunity using damped averaging, and calculate $E[A_j]$ by dividing this value by its current data rate. Specifically, starting with the size of a maximum transmission unit as the initial value, a station j computes its current payload estimate, $B_j^e(n)$, using its previous estimate, $B_j^e(n-1)$, via

$$B_j^e(n) = \alpha B_j^e(n-1) + (1-\alpha) B_j(n) \quad (5.13)$$

after each successful transmission, where $B_j(n)$ is the size of the transmitted payload. Here, $\alpha \in (0, 1)$ is a parameter to be selected.

5.6 Simulation Study

The simulation study is carried out via an event-based simulator using Matlab. The inter-frame spaces and the ACK mechanism including the ACK timeout are implemented, but propagation delays are ignored. We also assume there are no hidden nodes. We define the air-time fairness metric AF in a K -node scenario as the ratio of the minimum of the air-times that the nodes get, to their maximum:

$$AF = \frac{\min_{1 \leq i \leq K} U_i}{\max_{1 \leq i \leq K} U_i}.$$

The parameter AF varies between 0 (worst case) and 1 (ideal air-time fairness). We present six numerical examples by which we investigate the performance of MDCF with the help of three metrics:

- (i) air-time fairness AF ,
- (ii) channel utilization U ,
- (iii) cumulative throughput T which is defined as the long-term average rate of successful bits transmitted through the channel.

We experimented using IEEE 802.11b parameters, which are given in Table 5.1.

Another important point with MDCF is that it yields a system with more participants compared with the standard DCF. The optimal value of CW_{min} is shown in [93] to depend linearly on the number of stations. Therefore, the value of CW_{min} for DCF should be multiplied with the expected increase in the number of virtual nodes with MDCF. For the specific case of IEEE 802.11b, the mean increase in the number of (virtual) nodes with MDCF relative to DCF is $(1 + 2 + 5.5 + 11)/4 = 4.875$ in case we have an equal number of users at each raw bit rate. Therefore, we suggest that the CW_{min} value for MDCF, denoted by $CW_{min}^{(MDCF)}$, to be set to $156 = 32 \times 4.875$ (unless stated otherwise) as opposed to 32 which is the default value of CW_{min} for the standard DCF in typical implementations.

Throughout the simulations, we set $B = 100$. Experimentation with varying B reveal that unless B is chosen small (order of ones) or large (order of several thousands), the choice of the parameter B has marginal effect on MDCF performance. Note that depending on the system, it is possible that any one of the quantities $\frac{1}{a_i B}$

Slot time	20 μ s	CW_{min}	32
DIFS period	50 μ s	$CW_{min}^{(MDCF)}$	156
SIFS period	10 μ s	CW_{max}	1024
ACK timeout	300 μ s	$CW_{max}^{(MDCF)}$	4992
ACK frame	14 bytes	Max. frame size	1500 bytes

Table 5.1: Simulation parameters

or $\frac{1}{b_i B}$ for $1 \leq i \leq K$ does not correspond to a probability for this choice of B . One solution to this issue could be increasing B . However, this situation indicates that a_i or b_i is relatively small. Consequently, the number N_i should be very close to either N_{i-} or N_{i+} ; see identity (5.6). Therefore, in this case, switching would not be necessary at all, which is the approach we prefer.

We ran all the simulations until all virtual nodes have successfully transmitted at least 10000 frames except for Example IV, which has a fixed simulation time. The parameter A_{max} is fixed to 12 ms which is the air-time required to transmit a 1500 B frame at 1 Mbps.

Example I In the first example, a simulation study is carried out for a scenario with two nodes whose data rates are varied in the range 1 Mbps up to 11 Mbps in 0.5 Mbps steps. Obviously, this set of data rates includes non-standard values for the purpose of justifying the air-time fairness feature of MDCE. We allow one frame transmission at a given transmission opportunity and the frame length is fixed to 1500 bytes for all users. The results are given in figures 5.3, 5.4 and 5.5. It is observed that MDCE achieves almost perfect air-time fairness, whereas the air-time fairness metric of the standard DCF is proportional with the ratio of the data rates. Also, a substantial gain in the cumulative throughput is demonstrated especially for asymmetric data rates, which is an immediate consequence of air-time fairness. Note that the price of the air-time fairness in terms of channel utilization reduction is no more than 8% for MDCE. Standard DCF suffers a reduction in channel utilization only due to the increase in the rates of the nodes which makes the busy periods shorter, whereas MDCE also loses in terms of channel utilization U stemming from addition of new (virtual) nodes which leads to increased collision probability as shown in [92].

Example II With the second example, we demonstrate the effectiveness of MDCE in air-time fairness for non-integer N_i . We present a simulation example with two nodes. Node 1 has a rate of 1 Mbps, whereas the rate of node 2 is varied from 1 Mbps to 3 Mbps in 0.1 Mbps steps. Both stations transmit 1500 B frames. The proposed

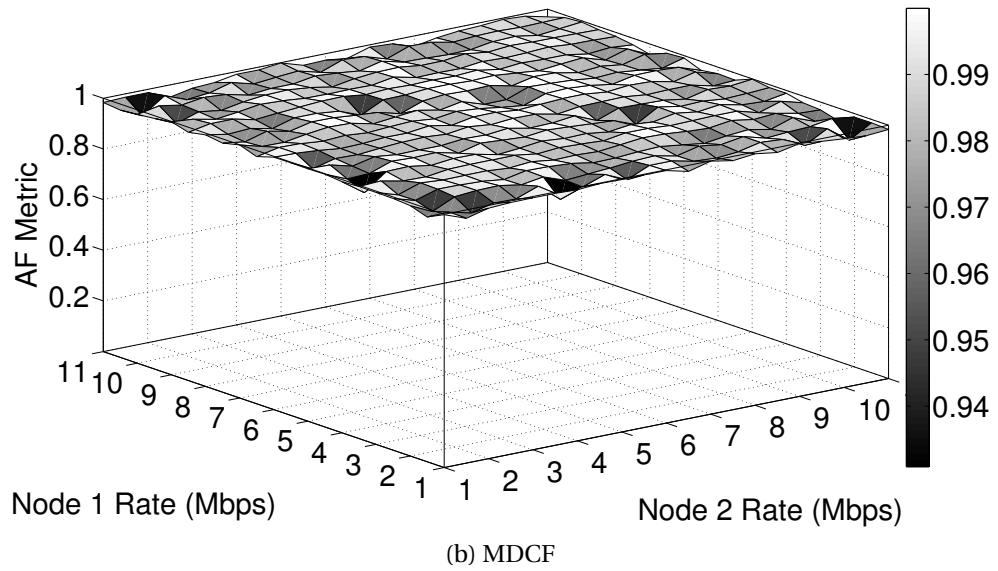
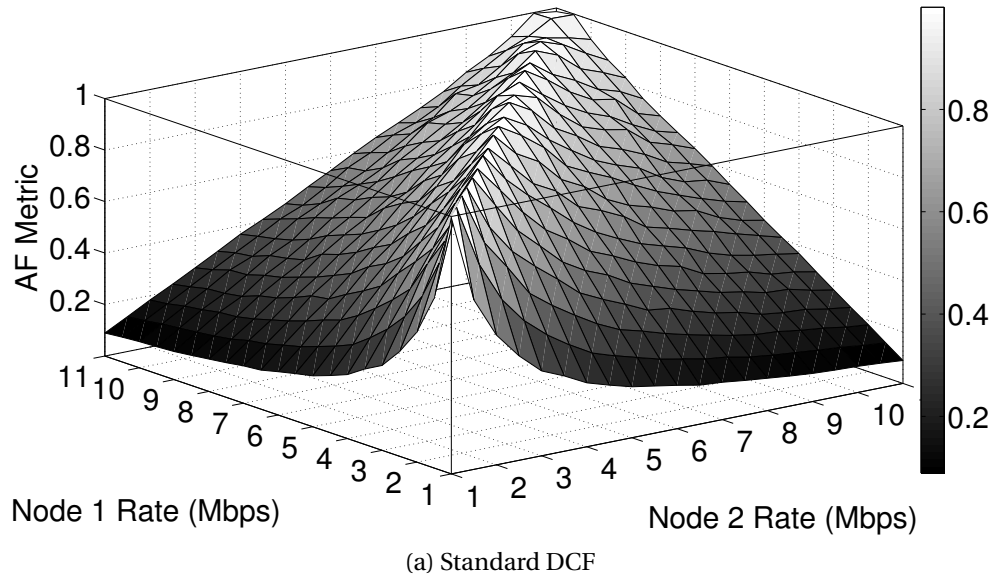
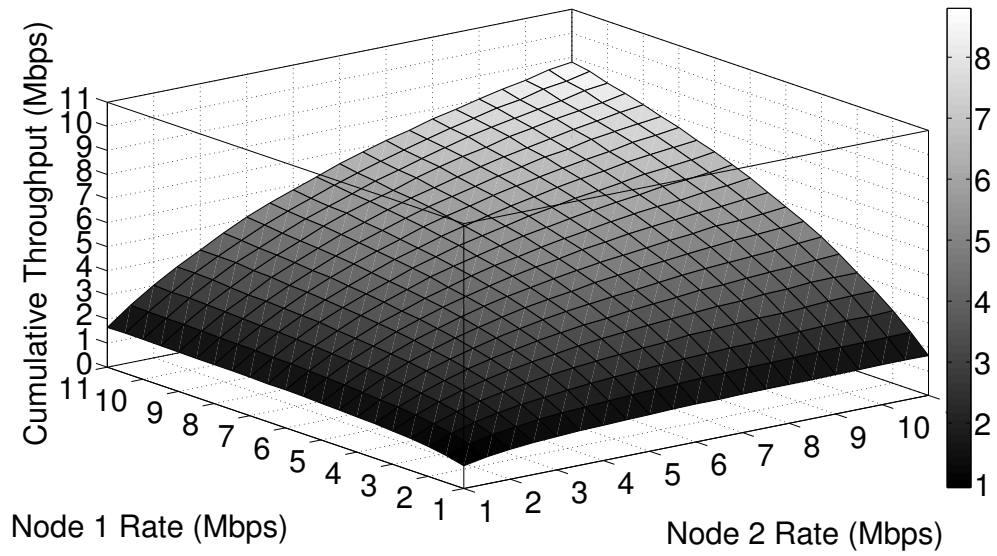
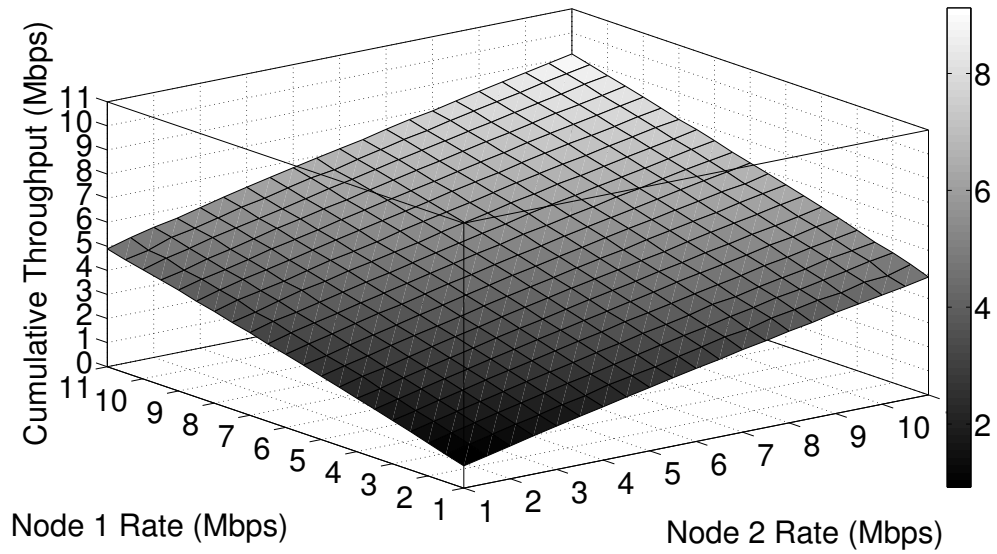


Figure 5.3: Air-time fairness comparison of standard DCF and MDCF as a function of raw bit rates.



(a) Standard DCF



(b) MDCF

Figure 5.4: Cumulative throughput comparison of standard DCF and MDCF as a function of raw bit rates.

method is compared to three cases where node 2 uses $\lfloor N_2 \rfloor$, $\lceil N_2 \rceil$, and the value N_2 rounded to the closest integer. The AF metrics obtained with each of these methods are plotted in Figure 5.6. Not surprisingly, when $1 \leq N_2 < 1.5$ and $2 \leq N_2 < 2.5$, the rounded value is equal to $\lfloor N_2 \rfloor$ and consequently, the two curves closely follow each other. We have a similar case with $\lceil N_2 \rceil$ when $1.5 \leq N_2 \leq 2$ and $2.5 \leq N_2 \leq 3$. All three curves deviate significantly from the ideal AF value of 1, whereas MDCF stays very close, thus illustrating the effectiveness of the proposed switching mechanism. Note that although the data rates used in this simulation example are not standard values, the corresponding N_2 values can be encountered due to varying payloads, frame aggregation mechanisms, and service differentiation schemes.

Example III We demonstrate the scalability of MDCF in terms of the number of nodes using the system. For this purpose, we define a basic group of 4 nodes with data rates 1, 2, 5.5, and 11 Mbps and simulate scenarios with up to 10 groups of nodes. The frame length is again set to 1500 bytes for all users. Our findings that are provided in Figure 5.7 show that MDCF maintains air-time fairness whereas the AF parameter of the standard DCF is stuck at around 1/11 as expected. Therefore, we conclude that MDCF remains precise even in scenarios with relatively large number of nodes.

Example IV With the fourth example, we demonstrate the performance of the proposed method under scenarios in which the behavior of the stations vary with time. We simulate a scenario with five stations A-E. The rates and the frame sizes of each station throughout the simulation are summarized in Table 5.2. To clarify, Station C for instance, has no frames to send in the first 100 seconds. Then, it starts transmitting frames whose lengths are uniformly distributed between 500 and 1500 bytes, with a raw rate of 5.5 Mbps. At 500 seconds, its frame size is fixed to 1500 bytes. Another 100 seconds later, its rate becomes 11 Mbps. In addition to stations becoming online and going offline, the envisaged scenario encompasses a number of possible stress conditions including variable frame sizes and rate changes. For the payload estimates, (5.13) is used online with $\alpha = 0.95$. The air-time utilizations of individual stations (U_i) are plotted in Figure 5.8. The U_i curves converging to the same

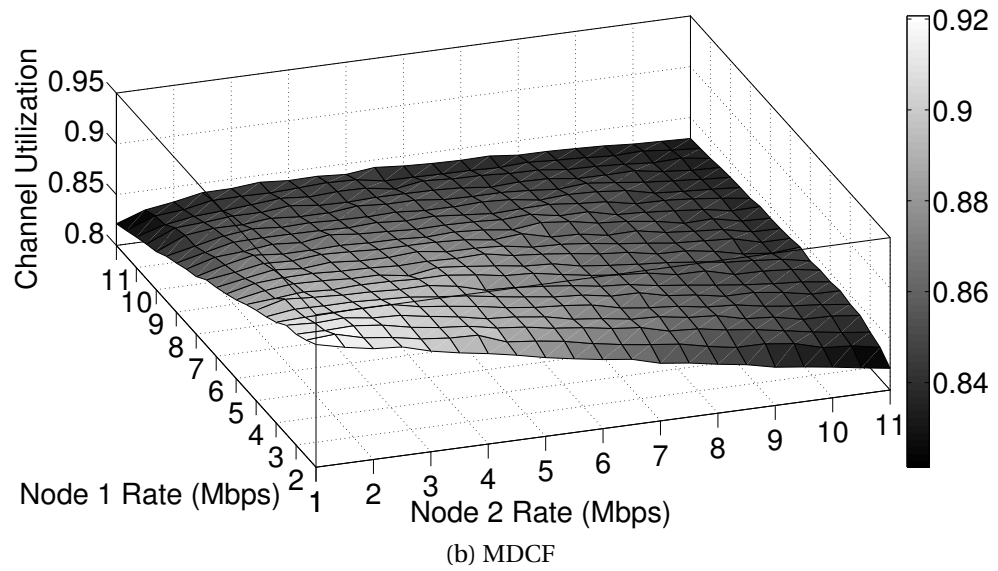
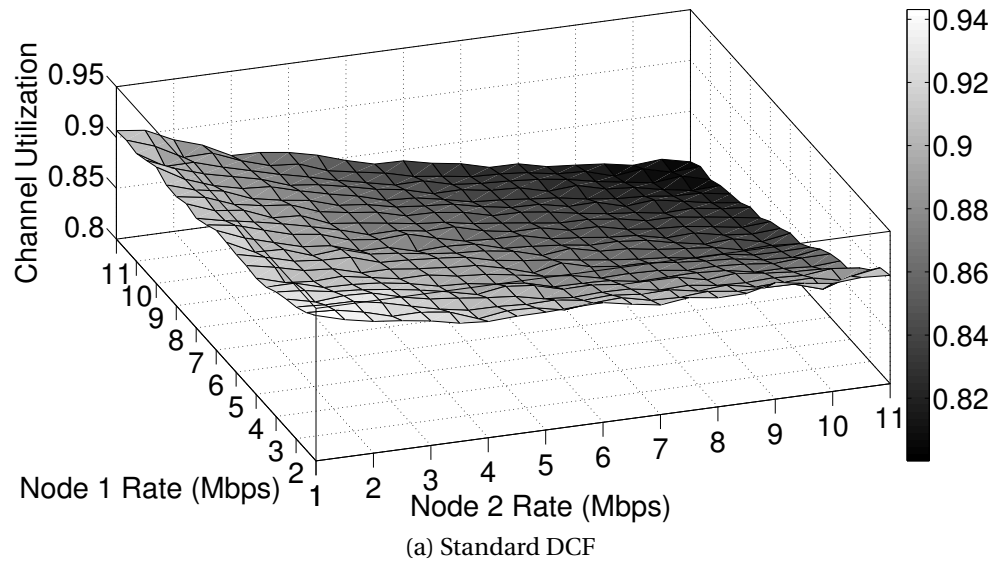


Figure 5.5: Channel utilization comparison of standard DCF and MDCF as a function of raw bit rates.

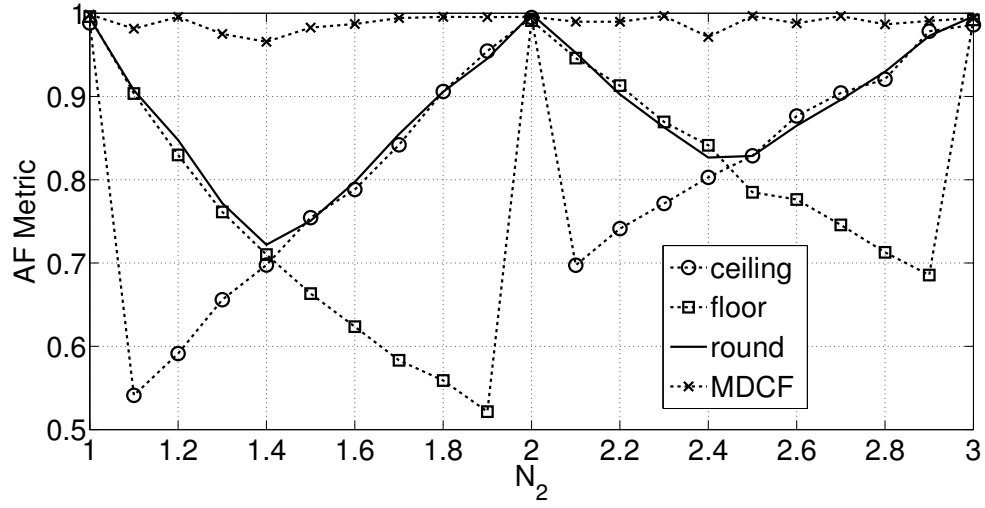


Figure 5.6: AF metrics obtained with using $\lfloor N_2 \rfloor$, $\lceil N_2 \rceil$, the value N_2 rounded to the closest integer, and with the switching algorithm of MDCF. There are two stations in this scenario with $N_1 = 1$.

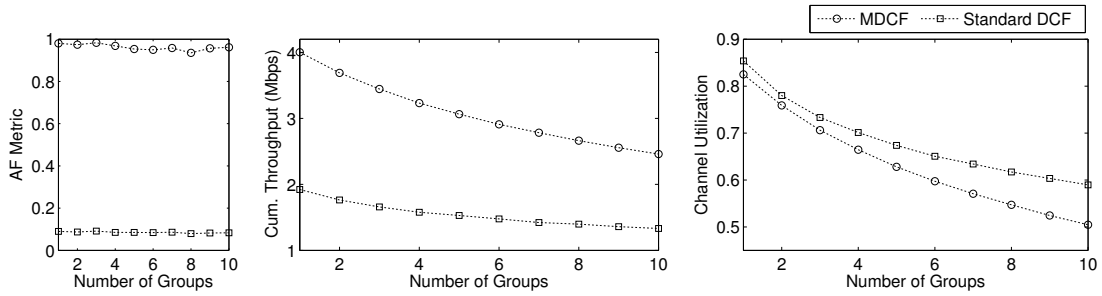


Figure 5.7: Comparison of standard DCF and MDCF under scenarios with up to 40 nodes.

value following each disturbance demonstrates the effectiveness of the proposed method in transient situations as well as the steady-state. The online averaging algorithm given in (5.13) increases the time required for convergence when frames are variable-sized. The particular common air-time utilization value after convergence depends on the number of online stations for the corresponding scenario as well as the type of online stations.

Example V An observation one can make due to examples I and III is that addition of new nodes to a system running MDCF slightly worsens the performance in terms of channel utilization more than it does in the case of standard DCF in general. This is due to the fact that adding a single node to an MDCF system simply means adding potentially multiple virtual nodes, leading to increased collision probability, as opposed to the addition of just one node under standard DCF. In order to mitigate this adverse effect, we propose using frame aggregation in conjunction with MDCF. We let each node aggregate a number of frames into a burst at each transmission attempt in a way that its air-time requirement for the burst does not exceed A_{max} (the air-time required for the slowest node to transmit one single frame), and the number of frames aggregated does not exceed a predetermined value denoted by F_{max} . In this example, we demonstrate the performance of MDCF with the described frame aggregation scheme. We simulate MDCF with four nodes with rates 1, 2, 5.5, and 11 Mbps, all using frame sizes of 1500 bytes while varying the parameter F_{max} . For MDCF, we use two CW_{min} values, 156 and 128, the latter being an integer power of two. We denote the number of aggregated frames with F_{agg} . The results are given in Table 5.3.

When we allow aggregation, i.e., $F_{max} > 1$, the number of virtual nodes used for MDCF per node is reduced, thus increasing the channel utilization with respect to the case $F_{max} = 1$. We conclude that MDCF with moderate frame aggregation, i.e., $F_{max} = 3$, meets the air-time fairness requirement with a channel utilization surpassing that of the standard DCF. Consequently, there may not be any need for aggressive frame aggregation policies such as the one with $F_{max} = 11$, recalling that such policies may lead to short-term unfairness among nodes. Moreover, the results of CW_{min} being 156 or 128 are only slightly different, meaning that CW_{min} can also

be chosen to be a power of two in MDCF in real implementations without compromising air-time fairness.

In this example, we assume that back-to-back frames are sent without any gaps within between, but more general frame aggregation schemes including the ones mentioned in [70] can also be used.

Example VI In the final example, we compare the performance of MDCF to the technique of CW_{min} adjustment in the context service differentiation. In this scenario, which has been inspired from [58] (see Figure 2), all stations have 11 Mbps rates and their frame sizes are 1500 bytes. There are two classes into which the stations are evenly distributed. The stations in the high priority class are to be given twice the throughput that the stations in the low priority class. For this purpose, in the CW_{min} adjustment technique, the low priority stations are assigned a CW_{min} value that is twice that of the high priority stations. On the other hand, with MDCF, high priority stations run two back-off algorithms ($N = 2$) each, as opposed to a single algorithm run by each low priority station.

In Figure 5.9, the ratios of the throughput obtained by the high priority class to that of the low priority class is plotted as a function of the total number of stations in the system for various CW_{min} values. In the graph for the CW_{min} adjustment technique, the CW_{min} values given in the legend are used by high priority nodes, whereas the CW_{min} values are the same for all stations in MDCF and are given in the corresponding legend. As the number of overall stations increases, CW_{min} adjustment technique has difficulty maintaining the required throughput ratio especially with lower CW_{min} values, dropping to as low as 1.75 whereas the ratios obtained by MDCF remains within the range [1.973, 2.025]. We plot the channel utilization under this scenario in Figure 5.10, which shows similarity between the performances of CW_{min} adjustment and MDCF. On the other hand, both figures indicate that the performance of CW_{min} adjustment is sensitive to the specific scenario it is run. Bearing in mind that we aim to come up with a distributed scheme, it is likely that a predetermined and fixed set of parameters will be used under any scenario. This

leads us to believe that one can not expect a consistent level of throughput differentiation from CW_{min} adjustment, contrary to what appears to be valid for MDCE.

5.7 Conclusion

We presented a distributed air-time fair MAC (MDCF) for multi-rate WLANs by allowing multiple instances of the DCF back-off algorithm to be run at each node. We also propose to use MDCF together with frame aggregation to alleviate the slight reduction in channel utilization observed for MDCF without frame aggregation. We show that MDCF achieves air-time fairness in all the scenarios studied. Although MDCF is proposed as an instrument for air-time fairness, it can also be used for service differentiation in WLANs as shown in Example VI, which is a topic for further exploration. The impact of non-zero wireless packet errors and the capture effect on MDCF performance is also left for future research.

Time (s)	St. A	St. B	St. C	St. D	St. E
0–100	1 Mbps 1500 B	11 Mbps 1500 B	offline	offline	offline
100–200			5.5 Mbps U[500,1500]		
200–300					
300–400					
400–500		5.5 Mbps 1000 B		2 Mbps 1000 B	
500–600			5.5 Mbps 1500 B		
600–700				11 Mbps 1500 B	
700–800					
800–900		5.5 Mbps U[500,1500]			2 Mbps 1500 B
900–1000		5.5 Mbps U[1000,1500]			
1000–1100		2 Mbps U[1000,1500]			
1100–1200				11 Mbps 1500 B	

Table 5.2: Simulation scenario for Example IV: The data rates in Mbps and payloads in bytes. U[] denotes the uniform distribution.

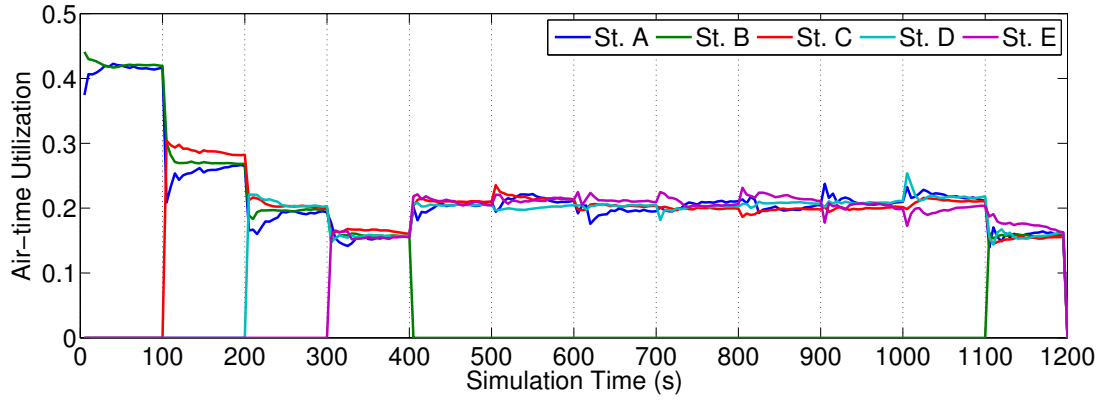


Figure 5.8: Air-time utilization for each station under the scenario summarized in Table 5.2.

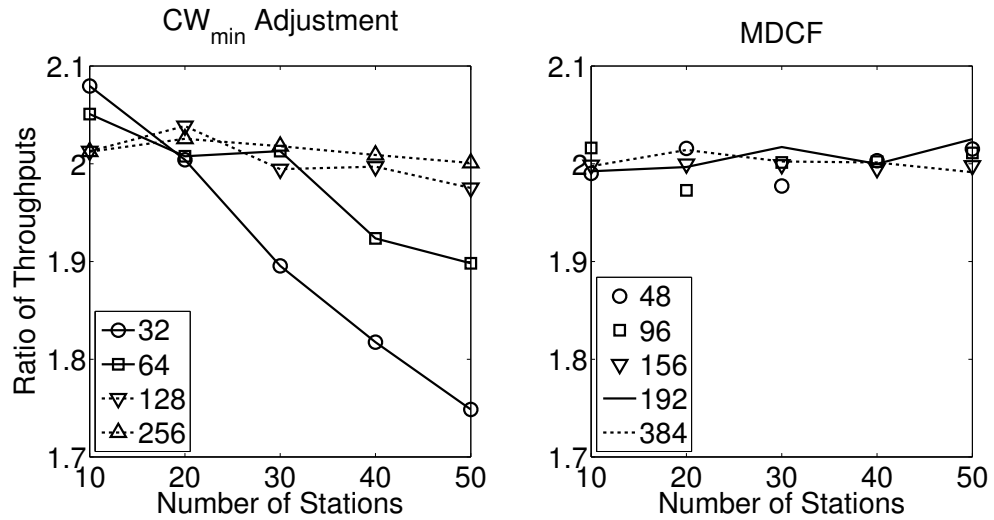


Figure 5.9: Ratios of the throughput obtained by the high priority class to that of the low priority class for various network sizes and CW_{min} values.

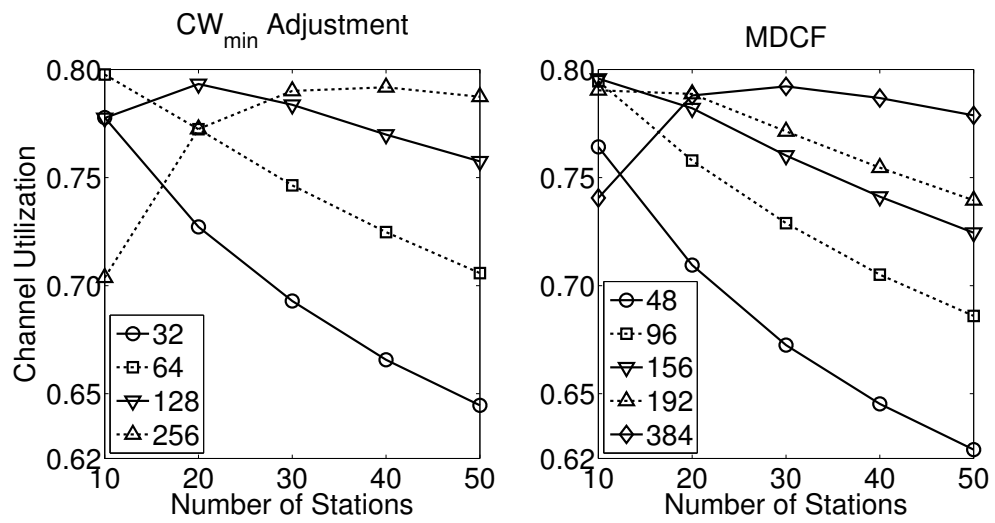


Figure 5.10: Channel utilization for various network sizes and CW_{min} values.

Node:		1 Mbps		2 Mbps		5.5 Mbps		11 Mbps		AF		T (Mbps)		U	
F_{max}	F_{agg}	N	F_{agg}	N	F_{agg}	N	F_{agg}	N	F_{agg}	$CW_{min}^{(MDCF)}$	$CW_{min}^{(MDCF)}$	156	128	156	128
1	1	1	1	2	1	5.5	1	11	1	0.9826	0.9774	4.011	3.952	0.8243	0.8136
2	1	1	2	1	2	2.75	2	5.5	2	0.9959	0.9859	4.313	4.285	0.8856	0.8805
3	1	1	2	1	3	1.833	3	3.667	3	0.9653	0.9843	4.453	4.397	0.9086	0.9061
4	1	1	2	1	4	1.375	4	2.75	4	0.9851	0.9798	4.484	4.471	0.9210	0.9202
5	1	1	2	1	5	1.1	5	2.2	5	0.9671	0.9667	4.501	4.530	0.9282	0.9276
6	1	1	2	1	5	1.1	5	1.833	6	0.9892	0.9914	4.546	4.533	0.9335	0.9318
7	1	1	2	1	5	1.1	5	1.571	7	0.9744	0.9681	4.535	4.526	0.9340	0.9349
8	1	1	2	1	5	1.1	5	1.375	8	0.9515	0.9742	4.513	4.616	0.9376	0.9392
9	1	1	2	1	5	1.1	5	1.222	9	0.9649	0.9688	4.599	4.636	0.9370	0.9414
10	1	1	2	1	5	1.1	5	1.1	10	0.9710	0.9615	4.587	4.632	0.9401	0.9408
11	1	1	2	1	5	1.1	5	1	11	0.9849	0.9835	4.579	4.593	0.9406	0.9412
Standard DCF:										0.0898		1.922		0.8538	

Table 5.3: MDCF system performance metrics AF , T , and U , as a function of the frame aggregation parameter F_{max} for two values of $CW_{min}^{(MDCF)}$ for a scenario with four nodes with four different raw bit rates.

Chapter 6

Conclusion

In this thesis, we develop a framework for solving continuous feedback Markov fluid queues (CFMFQ). The framework sits on top of a method that was proposed for multi-regime Markov fluid queues (MRMFQ). The CFMFQ is approximated by an MRMFQ by discretizing the buffer space appropriately. Then, the solution of the resulting MRMFQ is formed using the additive decomposition method, which is based on ordered Schur decomposition and known to be numerically stable.

The complete solution of the MRMFQ requires the solution of a system of linear equations that arise due to the boundary conditions. We describe an algorithm based on block-tridiagonal LU decomposition that has linear complexity with respect to the number of regimes. This enables us to use very large values for the number of regimes of the MRMFQ that approximates the given CFMFQ. In this way, the results we obtain turn out to be very accurate, as demonstrated by the numerous examples provided throughout the thesis. Note that although we propose to use the block-tridiagonal LU decomposition algorithm in the framework of CFMFQs, it is basically a tool for MRMFQs. Therefore, we also address the problem of solving large MRMFQs efficiently, a problem that has not been addressed before.

Within the CFMFQ framework, we solve two main problems. The first one is the workload-bounded MAP/PH/1 queue, in which arrivals occur according to a workload-dependent MAP, job sizes are PH-type distributed, and the service speed

is allowed to depend on the buffer level in a general manner. We model the workload-bounded MAP/PH/1 queue in three different settings:

- (i) The Infinite Buffer (IB): The buffer capacity is infinite in this setting and therefore, there are no rejections.
- (ii) The Finite Buffer with Partial Rejection (FB-PR): The buffer capacity is finite. When the size of an arriving job exceeds the available buffer space at the time of its arrival, the buffer is allowed to overflow. In this setting, part of the job that fits into the available buffer space is accepted, and the rest is lost. We solve the steady-state distribution of this system and obtain the workload loss probability, defined as the ratio of the amount of the workload rejected to the overall amount of workload that arrives at the buffer.
- (iii) The Finite Buffer with Complete Rejection (FB-CR): The buffer capacity is finite. When the size of an arriving job exceeds the available buffer space at the time of its arrival, the job is rejected completely. Note that with FB-CR, the resulting model turns out to be a CFMFQ even if none of the parameters of the queue depend on the buffer level. This is due to the rejection policy that causes an intrinsic feedback. The modeling of this system is a novel contribution of this thesis. We solve the job loss probability as well as the workload loss probability in this setting.

Several numerical examples show that our proposed method is able to solve these systems very accurately. Moreover, we describe the two-class version of the FB-CR setting.

Possible future directions in this problem are

- the comparison of our method to the existing methods that approach the problem as a boundary value problem,
- the investigation of the effect of the discretization method and the number of regimes,

- describing a method for identifying the optimal level of discretization for a given level of accuracy, and
- improving the speed of the overall method by detailed optimization of the algorithm, for instance by testing different methods for matrix exponentiation as this is identified as a major contributor to the overall run time.

The second problem we solve within the framework of CFMFQs is the stochastic model of the horizon-based reservation mechanism in OBS networks. We model the horizon parameter as a CFMFQ and solve its steady-state distribution. Using this distribution relevant statistics such as burst blocking probability can be computed. We also model the horizon-based reservation mechanism in combination with FDLs. Moreover, we allow multiple traffic classes for either setting. The contributions of the thesis on this topic can be listed as follows:

- We develop a method for representing the offset time with a single state (per replica due to the MAP states) by employing the hazard rate concept. This also allows us to model generally distributed offset times
- We propose methods for modeling systems in which the offset time distribution involves impulses, or is of finite support.
- In the most general setting, we are able to model systems with two traffic classes with separate offset distributions and that are allowed to access different sets of FDLs.

The focus on this part of the study is on the methodology, rather than provisioning systems that employ horizon-based reservation and/or FDLs. Therefore, an obvious research direction is a provisioning study based on the provided methodology. Also, other aspects of QoS such as delay performance can be investigated within this framework. Another research direction is the exploration of multi-channel systems. In addition, the method we proposed that is based on the hazard rate concept can be applied to the modeling of delayed reservation schemes in more general settings.

We have also identified a problem in the context of risk theory for which our CFMFQ framework is a good suitor. In this so-called “Finite Horizon Ruin Problem”,

an insurer starts a business with a certain amount of reserves, and collects premiums from its clients according to given function that can depend on the current amount of reserves it possesses. Claims from clients that have PH-type distributed sizes arrive according to a MAP. This specific problem searches for the answer of the following question: What is the probability of ruin within a given amount of time, called the horizon? Note that the infinite counterpart of this problem, the probability of ruin in the steady-state, can also be solved with our method.

Lastly, we propose a distributed algorithm that achieves air-time fairness in multi-rate IEEE 802.11 WLANs. Air-time fairness is a solution to the “performance anomaly” problem of the multi-rate IEEE 802.11 WLANs. The proposed method is based on each node running multiple instances of the distributed coordination function. When this number is not an integer, we propose a method that switches the number of DCF instances between the floor and the ceiling of the number of required instances.

Through extensive simulations, we show that our algorithm achieves air-time fairness even in dynamic scenarios where the frame sizes of the nodes are modeled as random variables, rate adaptation takes place, and nodes are allowed to go offline or become online. We also compare our method to an existing method that aims the same objective, the method of CW_{min} adaptation, and show that the proposed method has advantages over CW_{min} adaptation. Moreover, we provide a discrete Markov-chain based proof for air-time fairness in scenarios with arbitrary number of nodes and non-integer DCF instances. Further studies on this topic may be on the analysis of the proposed method in channels with non-negligible bit/frame error rates. The method can also be used for service differentiation, as briefly demonstrated, which is another exploration option.

Bibliography

- [1] D. Anick, D. Mitra, and M. M. Sondhi, “Stochastic theory of a data handling system with multiple sources,” *Bell Syst. Tech. Jour.*, vol. 61, pp. 1871–1894, 1982.
- [2] V. G. Kulkarni, “Fluid models for single buffer systems,” in *Frontiers in Queuing: Models and Applications in Science and Engineering* (J. H. Dshalalow, ed.), pp. 321–338, CRC Press, 1997.
- [3] L. Kosten, “Stochastic theory of data handling systems with groups of multiple sources,” *Performance of Computer Communication Systems*, pp. 321–331, 1984.
- [4] N. Akar and K. Sohraby, “Infinite- and finite-buffer markov fluid queues: A unified analysis,” *Journal of Applied Probability*, vol. 41, no. 2, pp. pp. 557–569, 2004.
- [5] A. da Silva Soares and G. Latouche, “Fluid queues with level dependent evolution,” *European Journal of Operational Research*, vol. 196, no. 3, pp. 1041 – 1048, 2009.
- [6] N. G. Bean and M. M. O’Reilly, “Performance measures of a multi-layer markovian fluid model,” *Annals of Operations Research*, vol. 160, pp. 99–120, April 2008.
- [7] G. Horváth and B. V. Houdt, “A Multi-layer Fluid Queue with Boundary Phase Transitions and Its Application to the Analysis of Multi-type Queues with General Customer Impatience,” in *Quantitative Evaluation of Systems (QEST), 2012 Ninth International Conference on*, pp. 23–32, sept. 2012.

- [8] A. Badescu, S. Drekić, and D. Landriault, "On the analysis of a multi-threshold Markovian risk model," *Scandinavian Actuarial Journal*, vol. 2007, no. 4, pp. 248–260, 2007.
- [9] M. Mandjes, D. Mitra, and W. Scheinhardt, "Models of network access using feedback fluid queues," *Queueing Syst. Theory Appl.*, vol. 44, no. 4, pp. 2989–3002, 2003.
- [10] H. E. Kankaya and N. Akar, "Solving multi-regime feedback fluid queues," *Stochastic Models*, vol. 24, pp. 425–450, July 2008.
- [11] W. Scheinhardt, N. van Foreest, and M. Mandjes, "Continuous feedback fluid queues," *Operations Research Letters*, vol. 33, no. 6, pp. 551–559, 2005.
- [12] W. Scheinhardt, N. van Foreest, and M. Mandjes, "Continuous feedback fluid queues," Technical Memorandum 1686, University of Twente, Department of Applied Mathematics, Enschede, The Netherlands, August 2003. Available at <http://www.math.utwente.nl/publications/>.
- [13] R. German, M. Gribaudo, G. Horváth, and M. Telek, "Stationary analysis of FSPNs with mutually dependent discrete and continuous parts," *Petri Nets and Performance Models, IEEE International Workshop on*, 2003.
- [14] M. Gribaudo and M. Telek, "Stationary analysis of fluid level dependent bounded fluid models," *Perform. Eval.*, vol. 65, pp. 241–261, March 2008.
- [15] G. H. Golub and C. F. van Loan, *Matrix Computations*. The Johns Hopkins University Press, 1996.
- [16] T. Dzial, L. Breuer, A. da Silva Soares, G. Latouche, and M.-A. Remiche, "Fluid queues to solve jump processes," *Performance Evaluation*, vol. 62, pp. 132–146, 2005.
- [17] M. F. Neuts, *Structured Stochastic Matrices of M/G/1 Type and Their Applications*. Marcel Dekker, N.Y., 1989.
- [18] D. M. Lucantoni, K. S. Meier-Hellstern, and M. F. Neuts, "A single server queue with server vacations and a class of nonrenewal arrival processes," *Adv. Applied Prob.*, vol. 22, pp. 676–705, 1990.

- [19] D. Perry and S. Asmussen, "Rejection rules in the M/G/1 queue," *Queueing Systems*, vol. 19, pp. 105–130, 1995.
- [20] R. Bekker, S. C. Borst, O. J. Boxma, and O. Kella, "Queues with workload-dependent arrival and service rates," *Queueing Systems*, vol. 46, pp. 537–556, 2004.
- [21] D. Perry, W. Stadje, and S. Zacks, "The M/G/1 queue with finite workload capacity," *Queueing Systems*, vol. 39, pp. 7–22, 2001.
- [22] R. Bekker, "Finite-buffer queues with workload-dependent service and arrival rates," *Queueing Systems*, vol. 50, pp. 231–253, 2005.
- [23] V. Sharma and J. T. Virtamo, "A finite buffer queue," in *Global Telecommunications Conference, 1999. GLOBECOM '99*, pp. 1053–1057, 1999.
- [24] V. Sharma and J. T. Virtamo, "A finite buffer queue with priorities," *Performance Evaluation*, vol. 47, pp. 1–22, 2002.
- [25] B. V. Houdt, "Analysis of the adaptive MMAP[K]/PH[K]/1 queue: A multi-type queue with adaptive arrivals and general impatience," *European Journal of Operational Research*, vol. 220, no. 3, pp. 695–704, 2012.
- [26] C. Qiao and M. Yoo, "Optical burst switching (OBS)-A new paradigm for an optical internet," *J. High Speed Networks*, vol. 8, no. 1, pp. 69–84, 1999.
- [27] J. Y. Wei and R. I. McFarland, "Just-In-Time signaling for WDM optical burst switching networks," *J. Lightwave Technol.*, vol. 18, p. 2019, December 2000.
- [28] M. Yoo and C. Qiao, "Just-Enough-Time (JET): A high speed protocol for bursty traffic in optical networks," in *IEEE/LEOS Technologies for a Global Information Infrastructure*, pp. 26–27, August 1997.
- [29] J. Turner, "Terabit burst switching," *J. High Speed Networks*, vol. 8, pp. 3–16, 1999.
- [30] M. Yoo and C. Qiao, "Supporting multiple classes of services in IP over WDM networks," in *Global Telecommunications Conference, 1999. GLOBECOM '99*, vol. 1B, pp. 1023–1027 vol. 1b, 1999.

- [31] V. M. Vokkarane, J. P. Jue, and S. Sitaraman, "Burst segmentation: an approach for reducing packet loss in optical burst switched networks," in *Communications, 2002. ICC 2002. IEEE International Conference on*, vol. 5, pp. 2673–2677 vol.5, May 2002.
- [32] V. M. Vokkarane and J. P. Jue, "Prioritized routing and burst segmentation for QoS in optical burst-switched networks," in *Optical Fiber Communication Conference and Exhibit, 2002. OFC 2002*, pp. 221–222, March 2002.
- [33] J. Phuritatkul, Y. Ji, and Y. Zhang, "Blocking probability of a preemption-based bandwidth-allocation scheme for service differentiation in OBS networks," *Journal of Lightwave Technology*, vol. 24, pp. 2986–2993, August 2006.
- [34] R. A. Barry and P. A. Humblet, "Models of blocking probability in all-optical networks with and without wavelength changers," in *INFOCOM '95. Fourteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Bringing Information to People. Proceedings. IEEE*, pp. 402–412 vol.2, April 1995.
- [35] I. Chlamtac, A. Fumagalli, L. Kazovsky, P. Melman, W. Nelson, P. Poggiolini, M. Cerisola, A. Choudhury, T. Fong, R. Hofmeister, C.-L. Lu, A. Mekittikul, I. Sabido, D.J.M., C.-J. Suh, and E. Wong, "CORD: contention resolution by delay lines," *Selected Areas in Communications, IEEE Journal on*, vol. 14, pp. 1014–1029, June 1996.
- [36] N. Barakat and E. Sargent, "An accurate model for evaluating blocking probabilities in multi-class OBS systems," *Communications Letters, IEEE*, vol. 8, pp. 119–121, February 2004.
- [37] D. Morató, M. Izal, J. Aracil, E. Magaña, and J. Miqueleiz, "Blocking time analysis of obs routers with arbitrary burst size distribution," in *Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE*, vol. 5, pp. 2488–2492 vol.5, December 2003.
- [38] H. L. Vu and M. Zukerman, "Blocking probability for priority classes in optical burst switching networks," *Communications Letters, IEEE*, vol. 6, pp. 214–216, May 2002.

- [39] K. Dolzer, C. Gauger, J. Späth, and B. Stefan, "Evaluation of reservation mechanisms for optical burst switching," *AEU - International Journal of Electronics and Communications*, vol. 55, no. 1, pp. 18–26, 2001.
- [40] J. Teng and G. N. Rouskas, "A comparison of the JIT, JET, and horizon wavelength reservation schemes on a single OBS node," in *In Proc. of the First International Workshop on Optical Burst Switching*, p. 2003, 2003.
- [41] A. Kaheel, H. Alnuweiri, and F. Gebali, "A new analytical model for computing blocking probability in optical burst switching networks," *Selected Areas in Communications, IEEE Journal on*, vol. 24, pp. 120–128, December 2006.
- [42] J. Hernandez, J. Aracil, L. de Pedro, and P. Reviriego, "Analysis of blocking probability of data bursts with continuous-time variable offsets in single-wavelength obs switches," *Lightwave Technology, Journal of*, vol. 26, pp. 1559–1568, June 2008.
- [43] H. Kankaya and N. Akar, "Exact analysis of offset-based service differentiation in single-channel multi-class OBS," *Communications Letters, IEEE*, vol. 13, pp. 148–150, February 2009.
- [44] ANSI/IEEE, *802.11: Wireless LAN Medium Access Protocol (MAC) and Physical Layer (PHY) Specifications*. IEEE, 2007.
- [45] ANSI/IEEE, *802.11a: Wireless LAN Medium Access Protocol (MAC) and Physical Layer (PHY) Specifications*. IEEE, 1999.
- [46] ANSI/IEEE, *802.11g: Wireless LAN Medium Access Protocol (MAC) and Physical Layer (PHY) Specifications*. IEEE, 2003.
- [47] Y. Xiao and J. Rosdahl, "Throughput and delay limits of IEEE 802.11," *IEEE Communications Letters*, vol. 6, no. 8, pp. 355–357, 2002.
- [48] T. Paul and T. Ogunfunmi, "Wireless LAN comes of age: Understanding the IEEE 802.11n amendment," *IEEE Circuits and Systems Magazine*, vol. 8, no. 1, pp. 28–54, 2008.

- [49] D. Skordoulis, Q. Ni, H.-H. Chen, A. Stephens, C. Liu, and A. Jamalipour, "IEEE 802.11n MAC frame aggregation mechanisms for next-generation high-throughput WLANs," *IEEE Wireless Communications*, vol. 15, no. 1, pp. 40–47, 2008.
- [50] ANSI/IEEE, *802.11b: Wireless LAN Medium Access Protocol (MAC) and Physical Layer (PHY) Specifications*. IEEE, 1999.
- [51] D. Qiao, S. Choi, and K. Shin, "Goodput analysis and link adaptation for IEEE 802.11a wireless LANs," *IEEE Transactions on Mobile Computing*, vol. 1, pp. 278–292, Oct. 2002.
- [52] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance anomaly of 802.11b," *Proc. IEEE INFOCOM'03*, April 2003.
- [53] G. Tan and J. Guttag, "Long-term time-share guarantees are necessary for wireless LANs," in *In Proceedings of the SIGOPS European Workshop*, 2004.
- [54] G. Tan and J. Guttag, "Time-based fairness improves performance in multi-rate WLANs," in *USENIX*, 2004.
- [55] A. Checco and D. Leith, "Proportional fairness in 802.11 wireless LANs," *Communications Letters, IEEE*, vol. 15, no. 8, pp. 807–809, 2011.
- [56] B. Li and R. Battiti, "Performance analysis of an enhanced IEEE 802.11 distributed coordination function supporting service differentiation," in *Quality for All, 4th COST 263 International Workshop on Quality of Future, Internet Services, QoFIS 2003, Stockholm, Sweden, October 1-2, 2003, Proceedings* (G. Karlsson and M. I. Smirnov, eds.), vol. 2811 of *Lecture Notes in Computer Science*, pp. 152–161, Springer, 2003.
- [57] H. Kim, S. Yun, I. Kang, and S. Bahk, "Resolving 802.11 performance anomalies through QoS differentiation," *IEEE Communications Letters*, vol. 9, pp. 555–557, July 2005.

- [58] I. Tinnirello, G. Bianchi, and L. Scalia, "Performance evaluation of differentiated access mechanisms effectiveness in 802.11 networks," in *Global Telecommunications Conference, 2004. GLOBECOM '04. IEEE*, vol. 5, pp. 3007 – 3011 Vol.5, nov.-3 dec. 2004.
- [59] V. A. Siris and G. Stamatakis, "Optimal CWmin selection for achieving proportional fairness in multi-rate 802.11e WLANs: test-bed implementation and evaluation," in *Proceedings of the 1st international workshop on Wireless network testbeds, experimental evaluation & characterization*, WiNTECH '06, (New York, NY, USA), pp. 41–48, ACM, 2006.
- [60] Y. Chetoui and N. Bouabdallah, "Adjustment mechanism for the IEEE 802.11 contention window: An efficient bandwidth sharing scheme," *Computer Communications*, vol. 30, pp. 2686–2695, September 2007.
- [61] C.-T. Chou, K. Shin, and N. Sai Shankar, "Contention-based airtime usage control in multirate IEEE 802.11 wireless LANs," *Networking, IEEE/ACM Transactions on*, vol. 14, no. 6, pp. 1179–1192, 2006.
- [62] P. Lin, W. i Chou, and T. Lin, "Achieving airtime fairness of delay-sensitive applications in multirate ieee 802.11 wireless lans," *Communications Magazine, IEEE*, vol. 49, no. 9, pp. 169–175, 2011.
- [63] T. Joshi, A. Mukherjee, Y. Yoo, and D. P. Agrawal, "Airtime fairness for IEEE 802.11 multirate networks," *IEEE Transactions on Mobile Computing*, vol. 7, pp. 513–527, Apr. 2008.
- [64] M. Heusse, F. Rousseau, R. Guillier, and A. Duda, "Idle Sense: An optimal access method for high throughput and fairness in rate diverse wireless LANs," *SIGCOMM'05*, August 2005.
- [65] L. Iannone and S. Fdida, "SDT. 11b: Un schéma à division de temps pour éviter l'anomalie de la couche MAC 802.11b," in *Colloque Francophone sur l'Ingénierie des Protocoles (CFIP'05)*, (Bordeaux, France), March 2005.
- [66] J. Dunn, M. Neufeld, A. Sheth, D. Grunwald, and J. Bennett, "A practical cross-layer mechanism for fairness in 802.11 networks," *Mob. Netw. Appl.*, vol. 11, no. 1, pp. 37–45, 2006.

- [67] T. Razafindralambo, I. G. Lassous, L. Iannone, and S. Fdida, "Dynamic packet aggregation to solve performance anomaly in 802.11 wireless networks," in *MSWiM '06: Proceedings of the 9th ACM international symposium on Modeling analysis and simulation of wireless and mobile systems*, pp. 247–254, 2006.
- [68] Y. Lin and V. Wong, "Frame aggregation and optimal frame size adaptation for IEEE 802.11n WLANs," in *Global Telecommunications Conference, IEEE GLOBECOM '06*, Nov. 2006.
- [69] M. Ergen and P. Varaiya, "Formulation of distributed coordination function of IEEE 802.11 for asynchronous networks: Mixed data rate and packet size," *Vehicle Technology, IEEE Transactions on*, vol. 57, no. 1, pp. 436–447, 2008.
- [70] I. Tinnirello and S. Choi, "Temporal fairness provisioning in multi-rate contention-based 802.11e WLANs," in *Proceedings of the Sixth IEEE International Symposium on World of Wireless Mobile and Multimedia Networks, WOWMOM '05*, (Washington, DC, USA), pp. 220–230, 2005.
- [71] I. Inan, F. Keceli, and E. Ayanoglu, "Analysis of the 802.11e enhanced distributed channel access function," *IEEE Transactions on Communications*, vol. 57, no. 6, pp. 1753–1764, 2009.
- [72] K. Kashibuchi, Y. Nemoto, and N. Kato, "Mitigating performance anomaly of TFRC in multi-rate IEEE 802.11 wireless LANs," *Proc. IEEE GLOBECOM'09*, 2009.
- [73] K. Kashibuchi, T. Taleb, Y. Nemoto, and N. Kato, "Channel occupancy time based TCP rate control for IEEE 802.11 DCF," *IWCMC'08*, August 2008.
- [74] C.-W. Huang, M. Loiacono, J. Rosca, and J.-N. Hwang, "Airtime fair distributed cross-layer congestion control for real-time video over WLAN," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 19, no. 8, pp. 1158–1168, 2009.
- [75] F. Keceli, I. Inan, and E. Ayanoglu, "Weighted fair uplink/downlink access provisioning in IEEE 802.11e WLANs," in *Proceedings of the ICC*, pp. 2473–2479, 2008.

- [76] W.-S. Lim, D.-W. Kim, and Y.-J. Suh, "Achieving fairness between uplink and downlink flows in error-prone WLANs," *Communications Letters, IEEE*, vol. 15, pp. 822–824, Aug. 2011.
- [77] C. Wang, H.-K. Lo, and S.-H. Fang, "Fairness analysis of throughput and delay in WLAN environments with channel diversities," *EURASIP Journal on Wireless Communications and Networking*, 2011.
- [78] J. Wilkinson, *The Algebraic Eigenvalue Problem*. Clarendon Press, Oxford, 1965.
- [79] H. E. Kankaya, *Performance Evaluation of Telecommunication Networks Based on Multi-Regime Fluid Queues*. PhD thesis, Bilkent University, July 2009.
- [80] B. Kågström, B. K. Ågstr, and P. Van Dooren, "A generalized state-space approach for the additive decomposition of a transfer matrix," 1992.
- [81] D. Mitra, "Stochastic theory of a fluid model of producers and consumers coupled by a buffer," *Advances in Applied Probability*, vol. 20, no. 3, pp. pp. 646–676, 1988.
- [82] P. M. V. Dooren, "Numerical linear algebra for signals, systems, and control." Draft notes prepared for the Graduate School in Systems and Control, Available at <http://perso.uclouvain.be/paul.vandooren/PVDnotes.pdf>, accessed Dec. 30, 2013, 2003.
- [83] M. Telek and M. Vecsei, "Finite queues at the limit of saturation," in *Quantitative Evaluation of Systems (QEST), 2012 Ninth International Conference on*, pp. 33–42, 2012.
- [84] O. Boxma, H. Kaspi, O. Kella, and D. Perry, "On/off storage systems with state-dependent input, output, and switching rates," *Probab. Eng. Inf. Sci.*, vol. 19, pp. 1–14, Jan. 2005.
- [85] P. J. Brockwell, S. I. Resnick, and R. L. Tweedie, "Storage processes with general release rule and additive inputs," *Adv. Appl. Prob.*, vol. 14, pp. 392–433, 1982.
- [86] M. F. Neuts, *Matrix-geometric Solutions in Stochastic Models: An Algorithmic Approach*. Baltimore, MD: Johns Hopkins University Press, 1981.

- [87] G. Latouche and V. Ramaswami, *Introduction to Matrix Analytical Methods in Stochastic Modeling*. ASA-SIAM Series on Statistics and Applied Probability, 2002.
- [88] H. C. Tijms, *Stochastic Modelling and Analysis: A Computational Approach*. John Wiley and Sons, 1986.
- [89] C. Moler and C. Van Loan, "Nineteen dubious ways to compute the exponential of a matrix," *SIAM Review*, vol. 20, no. 4, pp. 801–836, 1978.
- [90] C. Moler and C. Van Loan, "Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later," *SIAM Review*, vol. 45, no. 1, pp. 3–49, 2003.
- [91] H. Kankaya and N. Akar, "Exact analysis of single-wavelength optical buffers with feedback markov fluid queues," *Optical Communications and Networking, IEEE/OSA Journal of*, vol. 1, no. 6, pp. 530–542, 2009.
- [92] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 533–547, 2000.
- [93] E. Elhag and M. Othman, "Adaptive contention window scheme for WLANs," *The International Arab Journal of Information Technology*, vol. 4, pp. 313–321, October 2007.