Helsinki University of Technology Department of Communications and Networking
Teknillinen korkeakoulu Tietoliikenne- ja tietoverkkotekniikan laitos
Espoo 2008 Report 3/2008

FLOW-LEVEL PERFORMANCE ANALYSIS OF DATA NETWORKS USING PROCESSOR SHARING MODELS

Juha Leino

Dissertation for the degree of Doctor of Science in Technology to be presented with due permission for public examination and debate in Auditorium S1 at Helsinki University of Technology (Espoo, Finland) on the 9th of May, 2008, at 12 o'clock noon.

Helsinki University of Technology
Faculty of Electronics, Communications and Automation
Department of Communications and Networking

Teknillinen korkeakoulu
Elektroniikan, tietoliikenteen ja automaation tiedekunta
Tietoliikenne- ja tietoverkkotekniikan laitos

Distribution:
Helsinki University of Technology
Department of Communications and Networking
P.O.Box 3000
FI-02015 TKK
Tel. +358-9-451 5300
Fax +358-9-451 2474

ISBN 978-951-22-9300-1
ISBN 978-951-22-9301-8 (pdf)
ISSN 1797-478X
ISSN 1797-4798 (pdf)

Multiprint Oy
Espoo 2008

TEKNILLINEN KORKEAKOULU
TEKNISKA HÖGSKOLAN
HELSINKI UNIVERSITY OF TECHNOLOGY

HELSINKI UNIVERSITY OF TECHNOLOGY P.O.BOX 1000, FI-02015 TKK http://www.tkk.fi/	ABSTRACT OF DOCTORAL DISSERTATION

Author: Juha Leino

Name of the dissertation
Flow-Level Performance Analysis of Data Networks Using Processor Sharing Models

Date of manuscript 10th of December 2007	Date of the dissertation 9th of May 2008
☐ Monograph	☒ Article dissertation (summary + original articles)

Faculty	Faculty of Electronics, Communications and Automation
Department	Department of Communications and Networking
Field of research	Teletraffic theory
Opponent	Professor Rob van der Mei (Vrije Universiteit, The Netherlands)
Supervisor	Professor Jorma Virtamo (Helsinki University of Technology)

Abstract

Most telecommunication systems are dynamic in nature. The state of the network changes constantly as new transmissions appear and depart. In order to capture the behavior of such systems and to realistically evaluate their performance, it is essential to use dynamic models in the analysis. In this thesis, we model and analyze networks carrying elastic data traffic at flow level using stochastic queueing systems. We develop performance analysis methodology, as well as model and analyze example systems.

The exact analysis of stochastic models is difficult and usually becomes computationally intractable when the size of the network increases, and hence efficient approximative methods are needed. In this thesis, we use two performance approximation methods. Value extrapolation is a novel approximative method developed during this work and based on the theory of Markov decision processes. It can be used to approximate the performance measures of Markov processes. When applied to queueing systems, value extrapolation makes possible heavy state space truncation while providing accurate results without significant computational penalties. Balanced fairness is a capacity allocation scheme recently introduced by Bonald and Proutière that simplifies performance analysis and requires less restrictive assumptions about the traffic than other capacity allocation schemes. We introduce an approximation method based on balanced fairness and the Monte Carlo method for evaluating large sums that can be used to estimate the performance of systems of moderate size with low or medium loads.

The performance analysis methods are applied in two settings: load balancing in fixed networks and the analysis of wireless networks. The aim of load balancing is to divide the traffic load efficiently between the network resources in order to improve the performance. On the basis of the insensitivity results of Bonald and Proutière, we study both packet- and flow-level balancing in fixed data networks. We also study load balancing between multiple parallel discriminatory processor sharing queues and compare different balancing policies.

In the final part of the thesis, we analyze the performance of wireless networks carrying elastic data traffic. Wireless networks are gaining more and more popularity, as their advantages, such as easier deployment and mobility, outweigh their downsides. First, we discuss a simple cellular network with link adaptation consisting of two base stations and customers located on a line between them. We model the system and analyze the performance using different capacity allocation policies. Wireless multihop networks are analyzed using two different MAC schemes. On the basis of earlier work by Penttinen et al., we analyze the performance of networks using the STDMA MAC protocol. We also study multihop networks with random access, assuming that the transmission probabilities can be adapted upon flow arrivals and departures. We compare the throughput behavior of flow-optimized random access against the throughput obtained by optimal scheduling assuming balanced fairness capacity allocation.

Keywords	data networks, elastic traffic, flow-level performance analysis, Markov decision processes, balanced fairness, processor sharing, load balancing		
ISBN (print)	978-951-22-9300-1	ISSN (print)	1797-478X
ISBN (pdf)	978-951-22-9301-8	ISSN (pdf)	1797-4798
Number of pages	84 p.		
Publisher	Department of Communications and Networking / Helsinki University of Technology		
Print distribution			
☒ The dissertation can be read at http://lib.tkk.fi/Diss/			

TEKNILLINEN KORKEAKOULU
TEKNISKA HÖGSKOLAN
HELSINKI UNIVERSITY OF TECHNOLOGY

TEKNILLINEN KORKEAKOULU PL 1000, 02015 TKK http://www.tkk.fi/	VÄITÖSKIRJAN TIIVISTELMÄ
Tekijä: Juha Leino	
Väitöskirjan nimi Tietoverkkojen vuotason suorituskykyanalyysi käyttäen palvelimenjakomalleja	
Käsikirjoitus jätetty 10. joulukuuta 2007	Väitöstilaisuus 9. toukokuuta 2008
☐ Monografia	☒ Yhdistelmäväitöskirja (yhteenveto + erillisartikkelit)

Tiedekunta	Elektroniikan, tietoliikenteen ja automaation tiedekunta
Laitos	Tietoliikenne- ja tietoverkkotekniikan laitos
Tutkimusala	Teleliikenneteoria
Vastaväittäjä	Professori Rob van der Mei (Vrije Universiteit, Alankomaat)
Valvoja	Professori Jorma Virtamo (Teknillinen korkeakoulu)

Tiivistelmä

Useimmat tietoliikennejärjestelmät ovat luonteeltaan dynaamisia. Järjestelmän tila muuttuu jatkuvasti, kun lähetyksiä alkaa ja loppuu. Jotta tällaisten järjestelmien käyttäytyminen saataisiin kuvattua ja niiden suorituskyky analysoitua realistisesti, on niiden analyysissa käytettävä dynaamisia malleja. Tässä työssä mallinnamme ja analysoimme elastista dataliikennettä kuljettavia tietoverkkoja vuotasolla käyttäen stokastisia jonomalleja. Kehitämme suorituskykyanalyysimenetelmiä sekä mallinnamme ja analysoimme esimerkkijärjestelmiä.

Stokastisten mallien tarkka analyysi on vaikeaa ja muuttuu laskennallisesti vaativaksi järjestelmän koon kasvaessa, joten niiden analysoimiseksi tarvitaan tehokkaita likimääräisiä menetelmiä. Tässä työssä käytämme kahta likimääräistä menetelmää. Arvoekstrapolaatio on työssä kehitetty likimääräinen menetelmä, joka perustuu Markovin päätösprosessien teoriaan. Sitä voidaan käyttää approksimoimaan Markovin prosessien suorituskykysuureita. Jonosysteemeihin sovellettaessa se mahdollistaa tarkkojen tulosten saavuttamisen, vaikka tila-avaruutta olisi supistettu huomattavasti ja siten laskentatyö pidetty rajallisena. Tasapainotettu reiluus (balanced fairness) on Bonaldin ja Proutièren hiljattain esittelemä kapasiteetinjakomenetelmä, joka yksinkertaistaa suorituskykyanalyysia ja vaatii muita menetelmiä vähemmän rajoittavia oletuksia koskien järjestelmän liikennettä. Esittelemme myös uuden likimääräisen menetelmän, joka perustuu tasapainotettuun reiluuteen ja Monte Carlo -menetelmään, ja jota voidaan käyttää kohtuullisten kokoisten järjestelmien analysoimiseen matalilla ja keskikokoisillla kuormilla.

Suorituskykyanalyysimenetelmiä sovelletaan kahdessa yhteydessä: kuormantasauksessa kiinteissä verkoissa ja langattomien verkkojen analyysissä. Kuormantasauksen tavoite on jakaa liikennekuorma tehokkaasti verkon resurssien kesken suorituskyvyn parantamiseksi. Työssä tutkitaan sekä paketti- että vuotason kuormantasausta kiinteissä dataverkoissa hyödyntäen Bonaldin ja Proutièren insensitiivisyystuloksia. Lisäksi tutkitaan kuormantasausta usean rinnakkaisen DPS-jonon kesken sekä vertaillaan eri kuormantasauspolitiikkoja.

Työn viimeisessä osassa tutkitaan elastista dataliikennettä kuljettavien langattomien verkkojen suorituskykyä. Langattomien verkkojen suosio kasvaa jatkuvasti niiden etujen, kuten helpomman käyttöönoton ja käyttäjien liikkuvuuden mahdollistamisen, painaessa haittapuolia enemmän. Ensimmäiseksi tutkimme yksinkertaista, linkkiadaptaatiota käyttävää kahden tukiaseman soluverkkoa, jonka käyttäjät sijaitsevat tukiasemien välisellä suoralla. Mallinnamme järjestelmää ja analysoimme sen suorituskykyä eri kapasiteetinjakomenetelmiä käytettäessä. Lisäksi tutkimme langattomia monihyppyisiä verkkoja käyttäen kahta eri MAC-menetelmää. Penttisen et al.:in aikaisempaan työhön pohjautuen analysoimme STDMA MAC-protokollaa käyttäviä verkkoja. Tutkimme myös monihyppyisiä verkkoja, jotka käyttävät satunnaistettua kanavalle pääsyä, olettaen että lähetystodennäköisyydet pystytään mukauttamaan voiden alku- ja loppuhetkillä. Käyttäen tasapainotettua reiluutta kapasiteetinjakomenetelmänä tutkimme läpäisyä tällaisessa verkossa verrattuna järjestelmään, jossa kanavalle pääsy on optimoitu.

Avainsanat	dataliikenne, elastinen liikenne, vuotason suorituskykyanalyysi, Markovin päätösprosessit, tasapainotettu reiluus, palvelimenjakomalli, kuormantasaus		
ISBN (painettu)	978-951-22-9300-1	ISSN (painettu)	1797-478X
ISBN (pdf)	978-951-22-9301-8	ISSN (pdf)	1797-4798
Sivumäärä	84 s.		
Julkaisija	Tietoliikenne- ja tietoverkkotekniikan laitos / Teknillinen korkeakoulu		
Painetun väitöskirjan jakelu			
☒ Luettavissa verkossa osoitteessa http://lib.tkk.fi/Diss/			

PREFACE

This dissertation is the result of research started in 2003 in the Networking Laboratory of Helsinki University of Technology. The work has been carried out in the NAPS, FIT, and Fancy projects funded by the Academy of Finland as well as in the IRoNet project funded by TEKES and Finnish telecommunications industry. The personal scholarships received from Nokia foundation are gratefully acknowledged.

First, I wish to thank my supervisor, Prof. Jorma Virtamo. Without his enthusiastic guidance the thesis would not have been possible. I would also like to thank my co-author, Dr. Aleksi Penttinen, with whom working has been both inspiring and fun. I am grateful to the pre-examiners, Dr. Konstantin Avrachenkov and Dr. Matthieu Jonckheere, whose comments and suggestions improved the thesis. Also Dr. Samuli Aalto provided insightful comments on the manuscript. The past and present colleagues have made the years in the Networking Laboratory pleasant. In particular, members of the teletraffic group and floorball team have made work feel less like work.

I thank my family for the support during my studies. I would also like to thank my friends for providing much-needed counterbalance to my work.

CONTENTS

LIST OF PUBLICATIONS

[1] Juha Leino and Jorma Virtamo. An approximative method for calculating performance measures of Markov processes. In *Proceedings of Valuetools'06*, Pisa, Italy, 2006.

[2] Juha Leino, Aleksi Penttinen, and Jorma Virtamo. Approximating flow throughput in complex data networks. In *Proceedings of 20th International Teletraffic Congress (ITC-20)*, pages 422–433, Ottawa, Canada, 2007.

[3] Juha Leino. Approximating optimal load balancing policy in discriminatory processor sharing systems. In *Proceedings of SMCtools 2007*, Nantes, France, 2007.

[4] Juha Leino and Jorma Virtamo. Determining the moments of queue-length distribution of discriminatory processor-sharing systems with phase-type service requirements. In *Proceedings of NGI 2007*, pages 205–208, Trondheim, Norway, 2007.

[5] Juha Leino and Jorma Virtamo. Insensitive load balancing in data networks. *Computer Networks*, 50(8):1059–1068, 2006.

[6] Juha Leino and Jorma Virtamo. Insensitive traffic splitting in data networks. In *Proceedings of 19th International Teletraffic Congress (ITC-19)*, pages 1355–1364, Beijing, China, 2005.

[7] Juha Leino, Aleksi Penttinen, and Jorma Virtamo. Flow level performance analysis of wireless data networks: A case study. In *Proceedings of IEEE ICC 2006*, Istanbul, Turkey, 2006.

[8] Juha Leino, Aleksi Penttinen, and Jorma Virtamo. Flow-optimized random access for wireless multihop networks. In *Proceedings of ACM/IEEE MSWiM 2007*, Chania, Crete Island, 2007.

1 INTRODUCTION

1.1 Flow-level Modeling of Elastic Traffic

Mathematical modeling has been used in the performance analysis of telecommunication systems since the beginning of the 20th century [Erl09, Erl17]. Most telecommunication systems are dynamic in nature, i.e. new calls, transmissions, or packets arrive in and depart from the system. Many characteristics of such systems cannot be captured with static models, instead, stochastic models are needed.

Telecommunications traffic can be modeled at different levels [Eva96]. For instance, the dynamics of a single buffer can be analyzed at packet level [CH98]. At the other end of the spectrum, aggregated traffic in high-capacity core networks is often to be assumed constant when dimensioning or routing is considered [PM04]. The more detailed the traffic model is, the more complex the analysis and the smaller the instances that can be studied. Depending on the system and phenomena being considered, a suitable level of detail is needed in the analysis.

Network traffic can be broadly classified into "stream" or "elastic" traffic [Rob00]. Stream traffic, e.g. real-time voice or video, has a data rate that is usually limited by an upper and lower bound and the duration of a transfer does not depend on the available network capacity. Elastic traffic consists of flows, e.g. file transfers using the TCP protocol, which are characterized by the size of the transferred object. An elastic transfer typically uses all the available capacity. The duration of the transfer depends on the available capacity, which depends on how the network capacity is allocated among the competing flows and on the amount of concurrent traffic.

In this thesis we study networks carrying elastic data traffic. When the quality of service of elastic traffic is evaluated, the most important metric is the duration of the transfer of a flow. Typically, a user is interested in when a web page or FTP transfer is completed, but the transmission speed at any given point during the transfer is not significant. Packet-level characteristics such as packet delay or jitter are not relevant, and hence a flow-level model can be used to get meaningful results concerning data networks carrying elastic traffic.

If the network traffic is elastic, packet-level phenomena can be ignored and the network can be modeled at flow level (see, e.g., [Rob04]). New flows arrive stochastically and the flow sizes follow some random distribution. Network capacity is shared among the active flows according to some capacity allocation policy. If the flows are large when compared to individual packets, it can be assumed that the bandwidth resources of the network are instantaneously reallocated among the flows every time the number of flows in the system changes, for instance by the TCP congestion control. Thus, between the epochs of a flow arrival or departure each flow receives a constant bandwidth. In this case, the network can be modeled as a server system that allocates capacity to the flows. Queueing happens at the edges of the network and the packet-level phenomena inside the network can be ignored. The policy by which the capacity is shared among concurrent flows, for example max-min-fairness or proportional fairness [KMT98], affects the behavior and performance of the system.

Recently, egalitarian processor sharing (EPS) models have been introduced as a new paradigm for the flow-level modeling of elastic traffic [MR00, NMM98, HLN97]. If elastic flows share a common bottleneck and the capacity allocation is fair, the system can be modeled as an EPS queue, which allows powerful theoretical results to be used (see, e.g., [Jai82, YY07]). In this thesis, we study two different processor sharing concepts that can be used in teletraffic analysis: discriminatory processor sharing (DPS) and balanced fairness (BF). DPS is a non-egalitarian processor sharing discipline which can be used to model a resource that is shared unevenly among the customers. Customers are categorized into classes and the share of capacity depends on the class. If elastic traffic is restricted by a single bottleneck and the traffic classes are not treated equally, DPS can be used in the modeling. The other scheme studied here, balanced fairness, is a new capacity allocation scheme recently introduced by Bonald and Proutière [BP03a]. In many ways, balanced fairness is the most natural extension of the single-resource EPS system to the multiclass, multiple-resources case. Similarly to an EPS queue, BF systems are insensitive to detailed traffic characteristics, thus allowing the assumptions needed in the modeling to be relaxed and more general results to be obtained. BF also inherits the linearity property of EPS; the conditional sojourn times are proportional to the actual job sizes. BF is more tractable than other capacity allocation policies and it can be used in the approximation of other schemes [BMPV06, Tim03].

1.2 Load Balancing in Data Networks

Internet traffic is traditionally routed using the shortest paths. While this minimizes the resource usage, it may lead to congestion if the traffic concentrates on some links. Traffic engineering, specifically load balancing, tries to utilize the resources better and improve the performance of the system.

Load balancing can be broadly categorized into static and dynamic balancing. Static balancing does not rely on current traffic measurements, but the routes are fixed on the basis of prior knowledge or expectations of traffic amounts. On the other hand, dynamic load balancing reacts to the load of the network and the routes are changed on the basis of current traffic measurements. Static load balancing is usually done manually, while dynamic routing is typically automated. Load balancing can be done on different time scales. Static load balancing does not react to the stochastic fluctuations of the traffic. Dynamic load balancing reacts more quickly, even during short-timed temporary peaks in traffic. Load balancing can be conducted using traditional routing protocols, e.g. by tuning link weights in the open shortest path first (OSPF) protocol [Moy98], or by using more recent technologies such as multiprotocol label switching (MPLS) [RVC01, RTF$^+$01].

In this thesis, we study load balancing in two settings using flow-level modeling. First, load balancing among multiple parallel discriminatory processor sharing servers is analyzed. DPS queues can be used in modeling a variety of systems; the most interesting application is the flow-level modeling of elastic traffic.

The second setting is applicable to more general data networks. Applying the balanced fairness concept, load balancing can be studied assuming that the balancing is executed either at flow or packet level. When packet-level balancing is considered, a flow can be divided dynamically among multiple routes, while flow-level balancing routes an arriving flow to a route which is used during the whole transmission. BF makes the system more tractable when compared to other capacity allocation policies. Load balancing under balanced fairness in insensitive to flow size distribution, thus allowing performance results to be achieved that do not depend on the detailed characteristics of the traffic.

1.3 Wireless Networks

An increasing amount of telecommunications traffic is nowadays carried over wireless networks. Wireless technologies for both data (e.g. WLAN networks [sta01]) and telephone (e.g. GSM networks [rgppG]) traffic have become widespread. Wireless networks make possible user mobility and lower the cost of deployment, as there is no need for cabling. A downside is the lower performance and higher complexity, as the common radio channel needs to be shared efficiently.

Efficient bandwidth utilization is essential when high-capacity wireless networks are being designed. Networks require mechanisms that control the use of the shared medium. Medium access control (MAC) mechanisms try to ensure that the bandwidth utilization is efficient. MAC protocols vary from the simple CSMA/CA scheme used in 802.11 WLAN networks [sta99] to the more recent OFDMA protocol used, e.g., in 802.16 WiMAX networks [es05].

The performance analysis of wireless networks is more difficult than when fixed networks are studied. The variety of wireless technologies is great and the characteristics of the network being analyzed have to be taken into account in the modeling. The shared nature of the radio channel makes the dependencies between concurrent transmissions complex. Wireless settings often involve user mobility, which complicates the analysis further. This complexity typically makes performance analysis computationally demanding.

In this thesis, we focus on wireless data networks. In particular, we assume that the network carries elastic traffic. The networks are studied in a dynamic, flow-level setting. New transmissions start and end randomly and, naturally, the performance at any given point in time varies accordingly. While in some applications a static traffic model may capture the essential behavior of the system, dynamic modeling usually gives a significantly better insight into the performance perceived by the users. In our analysis, we are interested in mean flow throughput or mean transmission time. While mobility is an important aspect in many wireless networks, we do not consider user mobility but assume the users to be stationary.

We study two types of networks. In cellular networks, transmissions occur only between base stations or access points and users. The best-known cellular networks are mobile phone networks, but WLAN networks too often have a cellular structure. Typically, the base stations are connected to a fixed core network and the (often mobile) users' traffic is relayed via the base stations.

Wireless multihop networks have recently been the subject of major interest [KM07, BCG05, KRD06]. Instead of relying on fixed infrastructure, the nodes relay each other's traffic. Such networks have many applications in situations where fixed infrastructure is not available, for example in military and emergency situations. Multihop networks can also be used in sensor networks [ASSC02, BPC+07] or in extending the range of networks with fixed infrastructure.

1.4 Contributions of the Thesis

In this thesis, we model and analyze networks carrying elastic data traffic. The networks are modeled in a dynamic setting, where new transmissions arrive and depart stochastically. The networks are modeled at flow level using stochastic queueing systems. We develop performance analysis methodology as well as model and analyze example systems.

We are interested in mean performance measures related to the flows. The exact analysis of stochastic models is difficult and usually becomes computationally intractable when the size of the network increases, and hence efficient approximative methods are needed. In this thesis, we introduce a novel method called value extrapolation for approximating the performance measures of Markov processes (Publication 1). We also use the balanced fairness capacity allocation scheme of Bonald and Proutière, which facilitates performance analysis [BP03a]. Specifically, the asymptotic throughput approach introduced in [BPV06] is used. In addition, we present a computational scheme based on balanced fairness and the Monte Carlo method that can be used to approximate flow throughput in networks of moderate size with low or medium loads (Publication 2).

The performance analysis methods are applied in two fields: load balancing applications in fixed networks and the analysis of wireless networks. We study load balancing among multiple parallel DPS servers and compare different load balancing policies (Publication 3). As a new theoretical result, we show that the relative values corresponding to the queue lengths of a DPS queue with Poissonian arrivals and Cox distributed service requirements are polynomial functions of the system state (Publication 4). We show that using the relative values of a single queue, a so-called first policy iteration policy that leads to good performance can easily be derived.

We also discuss load balancing in fixed networks carrying elastic traffic, executed either at packet or flow level (Publications 5 and 6). On the basis of the insensitivity results of Bonald and Proutière [BP03a], we derive linear programming formulations that can be used to determine optimal insensitive load balancing policies. The flow-level balancing is optimized both independently of the capacity allocation and jointly with it. In both cases, the network remains insensitive to flow size distribution, thus allowing more robust results to be achieved.

In the final part of the thesis, we analyze the performance of wireless networks carrying elastic data traffic. First, we discuss a simple cellular network with link adaptation consisting of two base stations and customers located on a line between them. We model the system and analyze the performance using different capacity allocation policies (Publication 7).

We also study wireless multihop networks using two different MAC schemes and balanced fairness capacity allocation. The systems are insensitive to detailed traffic characteristics, which is a desirable property for performance analysis and dimensioning purposes. On the basis of earlier work by Penttinen et al. in [PVJ06], we apply the asymptotic throughput analysis to multihop networks using the STDMA MAC protocol (Publication 2). We also study multihop networks with random access, assuming that the transmission probabilities can be adapted upon flow arrivals and departures (Publication 8). We derive the exact throughput in the two-class scenario. In the general network case, we present an algorithm for optimizing the transmission probabilities and compare the throughput behavior of flow-optimized random access against the throughput obtained by optimal scheduling.

1.5 Outline of the Thesis

In Chapter 2, we introduce the Markovian modeling setting used in this thesis. Specifically, we discuss discriminatory processor sharing models and the balanced fairness capacity allocation scheme.

In Chapter 3, we discuss performance analysis methods suitable for our setting. We introduce the relevant parts of the theory of Markov decision processes used in this thesis. We also present value extrapolation, a new approximative method based on the MDP theory developed during this work. We present an approximative method based on balanced fairness, throughput asymptotics, introduced by Bonald et al. in [BPV06]. Finally, we discuss a new computational scheme based on balanced fairness and the Monte Carlo method.

In Chapter 4, the performance analysis methods are applied to two load balancing settings. First we study load balancing among several parallel DPS queues. We analyze and compare the performance of different routing policies using value extrapolation. We also study insensitive load balancing in fixed networks, assuming either packet- or flow-level balancing.

In Chapter 5, the performance of wireless networks is analyzed. We study a simple cellular scenario in Section 5.3 and multihop networks in Sections 5.4 and 5.5. In the cellular setting, the performance of different capacity allocation schemes is compared with balanced fairness using value extrapolation. Balanced fairness has been applied to multihop STDMA networks in [PVJ06]. We apply asymptotic throughput analysis to this setting in Section 5.4. Finally, we model the flow-level performance of wireless multihop networks with time-slotted random access and apply asymptotic BF analysis.

2.1 Introduction

In this chapter, we provide background information as an introduction to our own contributions in the later chapters.

Queueing models are suitable for analyzing systems where some limited resources are used to perform certain tasks. The resources are usually referred to as servers and the tasks as customers. Each customer has a service requirement and, after receiving it, he departs from the system.

The basic queueing model consists of a server, queueing places, and customers as illustrated in Figure 2.1. The properties of a queue are usually denoted using Kendall's notation [Ken53]. Typically, the arrivals of customers are stochastic and the service requirements are assumed to follow some probability distribution, and hence the queue process is random in nature. Queueing discipline determines how the capacity is divided among the customers at any given time; for example, the first-in-first-out-discipline serves the customers in their order of arrival. In this thesis we use queueing models that utilize a processor sharing discipline, i.e. the capacity of the server is divided between all concurrent customers. If the sharing is equal, the queueing discipline is called egalitarian processor sharing (EPS). Often, processor sharing refers only to egalitarian processor sharing but in this thesis we use this term to refer to both EPS and non-egalitarian PS, in particular the so-called discriminatory PS (DPS).

Queueing models have been used in teletraffic analysis since Erlang's results concerning telephone systems in the early 20th century [Erl17]. A telephone link or exchange can be modeled with a queue with one or multiple servers, where each accepted call reserves a server until the call is finished. After the early experiments with packet-switched computer networks in the 1960s, queueing models were applied to study the systems. Networks of queues were introduced as a way to model the packets and the routers [Kle64]. Again, one server served one packet at a time.

Processor sharing queues were originally used in the modeling of computer systems, where an operating system schedules processing time among the processes using a round robin policy [Kle67]. The egalitarian processor sharing model is an idealization of the round robin policy, where each job receives an equal amount of service in turn. In addition to EPS models, more general models, such as discriminatory processor sharing [Kle67] and generalized processor sharing [DKS89], were introduced, allowing customer classes with different weights or priorities to receive different levels of

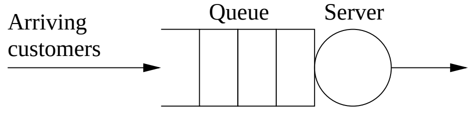


Figure 2.1: Basic queueing model with one server

service. Networks of queues were used to model systems in which processes or users use the resources of multiple servers or processors and a rich theory was developed during the 1970s [Kel76, BCMP75].

Queueing systems can be used in the modeling of data networks. The network acts as a server system and shares capacity among the flows. Recently, processor sharing queues have been used in the flow-level modeling of data networks [MR00, NMM98, HLN97]. If the users share a single capacity bottleneck in the network, the performance can be analyzed using a processor sharing queue. If the capacity allocation policy or congestion control mechanism works fairly, an EPS queue can be used in the modeling allowing the extensive theoretical results concerning EPS queues to be used [Jai82, YY07]. If the flows receive different shares of capacity, a single EPS queue is not sufficient, but other queueing disciplines or models are needed. Traffic differentiation can be either intentional, e.g., a quality-of-service mechanism giving preferential treatment to some flows, or unintentional, e.g., the RTT of a TCP flow affects the capacity share it receives in a link [AJK04].

In this thesis, we study two different processor sharing concepts that can be used in teletraffic analysis: discriminatory processor sharing and balanced fairness. DPS is a processor sharing discipline that can be used to model a resource that is shared unevenly among customers. The customers are categorized into classes and their share of capacity depends on the class. If elastic traffic is restricted by a single bottleneck and the traffic classes are not treated equally, DPS can be used in the modeling. The other scheme studied here, balanced fairness, is a recent capacity allocation scheme introduced by Bonald and Proutière [BP03a]. Based on the theory of queueing networks, it allows the performance of networks with several resources to be analyzed. In many ways, balanced fairness is the most natural extension of the single-resource EPS system to the multiclass, multiple-resources case. Contrary to other capacity allocation policies it makes the system more tractable, thus allowing flow-level performance to be analyzed. A clear distinction between the two approaches is that in DPS models, a queue models a concrete part of a network, e.g. a link or a router, while in the BF approach PS queues are related to individual traffic classes and do not relate to any physical component of the network.

It is well known that an M/G/1-PS queue with a Poissonian arrival process is insensitive, i.e. the steady state distribution does not depend on the service requirement distribution but only on the load of the system. DPS queues are not insensitive, in particular the mean queue length and state distribution depend on the service requirement distribution. Balanced fairness models are insensitive. The insensitivity to service requirement distribution makes the model more reasonable for applications in telecommunications. The insensitivity of BF is even stronger, allowing the assumption of Poisson flow arrivals to be relaxed and replaced by the assumption of Poissonian session arrivals, where each session consists of file transfers and think times [BP03a]. Internet traffic measurements have indicated Poissonian session arrivals [FP01] and heavy-tailed flow size distribution [HCMSS04], and hence the assumptions needed with DPS models do not fit real traffic well, while the assumptions of BF models are more reasonable.

2.2 Markov Modeling of Queueing Systems

In this thesis, Markov processes are used in the modeling and analysis of queueing systems. Let $X(t)$ be a continuous-time stochastic process describing a queueing process with state space $\mathcal{S}$. The number of states is denoted $|\mathcal{S}|$. The state of the process is denoted with vector $\mathbf{x} \in \mathcal{S}$. The transition intensity from state $\mathbf{x}$ to state $\mathbf{y}$ is denoted $q_{\mathbf{xy}}$. It should be noted, that Markov modeling does not restrict the system to Poissonian arrivals or exponentially distributed service requirements.

The steady state distribution of the process satisfies the global balance equations

$$\pi(\mathbf{x}) \sum_{\mathbf{y}} q_{\mathbf{xy}} = \sum_{\mathbf{y}} \pi(\mathbf{y}) q_{\mathbf{yx}} \quad \forall \mathbf{x}, \tag{2.1}$$

where $\pi(\mathbf{x})$ is the steady state probability of state $\mathbf{x}$. If the state space is infinite, state distribution can be solved only in some special cases. One way to deal with an infinite state space is to truncate it into a smaller state space $\tilde{\mathcal{S}} \subset \mathcal{S}$ in order to solve the set of equations (2.1). If the truncated state space is large enough, the probabilities of the truncated state space can be used to approximate the probabilities of the whole state space. Given the approximative state distribution, many performance measures can be approximated by summing over the states. The larger the truncated state space is, the more accurate the results.

2.3 Discriminatory Processor Sharing

Discriminatory processor sharing is a generalization of egalitarian processor sharing first introduced by Kleinrock in 1967 under the name priority processor sharing [Kle67]. Customers are categorized into customer classes and the service rate of a customer depends on its class. The customers of a given class residing concurrently in the system obtain an equal share of the service capacity, but the customers of different classes may receive different shares of the capacity according to class weights or priorities.

We study a DPS system with unit capacity and K customer classes. The state of the process is $\mathbf{x} = (x_1, \ldots, x_K)^{\mathrm{T}}$, where x_k is the number of class-k customers in the queue. The service rate of class k is

$$\phi_k(\mathbf{x}) = \frac{w_k x_k}{\sum_d w_d x_d}, \tag{2.2}$$

where w_k is the weight parameter of class k. Classes with higher weights receive better service at the server. If all the weights are equal, DPS reduces to the normal egalitarian processor sharing system.

Theoretical results concerning DPS queues are sparse when compared to EPS queues. A survey on analytical results for DPS systems was recently presented by Altman et al. [AAA06]. Two important performance measures, queue length and sojourn time distributions, have been studied in different settings. Rege and Sengupta presented an analysis that allows one to obtain the moments of the queue-length distribution as a solution to a system of linear equations assuming exponentially distributed service requirements [RS96]. The same approach was later generalized by van Kessel et al. assuming more general phase-type service requirement distributions [vKNB05]. Assuming exponentially distributed service requirements, the unconditional expected sojourn times may be solved from a system of linear equations as shown by Fayolle et al. in [FMI80]. Kim and Kim found that the higher moments are also solvable from linear simultaneous equations [KK04].

2.4 Balanced Fairness

In this section we provide a brief introduction to the concept of balanced fairness. First, we discuss the insensitivity results concerning queueing networks and then their applications to teletraffic analysis. For a more detailed presentation, see the original work by Bonald and Proutière, e.g., [BP03a] and [BP04].

Insensitivity in Processor Sharing Networks

We consider an open queueing network of N nodes. Let $X(t)$ be the continuous-time Markov process corresponding to the system. The state of the process is defined by the vector $\mathbf{x} = (x_1, \ldots, x_N)^{\mathrm{T}}$, where x_i is the number of customers at node i. The service rate $\phi_i(\mathbf{x})$ of node i may depend on the state of the network.

External customer arrival process at node i is Poissonian with intensity ν_i. Service requirements at node i are exponentially distributed with mean $1/\mu_i$. After receiving the requested service at node i, a customer continues to node j with probability p_{ij} and leaves the network with probability $p_i = 1 - \sum_j p_{ij}$. The total arrival rate λ_i at node i is defined by equations

$$\lambda_i = \nu_i + \sum_j \lambda_j p_{ji}, \quad j = 1, \ldots, N. \tag{2.3}$$

The load of node i is denoted $\rho_i = \lambda_i/\mu_i$ and $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_N)^{\mathrm{T}}$.

The network is a Whittle network if the service intensities $\phi(\mathbf{x})$ are balanced, i.e. the rates satisfy the balance conditions [Ser99]

$$\frac{\phi_i(\mathbf{x} - \mathbf{e}_j)}{\phi_i(\mathbf{x})} = \frac{\phi_j(\mathbf{x} - \mathbf{e}_i)}{\phi_j(\mathbf{x})} \quad \forall \mathbf{x}, i, j, x_i > 0, x_j > 0. \tag{2.4}$$

All balanced service rates can be expressed in terms of a unique balance function Φ so that $\Phi(\mathbf{0}) = 1$, $\Phi(\mathbf{x}) = 0, \forall \mathbf{x} \notin \mathbb{Z}_+^N$, and

$$\phi_i(\mathbf{x}) = \frac{\Phi(\mathbf{x} - \mathbf{e}_i)}{\Phi(\mathbf{x})} \quad \forall \mathbf{x} : x_i > 0. \tag{2.5}$$

The steady-state distribution of the process is

$$\pi(\mathbf{x}) = \frac{1}{G(\boldsymbol{\rho})} \Phi(\mathbf{x}) \prod_{i=1}^{N} \rho_i^{x_i}, \tag{2.6}$$

where $G(\boldsymbol{\rho})$ is the normalization constant

$$G(\boldsymbol{\rho}) = \sum_{\mathbf{x}} \Phi(\mathbf{x}) \prod_i \rho_i^{x_i}. \tag{2.7}$$

Assuming that EPS queueing discipline is used at every node, the assumption of exponential service requirements can be relaxed. A network with EPS nodes is insensitive to the service requirement distribution if and only if it is a Whittle network [BP02]. The EPS nodes of a Whittle network can be replaced with sets of EPS nodes allowing phase-type distributions without affecting the steady state distribution (see, e.g., [BP02, Bon06a]). The steady-state distribution (2.6) depends on the routing probabilities p_{ij}, arrival intensities λ_i and service requirement distributions only through the node loads ρ_i.

Application to Data Networks

The processor sharing networks defined in previous section can be used in teletraffic modeling. Both circuit-switched and data networks can be modeled at call or flow level [BP02]. In this thesis, we study only data networks. The customers in the queueing network represent data flows. The capacity of a PS node is the bit rate allocated for the flows in that node.

We consider a network used by K flow classes. A flow class represents a set of similar flows in terms of network resource usage. For example, a class in a wired network corresponds to a path. Let x_k be the number of class-k flows in progress and denote the network state by $\mathbf{x} = (x_1, \ldots, x_K)^{\mathrm{T}}$.

External arrivals in the queueing network correspond to session arrivals. Session arrival process is Poissonian and each session can consist of several file transfers and think times, see, e.g. [BP02, Bon06a]. Files sizes and think-time distributions can be arbitrary and even correlated. We denote by ρ_k the mean traffic intensity (in bits/s) of class-k flows, defined as the product of the flow arrival rate and the mean flow size. We use the notation $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_K)^{\mathrm{T}}$.

In each state $\mathbf{x}$ the network resources are shared by the contending flows. Let $\boldsymbol{\phi}(\mathbf{x}) = (\phi_1(\mathbf{x}), \ldots, \phi_K(\mathbf{x}))^{\mathrm{T}}$ be the vector of capacities allocated to each flow class in state $\mathbf{x}$. The allocation $\phi_k(\mathbf{x})$ is assumed to be equally shared between the x_k flows in each class k. The network resources are defined by the capacity set $\mathcal{C}$, which is a collection of capacity allocations $\boldsymbol{\phi}$ that can be supported by the network. Depending on the network, different restrictions limit $\mathcal{C}$. In fixed networks, capacity allocations are limited by the link capacities. In wireless networks, capacity constraints are more complex as the transmissions interfere each others.

The throughput experienced by flows depends on how the available capacity is shared between the active flows. A capacity allocation is insensitive if and only if it satisfies the balance condition (2.4). Balanced fairness is the most efficient balanced allocation. BF can be constructed recursively starting from an empty network. In each state $\mathbf{x} > \mathbf{0}$, the amount of allocated capacity is maximized within the capacity set $\mathcal{C}$ while satisfying the balance condition. BF is the only balanced allocation that is constrained by the capacity set $\mathcal{C}$ at every state. Of all balanced allocations, BF is the one for which the network is empty with the highest probability [BP04]. BF allocation can be determined recursively by setting $\Phi(\mathbf{0}) = 1$, $\Phi(\mathbf{x}) = 0, \forall \mathbf{x} \notin \mathbb{Z}_+^N$, and using recursion

$$\Phi(\mathbf{x}) = \min\{\alpha : \frac{(\Phi(\mathbf{x} - \mathbf{e}_1), \ldots, \Phi(\mathbf{x} - \mathbf{e}_K))^{\mathrm{T}}}{\alpha} \in \mathcal{C}\}. \qquad (2.8)$$

Capacity allocation in each state $\mathbf{x}$ is defined by the balance function:

$$\phi_k(\mathbf{x}) = \frac{\Phi(\mathbf{x} - \mathbf{e}_k)}{\Phi(\mathbf{x})}. \tag{2.9}$$

A capacity allocation is insensitive if and only if (2.9) holds in each state $\mathbf{x}$. The capacity ratios in state $\mathbf{x}$ are hence fixed by values $\Phi(\mathbf{x} - \mathbf{e}_i)$ and the capacities corresponding to BF are obtained by increasing the capacities until the border of the capacity set $\mathcal{C}$ is reached.

A network with BF capacity allocation is stable if and only if the traffic load vector $\boldsymbol{\rho}$ is within the capacity set $\mathcal{C}$ of the networks, i.e. $\boldsymbol{\rho} \in \mathcal{C}$ [BMPV06]. The steady state distribution of the system under BF is

$$\pi(\mathbf{x}) = \frac{1}{G(\boldsymbol{\rho})} \Phi(\mathbf{x}) \prod_i \rho_i^{x_i}, \tag{2.10}$$

where $G(\boldsymbol{\rho})$ is the normalization constant (2.7).

The steady state distribution is insensitive, hence all performance measures derivable from it are insensitive. We are interested in flow throughput, defined as the ratio of the mean flow size to the mean flow duration. The flow throughput γ_k of class-k can be expressed using Little's result and equations (2.7) and (2.6):

$$\gamma_k = \frac{\rho_k}{\mathrm{E}[x_k]} = \frac{\rho_k}{\sum_{\mathbf{x}} x_k \pi(\mathbf{x})} = \frac{G(\boldsymbol{\rho})}{\frac{\partial}{\partial \rho_k} G(\boldsymbol{\rho})} \; . \tag{2.11}$$

For some simple systems the normalization constant (2.7) and, therefore, the throughput (2.11) can be solved in closed form [BP03a, BV04, BPRV03]. Generally, however, one has to resort to numerical analysis. A straightforward method is to truncate the state space and compute the state probabilities in the truncated state space. The benefit of BF is that the (unnormalized) probabilities can be obtained recursively state-by-state using (2.10) and (2.8), which is much easier than solving the global balance equations (2.1) essentially entailing a matrix inversion. BF provides a reasonable approximation of flow throughput in systems where other fair resource sharing schemes, such as max-min or proportional fairness, are applied [BMPV06, Tim03]. Thus BF makes it possible to study larger systems than is feasible with other capacity allocation schemes. For very large systems, however, the recursive solution ultimately becomes infeasible calling for approximate methods, such as performance bounds [BP04, Bon06b] and asymptotic approximation [BPV06]. Whether state space truncation or some other performance approximation method is used, the capacity set $\mathcal{C}$ of the network needs to be defined. It can be determined either explicitly or implicitly. In many instances, it is infeasible to define the capacity set explicitly, but the border point of the capacity space is determined only in a given direction, for example when solving the recursion (2.8).

Sparse Matrix Notation for Balanced Fairness

In some cases, it is beneficial to define balanced fairness using a different notation. In systems with large number of flow classes it is useful to consider balanced fairness in sparse matrix notation. Especially if the customer classes are not numerable, the vector formulation is not feasible. The case with continuous index set is discussed in [BPV06], in which a wireless network is studied and the class index of a flow is the (continuous) position of the user.

Let ξ be the set of indices to active flows, i.e., $\mathbf{x} = \sum_{i \in \xi} \mathbf{e}_i$. Note that the same index may appear in ξ more than once. With this notation the recursion (2.8) can be written:

$$\Phi(\xi) = \min\{\alpha : \frac{(\Phi(\xi \setminus \{\hat{\xi}_1\}), \ldots, \Phi(\xi \setminus \{\hat{\xi}_L\}))^{\mathrm{T}}}{\alpha} \in \mathcal{C}^{\hat{\xi}}\}, \qquad (2.12)$$

where $\hat{\xi} = \bigcup \xi$, i.e. the set of different flow classes in ξ, $L = |\hat{\xi}|$, and $\mathcal{C}^{\hat{\xi}}$ is the capacity set defined for flow classes $\hat{\xi}$ only. In other words, to compute the value of the balance function, we remove active flows one by one until we reach $\Phi(\emptyset)$, which is 1, by convention.

The sparse matrix notation should be viewed as an alternative implementation of the recursion (2.8), which is especially suitable for evaluating values of balance function for a small set of flows when the total number of flow classes is large. In such a case most of the flow classes are empty and we can neglect all the resources used only by the empty classes in constructing $\mathcal{C}^{\hat{\xi}}$. This significantly reduces the computational burden and and memory consumption of the recursion compared to applying (2.8). The notation can often be applied even in cases where the whole capacity set $\mathcal{C}$ cannot be handled.

3 PERFORMANCE ANALYSIS METHODS

3.1 Introduction

In this chapter, we discuss different performance analysis methods applicable to the models discussed in the previous chapter. After a system has been modeled as a stochastic process, the performance of the original system can be analyzed using the model. With complicated systems and models, approximative methods are usually needed, as exact analysis is often either infeasible or computationally too heavy. In this chapter, we discuss several approximative performance analysis methods that are applied to different telecommunication settings in Chapters 4 and 5.

First, we give a brief introduction to the theory of Markov decision processes (MDPs). MDPs are used in modeling systems that are partly random and partly controllable, e.g., in a telecommunication network the arrival process is random but routing can be controlled. The aim is to control the process in such a way that some performance metric is optimized. We use two different optimization methods, namely linear programming formulation and policy iteration.

We also present a new approximative method based on the MDP setting called value extrapolation, developed during this work in Publications 1 and 7. It can be used to approximate any performance metric that can be formulated as the expected value of a function of the system state, which includes, e.g., the moments of the number of customers in a queueing system. Using Little's theorem, the mean time in the system can also be determined. Instead of the state probabilities being solved using the balance equations, the performance measure is determined directly using relative values of the states and so-called Howard equations. The advantage of this approach is that the relative values outside the truncated state space can often be well extrapolated using a polynomial function, allowing more accurate results to be obtained without any significant computational penalty. As illustrated in Publication 1, value extrapolation provides accurate performance measures of queueing networks even with heavily truncated state spaces.

Finally, we discuss the performance approximation of systems using balanced fairness capacity allocation. In [BPV06], Bonald et al. introduced asymptotic throughput analysis, a method for approximating the mean flow throughputs of a network. If the proportions of traffic in different classes are fixed, the throughput of a class can be approximated. Mean throughput and its derivatives can be determined at zero load, allowing throughput to be extrapolated as the load of the system increases. If a traffic class is saturated at the capacity limit, the end point of the curve is also known, thus making possible more accurate interpolation. In some cases, the throughput derivative at the capacity limit can also be determined. We also present a novel BF approximation method based on the Monte Carlo method and introduced in Publication 2. Instead of the state space being recursively gone through in order, it is sampled randomly and the average throughput is calculated.

3.2 Markov Decision Processes

In this section, we briefly introduce the most important concepts of the theory of Markov decision processes (MDP) relevant to this thesis. For a more extensive introduction to MDP theory, see, e.g., [Tij94]. Applications of MDPs in telecommunications are reviewed in [Alt02a].

Markov decision processes are a generalization of Markov processes. State transitions of Markov processes are random, but in MDPs the state transitions are partly random and partly controllable. The aim is to control the process so that the value of a revenue function corresponding to a performance metric is maximized (or minimized). Typically either the expected discounted revenue or the average revenue is maximized. In this thesis, we are mainly interested in the mean queue length of queueing systems, and hence the average revenue is optimized.

Depending on the characteristics of a process, different methods can be used to find the optimal controls. Time, as well as the state space, can be either discrete or continuous. In this thesis, we study systems with continuous time and discrete state space. The process can be controlled by taking actions. In each state, action a is selected from the action space $\mathcal{A}$. Depending on the model, action space can be either discrete or continuous. The set of feasible actions can depend on the system state, but in this thesis we only study systems that have the same action space in every state. Policy α defines an action for each system state. Transition intensities of the process depend on the actions. When policy α is used, transition intensity from state $\mathbf{x}$ to state $\mathbf{y}$ is denoted $q_{\mathbf{xy}}(\alpha)$. Given a fixed policy α, an MDP reduces to a normal Markov process.

We are interested in performance metrics that can be expressed as a mean value of a revenue rate that is a function of the system state, most importantly the mean number of customers in the system. In our applications, revenue does not depend on the actions, but the formulations can be generalized allowing that. Let the revenue rate of the process in state $\mathbf{x}$ be $r(\mathbf{x})$. The mean revenue rate of the process depends on the policy and is denoted $\overline{r}_\alpha$.

When queueing systems are considered, the mean occupancy $\mathrm{E}[|X|]$ can be determined by using revenue function $r(\mathbf{x}) = |\mathbf{x}|$, where notation $|\mathbf{x}| = \sum_i x_i$ is used. Similarly, other moments of $|X|$ may be determined. If metrics concerning only class-k customers are optimized, revenue functions such as $r(\mathbf{x}) = x_k$ or $r(\mathbf{x}) = x_k^2$ can be used.

There are several methods for finding the optimal policy, i.e. the one optimizing the mean revenue. In this thesis, we apply linear programming formulation and policy iteration.

Linear Programming Formulation

The optimal policy of an ergodic MDP can be found utilizing linear programming (LP). The problem is formulated as an LP problem and the solution gives the optimal policy optimizing the mean revenue $\overline{r}$. We formulate the problem so that it can be applied to the capacity allocation problems discussed in this thesis.

The steady state probability that the system is at state $\mathbf{x}$ and action $a \in \mathcal{A}$ is used is denoted $\pi(\mathbf{x}, a)$. We assume that the decision space $\mathcal{A}$ is a convex hull spanned by actions $a' \subset \mathcal{A}$ and that the state transition intensities are linear combinations of the intensities corresponding to the spanning vectors. If action $a = \sum_{i \in a'} b_i a', b_i \geq 0, \sum_i b_i = 1$ is used in state $\mathbf{x}$, transition intensities are $q_{\mathbf{xy}}(a) = \sum_{i \in a'} b_i q_{\mathbf{xy}}(a')$. For example, the capacity of a server can be divided between customers in any ratios and the state transition intensities of the process are proportional to the capacity shares. The optimization problem can now be formulated as an LP problem:

$$\max \sum_{\mathbf{x}} \sum_{a \in a'} r(\mathbf{x}) \pi(\mathbf{x}, a) \tag{3.1}$$

$$\text{s.t.} \sum_{\mathbf{y}} \sum_{a \in a'} \pi(\mathbf{x}, a) q_{\mathbf{xy}}(a) = \sum_{\mathbf{y}} \sum_{a \in a'} \pi(\mathbf{y}, a) q_{\mathbf{yx}}(a) \quad \forall \mathbf{x} \in \mathcal{S} \tag{3.2}$$

$$\sum_{\mathbf{x}} \sum_{a \in a'} \pi(\mathbf{x}, a) = 1 \tag{3.3}$$

$$\pi(\mathbf{x}, a) \geq 0 \quad \forall \mathbf{x}, a, \tag{3.4}$$

where (3.2) corresponds to the global balance equations (2.1) and (3.3) is the normalization constraint.

Using the solution of the LP problem, the state probabilities and actions can be determined as

$$\pi(\mathbf{x}) = \sum_{a \in a'} \pi(\mathbf{x}, a) \tag{3.5}$$

and

$$a(\mathbf{x}) = \frac{\sum_{a \in a'} a \pi(\mathbf{x}, a)}{\sum_{a \in a'} \pi(\mathbf{x}, a)}. \tag{3.6}$$

The optimal mean revenue is given by the objective function (3.1).

Policy Iteration

Policy iteration is another method for solving MDP problems. Starting from some initial policy α_0, the policy is iteratively improved until the optimum is found. In this section, we assume that the action space is discrete.

Instead of state probabilities, policy iteration is based on a state metric called relative value. Relative value $v_\alpha(\mathbf{x})$ of state $\mathbf{x}$ using policy α is the conditional expected difference in cumulative revenue over infinite time horizon when starting from state $\mathbf{x}$ rather than from equilibrium:

$$v_\alpha(\mathbf{x}) = \mathrm{E}\left[\int_{t=0}^{\infty} (r(X_\alpha(t)) - \overline{r}_\alpha) \, dt \;\middle|\; X(0) = \mathbf{x} \right], \tag{3.7}$$

where $\overline{r}_\alpha$ is the mean revenue when policy α is used. Relative values $v_\alpha(\mathbf{x})$ and the mean revenue $\overline{r}_\alpha$ satisfy the so-called Howard equations (the equivalent equations are usually called Poisson's equations in control theory)

$$r(\mathbf{x}) - \overline{r}_\alpha + \sum_{\mathbf{y}} q_{\mathbf{xy}}(\alpha)(v_\alpha(\mathbf{y}) - v_\alpha(\mathbf{x})) = 0 \quad \forall \mathbf{x}. \tag{3.8}$$

As seen, only the differences of the relative values appear in the equations, hence we may set, e.g., $v_\alpha(\mathbf{0}) = 0$ when solving the equations. From the $|\mathcal{S}|$ equations, the mean revenue rate $\overline{r}_\alpha$ along with the $|\mathcal{S}| - 1$ unknown relative values can be solved.

Given a fixed policy, the mean revenue and corresponding relative values can be solved from the Howard equations (3.8). The policy α minimizing the mean revenue rate $\overline{r}_\alpha$ can be found using policy iteration algorithm, which iteratively improves an initial policy until the optimum is found. The iteration typically converges quickly [Tij94]. Policy iteration algorithm proceeds as follows:

1. Select an initial policy α_0 and set $i = 0$
2. Compute the mean revenue $\overline{r}_{\alpha_i}$ and the relative values $v_{\alpha_i}(\mathbf{x})$ using policy α_i and the Howard equations (3.8)
3. Specify a new policy α_{i+1} by selecting in each state the action $a_{i+1}(\mathbf{x}) = \arg\max_a \frac{r(\mathbf{x})}{\sum_{\mathbf{y}} q_{\mathbf{xy}}(a)} + \sum_{\mathbf{y}} \frac{q_{\mathbf{xy}}(a)}{\sum_{\mathbf{z}} q_{\mathbf{xz}}(a)} v_{\alpha_i}(\mathbf{y})$
4. If $\alpha_{i+1} \neq \alpha_i$, set $i = i + 1$ and goto 2

Step 2 of the algorithm entails solving the Howard equations (3.8). The number of equations and unknown variables is equal to the number of states in the state space.

3.3 Value Extrapolation

Value extrapolation is a new approximation method based on the MDP theory. It was first briefly introduced in Publication 7 and more thoroughly discussed in Publication 1. Value extrapolation approach is applicable to performance measures that can be expressed as a mean value of a revenue that is a function of the system state. The idea is to solve the mean revenue using a truncated state space more efficiently than with the straightforward truncation.

Instead of state probabilities and global balance equations, relative values and Howard equations are used. Instead of simply ignoring the transitions from the truncated state space to its outside (we refer to this as regular truncation), the relative values outside are extrapolated using the values inside (note that the Howard equations remain a closed set of equations) as illustrated in Figure 3.1. Depending on the accuracy of the extrapolation, the accuracy of the mean revenue can be greatly improved. If the relative values outside $\tilde{\mathcal{S}}$ are correctly extrapolated, the mean revenue solved from the Howard equations (3.8) is exact. While exact results are obtained only in some special cases, value extrapolation often improves the accuracy significantly.

The extrapolation problem can be formulated as follows. We want to fit a function f to the relative values inside the truncated state space $\tilde{\mathcal{S}}$ so that it approximates also the values outside $\tilde{\mathcal{S}}$. Function f and the fitting method need to be chosen so that the approximated relative values outside $\tilde{\mathcal{S}}$ are linear functions of the values inside, so that the group of equations (3.8) remains linear.

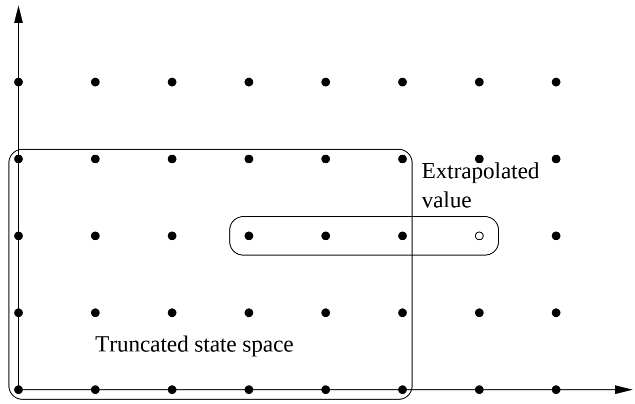


Figure 3.1: Two dimensional value extrapolation. Relative value of the point outside the truncated state space can be extrapolated using the values inside.

One linear extrapolation method is to use a polynomial function $f(\mathbf{x}) = \sum_{i=1}^{K} \sum_{j=0}^{n_i} a_{i,j} x_i^j$ and least squares fitting. The fitting can be done either globally or locally. When global fitting is used, all the $(\mathbf{x}, v(\mathbf{x}))$-pairs in $\tilde{\mathcal{S}}$ are used. The fitting can also be done locally, i.e. using only a subset $\mathcal{S}_f(\mathbf{x})$ of the truncated state space. The choice of $\mathcal{S}_f(\mathbf{x})$ may depend on the extrapolated point $\mathbf{x}$. Parameters $a_{i,j}$ are determined so that the sum of squared errors

$$Q = \sum_{\mathbf{x} \in \mathcal{S}_f} (f(\mathbf{x}) - v(\mathbf{x}))^2 \tag{3.9}$$

is minimized. Parameter values a_{ij} minimizing (3.9) are linear functions of $v(\mathbf{x})$ and hence also the extrapolation function $f(\mathbf{x})$ is a linear function of $v(x)$. If the number of free parameters in the extrapolation function and the number of states in $\mathcal{S}_f(\mathbf{x})$ are equal, the least squares fitting reduces to ordinary polynomial fitting. Function f and set $\mathcal{S}_f(\mathbf{x})$ need to be chosen so that the parameters have unambiguous values, i.e. the number of points in $\mathcal{S}_f(\mathbf{x})$ is equal or greater than the number of parameters. The optimal parameter values are found by minimizing the error (3.9). The parameter values and hence also the function $f(\mathbf{x})$ are linear functions of relative values $v(\mathbf{x})$ inside the truncated state space $\mathcal{S}_f(\mathbf{x})$.

The mean revenue rate $\overline{r}$ can be approximated by defining the Howard equations (3.8) for the truncated state space and extrapolating the relative values outside $\tilde{\mathcal{S}}$ that appear in the equations. Value extrapolation does not alter the number of equations or variables in the group of equations, hence there is no significant computational penalty when value extrapolation is used.

In this thesis, the extrapolation is done using polynomial fitting. In particular, quadratic polynomials are used when mean queue lengths are approximated, e.g., $v(\ldots, N+1, \ldots) = 3v(\ldots, N, \ldots) - 3v(\ldots, N-1, \ldots) + v(\ldots, N-2, \ldots)$, where N is the maximum number of customers in a class. A strong motivation for this procedure is that value extrapolation with polynomial fitting leads to exact results in certain queueing systems.

Consider for example the mean queue length of an M/M/1-queue with unit capacity. The revenue rate $r(x)$ in a given state x is then simply the number of customers in that state, i.e. the state index itself. Let arrival rate be λ, mean service requirement $1/\mu$ and denote $\rho = \frac{\lambda}{\mu}$. Now the Howard equations can be written as

$$x - \overline{r} + \lambda(v(x+1) - v(x)) + \mu(v(x-1) - v(x)) = 0, \quad \forall x > 0. \tag{3.10}$$

The equations are clearly solved by

$$\overline{r} = \frac{\rho}{1-\rho}, \qquad v(x+1) - v(x) = \frac{x+1}{\mu - \lambda}, \tag{3.11}$$

from which by setting $v(0) = 0$, we get

$$v(x) = \frac{x(x+1)}{2(\mu - \lambda)}. \tag{3.12}$$

Relative values are defined by a quadratic polynomial of the state variable. Thus, extrapolating the relative value with a second order polynomial yields exact value for $\overline{r}$ no matter how small the truncated space is as long as the fitting can be done, i.e. at least 3 states are needed. While the results are not exact for general queueing systems, accurate results can be expected.

Numerical Example

We illustrate value extrapolation and its accuracy using a generalized processor sharing (GPS) queue with unit capacity and two customer classes, see, e.g., [vU03]. GPS is another processor sharing variant with customer class differentiation. Capacity allocation of class k is defined as follows:

$$\phi_k(\mathbf{x}) = \frac{w_k}{\sum_{j:x_j>0} w_j}, \tag{3.13}$$

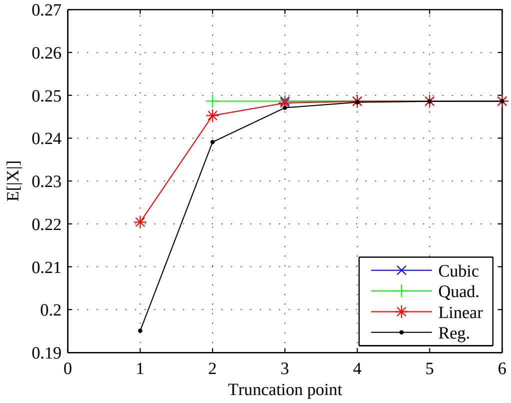
where w_k is the weight of class k. The arrivals are Poissonian with rates λ_1 and λ_2 and the service requirements are exponentially distributed with means $1/\mu_1$ and $1/\mu_2$. The Howard equations of the system read

$$\begin{aligned} r(\mathbf{x}) - \overline{r} + \sum_{i=1}^{2} \lambda_i (v(\mathbf{x} + \mathbf{e}_i) - v(\mathbf{x})) + \\ + \sum_{i=1}^{2} \frac{w_i \mu_i}{\sum_{j:x_j>0} w_j} (v(\mathbf{x} - \mathbf{e}_i) - v(\mathbf{x})) = 0, \quad \forall \mathbf{x}. \end{aligned} \tag{3.14}$$

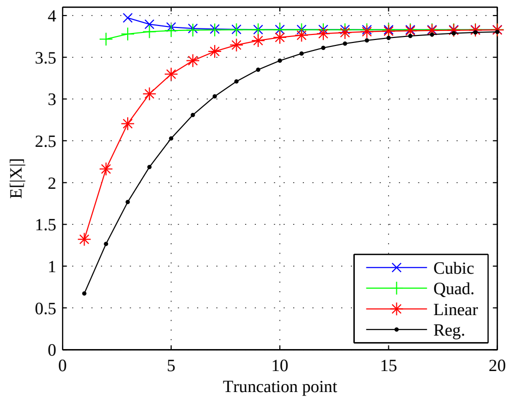
We compare accuracy of different polynomial extrapolation functions. Extrapolation is conducted using polynomial fitting, hence $n+1$ data points are needed when nth order polynomial is used. For example, if class 1 is truncated at point N, the linearly and quadratically extrapolated values outside the truncated state space are

$$v(N+1, x_2) = 2v(N, x_2) - v(N-1, x_2) \tag{3.15}$$

$$v(N+1, x_2) = 3v(N, x_2) - 3v(N-1, x_2) + v(N-2, x_2). \tag{3.16}$$



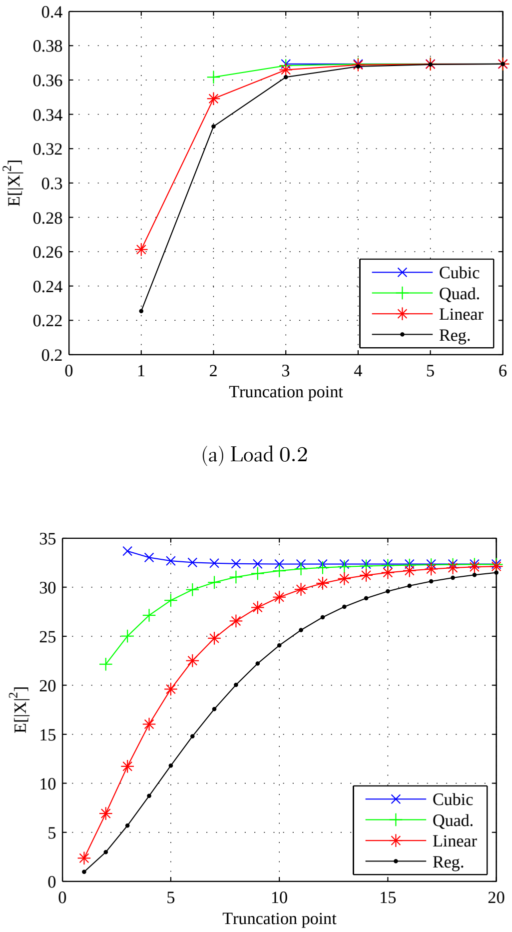
(a) Load 0.2



(b) Load 0.8

Figure 3.2: E[$|X|$] of a GPS process with different extrapolation functions

The parameter values used are $w_1 = 3/10, w_2 = 7/10, \lambda_2 = 2\lambda_1, \mu_1 = 1/2$ and $\mu_2 = 2/3$. Arrival intensity is varied so that the load of the system is either 0.2 or 0.8. When the size of the truncated state space is increased, the convergence of the queue length approximation can be observed. The mean queue length is illustrated in Figure 3.2. With the lower load level, there is no significant difference between the functions as the mean queue length converges quickly regardless of the extrapolation method. With the higher load, the differences are more remarkable. Quadratic extrapolation converges the quickest, allowing accurate results with a heavily truncated state space. Second moment of the queue length is illustrated in Figure 3.3. In this case, cubic extrapolation yields the best results. In both cases, linear extrapolation also clearly outperforms regular truncation.



(a) Load 0.2

(b) Load 0.8

Figure 3.3: $\mathrm{E}[|X|^2]$ of a GPS process with different extrapolation functions.

Discriminatory Processor Sharing Queue

As discussed earlier, relative values of an M/M/1 queue are defined by a quadratic function of the system state. In this section, we provide similar results related to a DPS queue based on Publication 4.

First, we show that the first moment of the total occupancy of a DPS system with K customer classes can be exactly calculated using value extrapolation. In this case, the revenue rate in state $\mathbf{x}$ is $r(\mathbf{x}) = \sum_k x_k$. The Howard equations of the system read

$$\begin{aligned} r(\mathbf{x}) - \overline{r} + \sum_{k=1}^{K} \lambda_k (v(\mathbf{x}+\mathbf{e}_k) - v(\mathbf{x})) + \\ + \sum_{k=1}^{K} \frac{w_k x_k \mu_k}{\sum_{d=1}^{K} w_d x_d} (v(\mathbf{x}-\mathbf{e}_k) - v(\mathbf{x})) = 0 \quad \forall \mathbf{x} \geq \mathbf{0}. \end{aligned} \tag{3.17}$$

As a trial for the relative values in the Howard equations (3.17), we use a second order polynomial function of the form

$$v(\mathbf{x}) = \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} a_{k_1,k_2} x_{k_1} x_{k_2} + \sum_{k=1}^{K} a_k x_k, \tag{3.18}$$

and assume that $a_{k_1,k_2} = a_{k_2,k_1}$. When the trial is substituted into the Howard equations (3.17), the differences of the relative values appearing in the expressions reduce to linear polynomials, for example

$$v(\mathbf{x}-\mathbf{e}_k) - v(\mathbf{x}) = -a_k + a_{k,k} - 2 \sum_{k_1=1}^{K} a_{k,k_1} x_{k_1}. \tag{3.19}$$

After the substitution, we multiply the equation with $\sum_{d=1}^{K} w_d x_d$ and regroup the terms. The following equation is obtained:

$$\begin{aligned} \sum_{k_1}^{K} \Big(-\overline{r} + (a_{k_1,k_1} - a_{k_1})\mu_{k_1} + \sum_{k_2}^{K} (a_{k_2} + a_{k_2,k_2})\lambda_{k_2} \Big) w_{k_1} x_{k_1} + \\ + \sum_{k_1}^{K} \sum_{k_2}^{K} \Big(1 - 2\mu_{k_1} a_{k_1,k_2} + 2 \sum_{k_3}^{K} \lambda_{k_3} a_{k_2,k_3} \Big) w_{k_1} x_{k_1} x_{k_2} \\ = 0. \end{aligned} \tag{3.20}$$

The Howard equations (3.17) must be satisfied for all $\mathbf{x} \in \mathcal{S}$. Equation (3.20) holds for all $\mathbf{x}$ if the coefficient of each x_k and each unique x_{k_1}, x_{k_2}-combination equals zero. When all the coefficients are set equal to zero we get $K^2/2 + 3K/2$ linear equations. This is exactly the number of unknown parameters in the trial function (3.18), hence we may select the parameter values in the trial function so that the Howard equations are satisfied regardless of the values of the system parameters. Our numerical experimentations indicate that the group of equations is indeed solvable.

We conclude that $v(\mathbf{x})$ is a quadratic function of $\mathbf{x}$ hence it can be extrapolated exactly with a quadratic polynomial and value extrapolation yields the total mean occupancy exactly.

A similar deduction can be done for the mean occupancy of class k. In this case, the revenue rate function is $r(\mathbf{x}) = \sum_q x_{k,q}$. Using the same trial function (3.18), we obtain an expression similar to (3.20). The number of unknown parameters is the same as the number of linear equations, hence the value function is quadratic and value extrapolation yields the exact mean number of class-k customers.

Higher moments of the occupancy can be determined similarly to the first moment. When the nth moment is considered, an $(n+1)$th order polynomial trial function is used:

$$v(\mathbf{x}) = \sum_{k_1=1}^{K} \cdots \sum_{k_{n+1}=1}^{K} a_{k_1,\ldots,k_{n+1}} \cdot x_{k_1} \cdots x_{k_{n+1}} + \cdots + \sum_{k_1=1}^{K} a_{k_1} x_{k_1}. \tag{3.21}$$

The differences of the relative values in (3.17) reduce to nth order polynomials and substitution into (3.17) yields an $(n+1)$th order polynomial equation. Similarly to the first moment, the number of unknown parameters in the trial function is equal to the number of linear equations, hence the parameter values may be chosen so that the Howard equations are satisfied for all $\mathbf{x}$. We conclude that $v(\mathbf{x})$ is an $(n+1)$th order polynomial, hence the nth moment can be solved exactly using $(n+1)$th order value extrapolation.

The same inspection can be conducted with more general processes by using Cox-distributed service requirements (Publication 4). Cox distributions can be used to approximate any distribution with arbitrary accuracy. Also in this case, the relative values are polynomials of the system state. However, the computational complexity grows quickly as the number of phases in the distributions increase.

While the exact results obtained with DPS queues are an exception, the accurate results with the GPS queue suggest that value extrapolation can be successfully applied to queueing systems. When the mean queue length is computed, a quadratic polynomial seems to yield the most accurate results.

3.4 Approximative Methods Utilizing Balanced Fairness

Balanced fairness capacity allocation scheme introduced in Section 2.4 facilitates the analysis of telecommunication systems. Equilibrium state probabilities are easier to solve with balanced fairness than with other capacity allocation policies as the probabilities can be determined state-by-state using recursion (2.8), instead of solving the global balance equations (2.1) entailing a matrix inversion. Still, if the number of traffic classes is high or the system is heavily loaded, the problem becomes numerically intractable. Two approximative methods, throughput asymptotics and performance bounds, based on BF have been introduced to overcome the computational problems.

Bonald et al. approximated the performance of BF systems by studying the asymptotic throughput in [BPV06]. The mean flow throughput and its derivatives at zero load can be determined and used to extrapolate the throughput with higher loads. In some cases, mean throughput and its first derivative can also be computed at the capacity limit of the system allowing more accurate approximation. In Chapters 4 and 5, we use throughput asymptotics to analyze various telecommunication systems.

Another approximative approach is to derive performance bounds. In [BP04], Bonald and Proutière proved that in fixed networks BF always performs better than a network operated using the store-and-forward technique, hence the store-and-forward network gives a lower performance bound. A tighter set of bounds was derived in [Bon06b] assuming that the capacity constraints of the network are linear which covers both fixed and many wireless networks.

In addition to the existing approaches, we introduce a new approximative method based on the Monte Carlo method for evaluating large sums. Instead of the state space being recursively gone through in order, it is sampled randomly and the average throughput is calculated.

Throughput Asymptotics

In [BPV06], Bonald et al. introduced a method for approximative BF performance analysis. By computing the asymptotic throughput value and its derivative at zero load, one can sketch the throughput behavior in the system when load is increased from zero along a given load line. If a traffic class is saturated at the capacity limit of the system, asymptotic throughput and, in some cases, its derivative can be computed also at heavy loads allowing more accurate sketching using interpolation.

We denote by $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_K)^{\mathrm{T}}$ the traffic profile of the network, where ρ_k is the proportion of class-k traffic and $\sum \rho_k = 1$. Load line is the line segment from zero to the boundary point $\hat{\boldsymbol{\rho}}$ of the capacity set $\mathcal{C}$ in the direction of vector $\boldsymbol{\rho}$. The load line can be parameterized as $r\hat{\boldsymbol{\rho}},\ r \in [0, 1]$.

The derivatives of the throughput (2.11) with respect to r at load $r = 0$ can be determined. The first derivative was presented in [BPV06]. We extend the approach by providing the explicit expressions needed for the second derivative. Using notation $G_i(\boldsymbol{\rho}) = \frac{\partial}{\partial \rho_i} G(\boldsymbol{\rho})$ the first two derivatives read

$$\begin{aligned}
\gamma_i(0) &= \frac{G(0)}{G_i(0)}, \\
\gamma_i'(0) &= \frac{G_i(0)G'(0) - G(0)G_i'(0)}{G_i(0)^2}, \\
\gamma_i''(0) &= \frac{2G_i'(0)\left(G(0)G_i'(0) - G'(0)G_i(0)\right)}{G_i(0)^3} + \\
&\quad + \frac{G_i(0)\left(G_i(0)G''(0) - G(0)G_i''(0)\right)}{G_i(0)^3},
\end{aligned} \tag{3.22}$$

where $G(0) = 1$ and the rest of the terms are

$$G'(0) = \sum_{j=1}^{N} \Phi(\mathbf{e}_j)\hat{\rho}_j, \tag{3.23}$$

$$G''(0) = 2\sum_{j=1}^{N}\sum_{k=j}^{N}\Phi(\mathbf{e}_j + \mathbf{e}_k)\hat{\rho}_j\hat{\rho}_k, \tag{3.24}$$

$$G_i(0) = \Phi(\mathbf{e}_i), \tag{3.25}$$

$$G'_i(0) = \sum_{j=1}^{N}\Phi(\mathbf{e}_i + \mathbf{e}_j)\hat{\rho}_j + \Phi(2\mathbf{e}_i)\hat{\rho}_i, \tag{3.26}$$

$$G''_i(0) = 2\left(\sum_{j=1}^{N}\sum_{k=j}^{N}\Phi(\mathbf{e}_i + \mathbf{e}_j + \mathbf{e}_k)\hat{\rho}_j\hat{\rho}_k + \right. \\ \left. + \sum_{j=1}^{N}\Phi(2\mathbf{e}_i + \mathbf{e}_j)\hat{\rho}_i\hat{\rho}_j + \Phi(3\mathbf{e}_i)\hat{\rho}_i^2\right). \tag{3.27}$$

Similarly, the higher derivatives can be expressed in terms of the values of the balance function. In order to determine the nth low-load derivative, the balance function values are needed for all states containing up to $n+1$ flows. Given the starting point and the derivatives, throughput curves can be sketched.

If a traffic class is constrained at the boundary point $\hat{\boldsymbol{\rho}}$, its throughput tends to zero when $r \to 1$. In this case, both the end points are known and the throughput curve can be interpolated resulting in a more accurate fit. The derivative of a saturated traffic class at high load can be computed, if the capacity set of the system is a polytope, i.e., $\mathcal{C} = \{\boldsymbol{\phi} : \mathbf{B}\boldsymbol{\phi} \leq \mathbf{e}\}$, where $\mathbf{B}$ is a matrix representing the constraints [BPV06]. Many interesting systems fall into this category [BMPV06]. Let $\mathcal{L}$ represent the set of saturated constraints $l \in \{1, \ldots, L\}$ at $r = 1$. It is conjectured (proven if $|\mathcal{L}| = 1$, conjectured if $|\mathcal{L}| > 1$) that the heavy load derivative is given by

$$\gamma'_i(1) = -\frac{1}{\sum_{l\in\mathcal{L}} b_{li}}, \tag{3.28}$$

where the b_{li} are the corresponding elements in $\mathbf{B}$. However, the heavy traffic derivative may not be the best overall descriptor for the heavy load behavior. As already discussed in [BPV06], the throughput curve may change very quickly at the heavy load end of the curve if only one constraint is saturated but several others are close to saturation. Numerical results in Publication 2 also suggest that the heavy load derivative usually does not improve the accuracy of the interpolation.

In some simple systems the functional form of the throughput curve is known, but in most cases the correct form is unknown. In this thesis, we interpolate the throughput curves using polynomial functions as they seem to produce reasonable results. The polynomials are fitted to the end points and the derivatives at zero load.

Throughput Evaluation in Low to Medium Load Range

In this section, we propose an alternative method for approximating the throughput directly in low and medium load range (Publication 2). At small and moderate loads the number of flows in the network is typically small. Correspondingly, the performance of the network at this load range is dominated by the states with only a few active flows. The idea of our approach is to write the throughput expression in a suitable form and then approximate the sums contained in the expression by the Monte Carlo method. The approach is based on the assumption that the capacity set can be constructed for any given set of flows if the number of flows is small.

The normalization constant (2.7) can be written in terms of increasing number of flows in the system. Accordingly, (2.11) becomes

$$\gamma_i(r) = \frac{1 + r\sum_{j\in\mathcal{A}}\Phi_j\hat{\rho}_j + r^2\sum_{j\in\mathcal{A}}\sum_{k\geq j}^{K}\Phi_{j,k}\hat{\rho}_j\hat{\rho}_k + \cdots}{\Phi_i + r\sum_{j\in\mathcal{A}}c_i(i,j)\Phi_{i,j}\hat{\rho}_j + r^2\sum_{j\in\mathcal{A}}\sum_{k\geq j}^{K}c_i(i,j,k)\Phi_{i,j,k}\hat{\rho}_j\hat{\rho}_k + \cdots}, \tag{3.29}$$

where notations $\Phi_i = \Phi(e_i)$ and $\Phi_{i,j} = \Phi(e_i + e_j)$ are used and the function $c_i(\cdot)$ gives the number of indices equalling i in its argument list.

Each n-fold summation in the expression (3.29) corresponds to going through all states where there are n active flows in total. Obviously, only the sums corresponding to a few active flows can be evaluated numerically. For others we use the Monte Carlo method: we draw n flows randomly, compute the corresponding term and repeat the procedure sufficiently many times to get an average which is then multiplied with the number of terms in the sum, $(K + n - 1)!/K!/(n - 1)!$.

Example
Consider the fixed network shown in Figure 3.4 (left), with 20 nodes and 40 links with unit capacity. The flows are characterized by the source-destination pairs with shortest path routing, totalling in 380 flow classes. Traffic pattern is uniform, i.e. the traffic load between all node pairs is equal. We study the throughput of the route shown in the figure. The boundary point of the capacity set is $\hat{\rho}_i = 1/112$, for all i. Figure 3.4 (right) shows the approximations of the route throughput. The curves (top-down) correspond to the throughput when the terms up to $\{1, 3, 5, 7, 9, 11\}$-fold sums are taken into account, respectively. Two first sums are computed by exhaustive enumeration and the subsequent sums with the Monte Carlo method using 10^5 samples. The figure shows also the asymptotic derivatives and the interpolated throughput using a cubic polynomial fitted to the end points and to the first two derivatives at $r = 0$. The interpolated curve fits well with the numerical results. The low load derivatives provide useful information on the curve but inclusion of the heavy load derivative would degrade the accuracy of the polynomial fitting.

3.5 Summary

Markov processes may be used in the modeling of many systems. The performance of such systems can be analyzed exactly only in some special cases, and hence approximative methods are needed. The traditional ap-

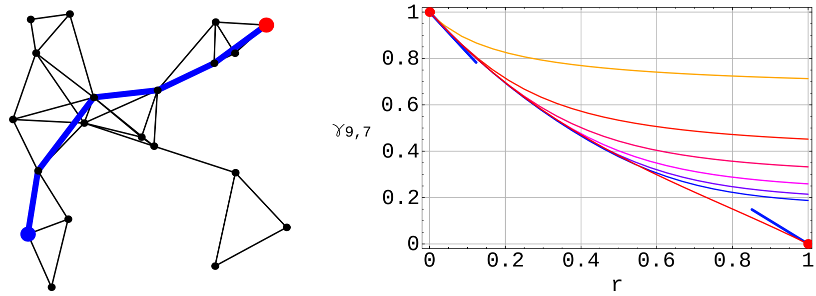


Figure 3.4: Left: Network topology and the studied route. Right: Throughput of the route by the Monte Carlo method with up to 11-fold sums. The curves (top-down) correspond to the throughput when the terms up to $\{1, 3, 5, 7, 9, 11\}$-fold sums are taken into account.

proach is to truncate the state space, solve the equilibrium state distribution, and use it to approximate the performance.

In this chapter, we introduced numerical methods that are used further in this thesis. We gave a brief overview of Markov decision processes, mostly on policy iteration and linear programming formulations. Both methods can be used to find a policy optimizing some performance criteria.

On the basis of the theory of MDPs, we introduced a new approximation method called value extrapolation, presented in Publications 1 and 7. It can be used to approximate performance metrics that can be formulated as the expected value of a function of the system state, which includes, e.g., the moments of the number of customers in a queueing system. Instead of the state probabilities being solved using the balance equations, the performance measure is determined directly using the relative values of the states and the Howard equations. The advantage of this approach is that the relative values outside the truncated state space can often be well extrapolated using a polynomial function and least squared error sum fitting without any significant computational penalty. We demonstrated the accuracy of value extrapolation by approximating the moments of queue length of a GPS system and polynomial fitting as the extrapolation method. In addition, we showed that value extrapolation yields exact queue length moments when applied to a DPS queue with Poissonian arrivals and Cox-distributed service requirement distributions. In addition, the exact relative values of the states are obtained.

We presented also an approximative method based on the throughput asymptotics of balanced fairness introduced by Bonald et al. in [BPV06]. Given a traffic profile, i.e. the traffic ratios of different classes, the throughput and its derivatives can be determined in the low load regime. If the traffic class is saturated at the capacity limit, the throughput tends to zero and the first derivative can be computed. Using this information, the throughput behavior can be sketched. As a new result, we gave the explicit expression of the second derivative at zero load in Publication 2.

We also introduced a computational scheme based on the Monte Carlo method and applicable to BF systems with low or medium loads (Publication 2). Instead of the state space being recursively gone through in order, it is sampled randomly and the average throughput is calculated. Even in large systems, the approach converges relatively quickly at moderate loads, allowing non-trivial networks to be analyzed.

4 LOAD BALANCING IN PROCESSOR SHARING MODELS

4.1 Introduction

In this chapter, we discuss load balancing, which has important applications, e.g., in computer and telecommunications systems. Many communication networks and computer systems use load balancing to improve performance and resource utilization. The ability to divide the service demands efficiently between the resources of the system can have a significant effect on the performance.

The most basic load balancing scenario consists of a single source of service demands and two parallel servers. Even in this simple setting, the problem can be formulated in many ways and the optimal policy depends on various system parameters. While the greedy policy of always joining the shortest queue is the best in many cases [EVW80, TPSC92], it is not always the optimal policy [Whi86]. All in all, sensitivity to detailed system characteristics makes the solving of the optimal load balancing policy difficult even in simple scenarios.

In this chapter, we discuss load balancing in two settings. In Section 4.3, we analyze load balancing between parallel DPS queues. In Section 4.4, balanced fairness is used in the load balancing analysis of fixed data networks.

When we study DPS queues, the system consists of parallel servers and arriving customers. On the basis of the results of Publication 3 on DPS queues, we study the effect of load balancing policy on the mean queue length. We use value extrapolation to compare three heuristic policies to the optimal policy obtained with policy iteration. In addition, we use so-called first policy iteration method to obtain well-performing policies that can be determined without heavy computations.

Similarly to balanced capacity allocations, routing in a network can be balanced, leading to insensitivity to detailed traffic characteristics. In Publications 2, 5 and 6, we studied flow- and packet-level load balancing in fixed networks on the basis of the insensitivity results of Bonald and Proutière [BP03a]. In packet-level balancing, an ongoing flow can be divided between multiple routes. In flow-level balancing, we assume that an arriving flow is routed to one of the routes and the same route is used until the flow is finished. We formulate the problem as an LP problem in both cases and provide numerical results.

4.2 Related Research

In this thesis, we discuss insensitive load balancing policies based on Publications 5 and 6. When BF systems are considered, there is little work discussing load balancing. When flow-level balancing is used and the capacity allocation is fixed to some balanced allocation, the network is insensitive if and only if the routing is balanced [BP02]. In [BJP04], Bonald et al. present a method to determine optimal insensitive routing with local information when capacity allocation and routing are balanced separately. For a variety of objective functions, including blocking probability, the optimal policy is simple, i.e. there is only one local state where traffic is rejected. Finding the optimal policies is straightforward and fast. Instead of separately balancing capacity allocation and routing, better performance is achieved while retaining insensitivity if the allocation and routing are balanced jointly as noted in [BP02]. Jonckheere and Virtamo applied this idea to the flow-level load balancing problem with one traffic class and local information [JV05]. Jonckheere has compared insensitive load balancing to sensitive balancing analytically in the case of infinite state space in [Jon06].

4.3 Discriminatory Processor Sharing Systems

We study systems consisting of multiple DPS servers with different service capacities. Class weights are queue-specific, i.e. the priorities of traffic classes are not identical in different queues. When a new customer arrives at the system, it is directed to one of the queues according to the load balancing policy, which defines how arriving customers are divided between the queues at each system state. The aim is to balance the load between the servers so that the performance of the system is optimal. In case of selfish users optimizing their own performance, the problem could be studied using game theory, see, e.g., [ORS93]. Specifically, we are interested in minimizing the mean total queue length of the system.

We compare three heuristic policies with the optimal policy obtained using policy iteration. Value extrapolation is used in the performance evaluation. As seen in Section 3.3, value extrapolation gives exact results when applied to a single DPS queue with Poissonian arrivals, hence it can be expected that accurate results are also obtained in systems consisting of multiple DPS queues. In addition, relative values of a single DPS queue can be solved with value extrapolation and this result can be used to approximate the optimal policy using so-called first policy iteration approach.

Load Balancing between Parallel DPS Queues

The studied system is defined as follows. The number of parallel servers is denoted L. The are K customer classes. The state of the system is denoted with a vector $\mathbf{x} = (x_{1,1}, \dots, x_{1,K}, \dots, x_{L,1}, \dots, x_{L,K})^{\mathrm{T}}$, where $x_{s,k}$ is the number of class-k customers at queue s, $s = 1, \dots, L$. State space of the process is $\mathcal{S} = \{\mathbf{x} \mid \mathbf{x} \geq \mathbf{0}\}$.

The capacity allocated to a customer depends on the system parameters and on the system state. Each queue allocates capacity independently depending only on the number of customers at that queue. Capacity allocated for class-k customers at queue s at state $\mathbf{x}$ is

$$\phi_{s,k}(\mathbf{x}) = \frac{w_{s,k} x_{s,k}}{\sum_i w_{s,i} x_{s,i}} C_s, \tag{4.1}$$

where $w_{s,k}$ is the weight corresponding to class k at server s and C_s is the capacity of server s.

Arrival process of each customer class is assumed Poissonian and the arriving customers are divided into different queues depending on the load balancing policy. The arrival intensity of class k is denoted λ_k. Load balancing policy α is a function that defines the routing at each system state $\alpha(\mathbf{x}) = (\lambda_{1,1}(\mathbf{x}), \ldots, \lambda_{1,K}(\mathbf{x}), \ldots, \lambda_{L,1}(\mathbf{x}), \ldots, \lambda_{L,K}(\mathbf{x}))^{\mathrm{T}}$, where class-$k$ arrival intensity at server s is denoted $\lambda_{s,k}(\mathbf{x})$. Policies can be categorized to static and dynamic ones. If the routing decisions depend on the system state $\mathbf{x}$, a policy is dynamic, otherwise static. We assume that all customers are accepted, i.e. $\sum_s \lambda_{s,k}(\mathbf{x}) = \lambda_k \ \forall \mathbf{x}$. Service requirements of class-k customers are exponentially distributed with mean $1/\mu_k$.

Prior results concerning a single DPS queue can be used in the analysis of more complex systems. The mean queue length of a single DPS queue with Poissonian arrivals and exponential service requirements can be solved from a system of linear equations [FMI80]. If a static load balancing policy is used, the queues are independent and the total queue length can be determined as a sum of the queue lengths of the individual queues. More importantly, in Section 3.3 we showed that the relative values are polynomial functions of the state $\mathbf{x}$. This result can be used to approximate the optimal policy using first policy iteration.

Heuristic Policies

We present three dynamic heuristic policies that can be used in the balancing.

Shortest queue (SQ) An arriving customer is routed to the queue with the least customers. If several queues are of equal length, the one with the highest capacity is used.

Least expected work (LEW) An arriving customer is routed to the queue with the least amount of expected work proportional to the capacity, i.e. the queue with the lowest value $\sum_k x_{s,k}/\mu_k/C_s$.

Maximal capacity (MC) From a customers point of view, the best queue is the one providing the highest capacity, i.e. the one with highest value $\phi_{n,k}(\mathbf{x}+\mathbf{e}_{s,k})$, where $\mathbf{e}_{s,k}$ is the unit vector corresponding to server s and class k. MC policy maximizes the quality of service experienced by the customer on the short term, but the greedy behavior may use the resources of the system inefficiently leading to poor long term performance.

Optimal Policies

In addition to the heuristic policies, we use two optimized policies. Optimal stochastic routing (SR) refers to the best static policy. Each flow class is stochastically routed among the servers and the state-independent probabilities are chosen to minimize the mean total queue length.

Using the theory of Markov decision processes, the optimal dynamic load balancing policy (OP) can be determined. In each state, an arriving customer is directed to the queue that minimizes the expected total queue length over an infinite time horizon. We optimize the load balancing using policy iteration algorithm described in Section 3.2 as it allows us to truncate the infinite state space using value extrapolation. State spaces of the systems studied here are infinite, and hence the relative values needed by the algorithm cannot be solved. This can be avoided by using value extrapolation at each iteration round, thus reducing the infinite state space into a finite one while still getting an approximation of the optimal mean queue length. The downside of this approach is that the relative values are only solved in the truncated state space. As the policy is defined using the relative values at step 3 of the algorithm, the approximative optimal policy is only known in the truncated state space limiting its usefulness in practice.

First Iteration Step Policy

The last studied policy is the policy obtained using only one iteration round of the policy iteration algorithm. If the relative values of a system can be derived using some policy, first policy iteration approach can be used to get a policy approximating the optimal one without the computational burden needed in solving the Howard equations. Instead of iterating until the optimum is found, only one iteration round is taken using step 3 of the algorithm starting from the initial policy. While not optimal, the results of the first round are often very good. However, the results depend significantly on the choice of the initial policy, see, e.g., [Koo98].

Static policies are good choices for an initial policy, because the results concerning a single DPS queue can be used. Relative values of states in DPS systems with static policies are known, because the queues are independent (probabilistic splitting of Poisson arrival process leads to independent Poisson processes) and relative values of a single DPS queue can be determined as discussed in Section 3.3. However, the choice of the initial static policy affects the quality of the first iteration policy, hence it should be selected carefully. The exact relative values of the system are known, hence the first iteration policy is defined in the whole infinite state space and the policy can be applied in practice. In contrast, the policy obtained with value extrapolation and complete iteration is only defined in the truncated state space, hence it can only be used to approximate the performance.

Numerical results

In this section, we demonstrate our approach using numerical examples. We study a system with two DPS servers and two customer classes and provide various results illustrating the accuracy of the methods used and the performance of the different policies. As a specific example we study a system with parameter values $C_1 = 3/2$, $C_2 = 2/3$, $w_{1,1} = 9$, $w_{1,2} = 1$, $w_{2,1} = 1$, $w_{2,2} = 9$, $\lambda_1 = 5\lambda_2$, $\mu_1 = 10$, and $\mu_2 = 2$. The arrival rates are varied in order to study the effect of system load on the results. Total queue length is used as a performance metric.

Accuracy of the results obtained using value extrapolation depends on the size of the truncated space. In general, the more states are used the more accurate results. We truncate the state space symmetrically, i.e. the truncated state space is of the form $\tilde{\mathcal{S}} = \{\mathbf{x} \mid \mathbf{0} \leq \mathbf{x} \leq (N, N)^{\mathrm{T}}\}$, where N is referred to as truncation point. Figure 4.1 illustrates the convergence with different load balancing policies with system load 0.7. For comparison, convergence of the SQ policy is also illustrated without value extrapolation, i.e. the mean queue length is solved in the truncated state space and the rest of the state space is neglected. Regardless of the policy, the mean

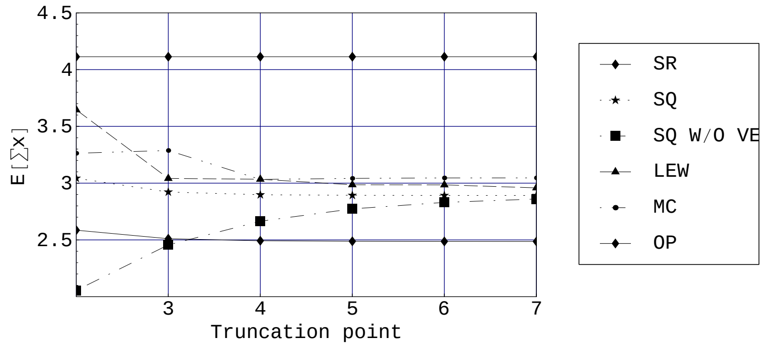


Figure 4.1: Convergence of mean occupancy as a function of the truncation point using different policies.

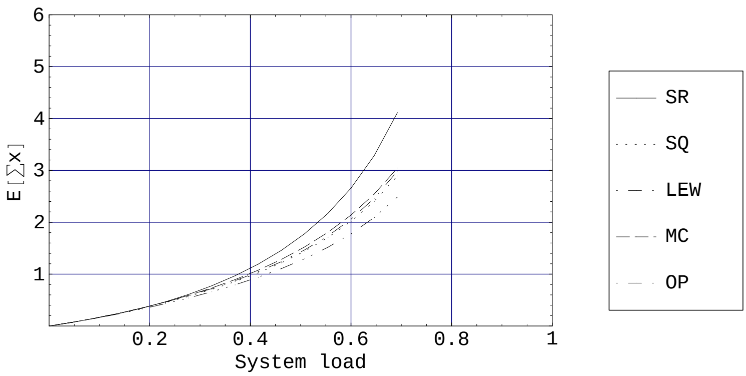


Figure 4.2: Mean queue length as a function of system load.

queue length converges quickly when value extrapolation is used. With the relatively high load 0.7, the results do not get significantly more accurate when the truncation point is increased from 5, while without extrapolation the point should be over 10 for similar accuracy. It seems that even though the results are not exact with systems consisting of several DPS queues, value extrapolation works very well allowing heavy truncation of state space without significant loss of accuracy. The higher the system load, the larger truncated state space is needed for accurate results.

Next, we illustrate the performance of the different policies. Mean queue lengths of the policies are illustrated in Figure 4.2 as a function of the system load. The results are computed using truncation point 6. As seen in Figure 4.1, the results should approximate the infinite state space very accurately. With low loads, there are no significant differences in the mean queue lengths. As the load increases, the differences of the policies become more obvious. The optimal policy always outperforms the other policies and the optimal static policy always performs worse than the heuristic policies. Regardless of the load, the shortest queue policy outperforms the other heuristic policies, though the differences are small.

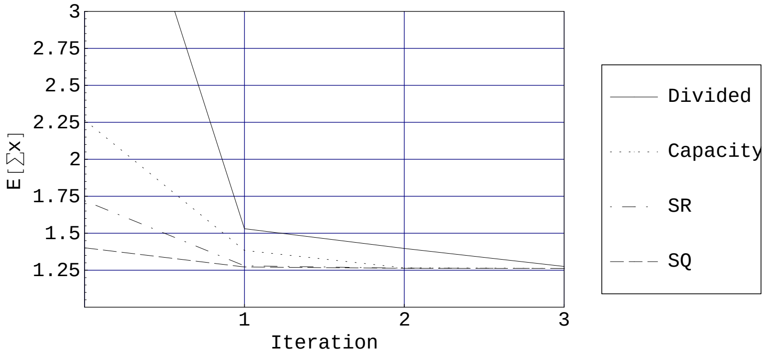


Figure 4.3: Convergence of policy iteration depends on the initial policy. Policy iteration is carried using truncation point 6. Three static policies are compared to SQ policy as the initial policy.

Finally, we study first iteration policy, especially the effect of the initial policy. It is known that the initial policy may affect the convergence of policy iteration algorithm significantly. Figure 4.3 illustrates the effect of the starting policy with system load 0.5. Four initial policies are used. "Divided" routes class 1 to server 1 and class 2 to server 2. "Capacity" stochastically routes the customers in proportions relative to the server capacities. SR is the optimal stochastic routing policy. The best of the heuristic policies, SQ, is used as a point of comparison. Regardless of the starting policy, the iteration ends up close to optimum in three rounds. It can also be seen that with a suitable initial policy even the first iteration round ends very close to the optimum. Another important observation is that the first round policy starting from a sufficiently good static policy outperforms the heuristic policies. When starting from a static policy, the first iteration round can be conducted without any significant computational effort as the relative values are known as seen in Section 3.3, hence the observation has also practical value.

4.4 Fixed Networks

In this section, we study load balancing in fixed networks conducted at two different levels. When flow-level balancing is considered, each arriving flow is routed to a path and the same path is used until the transfer is completed. On the other hand, packet-level balancing may divide a single flow between multiple routes and the routes may be changed dynamically as other flows arrive and depart in the network.

We consider insensitive load balancing, i.e. the steady state probabilities and performance metrics derivable from them do not depend on the flow size distribution. When packet-level balancing is considered, the problem reduces to the balanced fairness recursion defined by (2.8) and can be solved using network flow algorithms. When flow-level balancing is considered, the routing as well as capacity allocation needs to be balanced. In the flow-level setting, we assume that the network has an access control mechanism limiting the number of flows. While the assumption is restrictive, it can be argued that such a mechanism would be beneficial [MR00, Rob04, MR99].

We do not consider implementation issues related to the approaches. Packet-level balancing may mix up the packet order as delays on different routes usually differ, hence it causes problems with TCP transfers. Load balancing is typically conducted on time scales slower than what we are studying, hence the flow-level balancing approach is not implementable in currently deployed systems.

Packet-Level Balancing

First, we study load balancing in fixed networks, where each flow can use resources on multiple routes. This corresponds to packet-level load balancing. This way, the resources of the network are used more efficiently and the performance is better than when only one route is used. The problem is formulated using two different approaches. First, we assume that each traffic class uses a set of predefined routes. Second, we assume that the traffic classes can utilize all the possible routes in the network. While the first approach has more practical value, for example in modeling of load balancing, the second approach gives an theoretically interesting upper limit of the performance. In both cases, the balanced fairness recursion (2.8) can be formulated as a network flow problem. In addition, the capacity limit $\hat{\boldsymbol{\rho}}$ can be derived using the same problem formulation.

First, we formulate the balanced fairness recursion corresponding to (2.8) in a network with K traffic classes. Each class has one or more predefined routes. System state is denoted by a vector $\mathbf{x} = (x_1, \ldots, x_K)^{\mathrm{T}}$, where x_k is the number of active class-k flows. $\boldsymbol{\phi}$ now denotes the flow matrix, where element $\phi_{i,r}$ is the amount of class-i traffic on route r. $\mathbf{R}$ is the routing matrix, where element $R_{r,j}$ is 1 if route r uses link j and 0 otherwise. Vector $\mathbf{c}$ contains the link capacities. Starting from $\Phi(\mathbf{0}) = 1$, the balanced fairness recursion can be solved as an LP problem:

$$\begin{aligned} \Phi(\mathbf{x})^{-1} = \max_{\boldsymbol{\phi}} &\, \alpha, \\ \text{s.t. } & \boldsymbol{\phi}\,\mathbf{e} = \alpha\, \tilde{\boldsymbol{\Phi}}(\mathbf{x}), \\ & \mathbf{e}^{\mathrm{T}} \boldsymbol{\phi}\, \mathbf{R} \le \mathbf{c}^{\mathrm{T}}, \\ & \boldsymbol{\phi} \ge \mathbf{0}, \end{aligned} \tag{4.2}$$

where $\tilde{\boldsymbol{\Phi}}(\mathbf{x}) = (\Phi(\mathbf{x}-\mathbf{e}_1), \ldots, \Phi(\mathbf{x}-\mathbf{e}_K))^{\mathrm{T}}$ and $\Phi(\mathbf{x}) = 0$ for all $\mathbf{x} \notin \mathbb{Z}_+^K$. The first constraint ensures that the capacity allocation is balanced and the second constraint is the link capacity constraint.

If the traffic classes can use all possible routes in the network, the previous approach is not feasible as the number of possible routes explodes with the size of the network. However, the BF recursion step problem corresponds to the well-known multicommodity flow problem (see, e.g., [AMO93]) and enumeration of all the paths can be avoided by formulating the problem in the linear programming setting using directed links. In our formulation, the directed links have individual capacities, but the problem can be modified allowing a shared capacity between the links in opposite directions. Define the link-node incidence matrix $\mathbf{S}$ such that element $S_{j,n}$ has the value -1 if link j originates from node n, 1 if the link ends in node n, and 0 otherwise. $\mathbf{D}$ is the divergence matrix, i.e. the difference between the incoming and outgoing traffic in each node. Element $D_{i,n}$ is ± 1 if n is the source (destination) node of class i and 0 otherwise. Matrix $\boldsymbol{\phi}$ now contains the link traffics. Element $\phi_{i,j}$ is the class-i traffic on link j. The problem formulation reads

$$\begin{aligned} \Phi(\mathbf{x})^{-1} = \max_{\boldsymbol{\phi}} \alpha, & \\ \text{s.t. } \boldsymbol{\phi}\,\mathbf{S} = \alpha \text{ diag}(\tilde{\boldsymbol{\Phi}}(\mathbf{x}))\,\mathbf{D}, & \\ \mathbf{e}^{\mathrm{T}}\boldsymbol{\phi} \leq \mathbf{c}^{\mathrm{T}}, & \end{aligned} \tag{4.3}$$

where diag($\mathbf{a}$) denotes the matrix with the elements of the vector $\mathbf{a}$ on the diagonal and 0 elsewhere. The first constraint is the divergence constraint expressing the balance of incoming and outgoing traffic of each class at each node, and the second is the link capacity constraint.

The asymptotic performance analysis method discussed in Section 3.4 can be applied straightforwardly to the packet-level balancing setting. Balance function values needed in equations (3.23) for calculating the low load derivatives (3.22) are solved using equations (4.2) or (4.3). Given a traffic profile $\mathbf{p}$ defining the proportions of the loads in different classes, the capacity limit of the network can be determined by solving the optimal α for the LP problems (4.2) and (4.3) with $\tilde{\boldsymbol{\Phi}}(\mathbf{x})$ being replaced by $\mathbf{p}$. The end point $\hat{\boldsymbol{\rho}}$ of the load line is then given by $\hat{\boldsymbol{\rho}} = \alpha\mathbf{p}$.

Example Consider the fixed network illustrated in Figure 4.4. The network has 8 nodes and 19 (undirected) links with unit capacities and an equal amount of traffic between all the 28 node pairs. Both packet-level formulations are studied. For predefined routes, we choose two shortest link-disjoint routes between every node pair. The throughput between the extremes is sketched using a cubic polynomial fitted to the end points and to the first two derivatives at $r = 0$. The low load derivatives fit the curve well, but the heavy load derivative is omitted from the fitting as it does not improve the results. The curve corresponding to arbitrary routing outperforms the one with predefined routes. This is expected as the network resources are utilized more efficiently.

Flow-Level Balancing

Next, we discuss load balancing in fixed networks at flow level. The approach is based on the insensitivity results by Bonald and Proutière in [BP02]. In Publications 5 and 6, we presented a method for determining the optimal insensitive flow-level load balancing policy. Using the theory of Markov decision processes, the problem is formulated and solved as a linear programming (LP) problem. The MDP-LP formulation is also used to formulate the more general problem with jointly balanced capacity allocation and routing. A new feature is that whereas in the ordinary LP formulation of the MDP theory the global balance conditions (2.1) appear as linear constraints on the decision variables, now, in order to retain insensitivity, we impose the stricter detailed balance conditions as constraints.

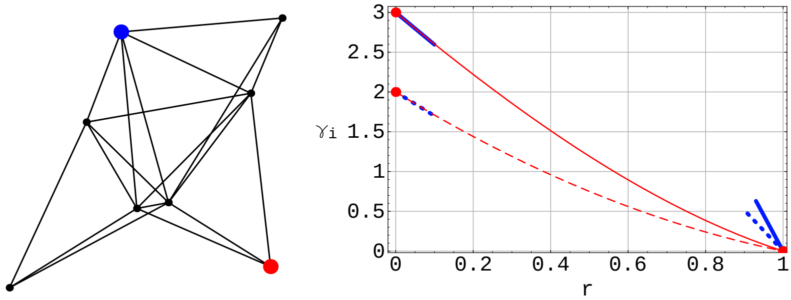


Figure 4.4: Multipath network (left) and throughput asymptotics of the class denoted by the bigger nodes (right). Dotted lines illustrate routing with predefined routes, solid lines arbitrary routing.

Flow-level load balancing can be modeled with a processor sharing network similar to the one in Chapter 2. Again, EPS queues correspond to traffic classes and customers to flows. We now assume that a customer who has received the requested service always exits the system. In the queueing network discussed in Chapter 2, in contrast, a serviced customer could continue to another node with a given probability. When the queueing model is applied to data networks, we now have to assume Poissonian flow arrivals instead of the less restrictive assumption of Poissonian session arrivals.

Similarly to the model in Chapter 2, we study a queueing network with K nodes. In contrast to the model with static routing, we now assume that the arrival intensities at the nodes depend on the system state. Arrival intensity at node i in state $\mathbf{x}$ is denoted $\lambda_i(\mathbf{x})$. Service rate at node i is $\phi(\mathbf{x})$ and the mean service requirement is σ_i.

If the service rates are balanced by some function Φ, a network with state-dependent arrival rates is balanced if and only if the arrival rates satisfy balance conditions [BP02]

$$\frac{\lambda_i(\mathbf{x}+\mathbf{e}_j)}{\lambda_i(\mathbf{x})} = \frac{\lambda_j(\mathbf{x}+\mathbf{e}_i)}{\lambda_j(\mathbf{x})} \quad \forall i, j. \tag{4.4}$$

The balance conditions are equivalent to the existence of a balance function Λ so that $\Lambda(\mathbf{0}) = 1$ and

$$\lambda_i(\mathbf{x}) = \frac{\Lambda(\mathbf{x}+\mathbf{e}_i)}{\Lambda(\mathbf{x})} \quad \forall \mathbf{x}, i. \tag{4.5}$$

In this case, the steady-state distribution of the process is

$$\pi(\mathbf{x}) = \frac{\Phi(\mathbf{x})\Lambda(\mathbf{x})}{G(\boldsymbol{\rho})}, \tag{4.6}$$

where $G(\boldsymbol{\rho})$ is the normalization constant.

More general results are obtained if the routing is balanced jointly with the capacity allocation. In this case, a network is insensitive if and only if the function $\psi_i(\mathbf{x})$ defined as

$$\psi_i(\mathbf{x}) = \frac{\lambda_i(\mathbf{x}-\mathbf{e}_i)\sigma_i}{\phi_i(\mathbf{x})} \tag{4.7}$$

is balanced [BP02]:

$$\frac{\psi_i(\mathbf{x}-\mathbf{e}_j)}{\psi_i(\mathbf{x})} = \frac{\psi_j(\mathbf{x}-\mathbf{e}_i)}{\psi_j(\mathbf{x})} \quad \forall i, j, x_i > 0, x_j > 0. \tag{4.8}$$

Balance condition (4.8) is equivalent to the existence of a balance function Ψ so that $\Psi(0) = 1$ and

$$\psi_i(\mathbf{x}) = \frac{\Psi(\mathbf{x}-\mathbf{e}_i)}{\Psi(\mathbf{x})} \quad \forall \mathbf{x} : x_i > 0, i. \tag{4.9}$$

Balance condition (4.8) is also equivalent to detailed balance conditions, which can be formulates as follows:

$$\lambda_i(\mathbf{x})\pi(\mathbf{x}) = \frac{\phi_i(\mathbf{x}+\mathbf{e}_i)}{\sigma_i}\pi(\mathbf{x}+\mathbf{e}_i) \quad \forall \mathbf{x}, i. \tag{4.10}$$

The steady state distribution of the system is

$$\pi(\mathbf{x}) = \frac{1}{G(\boldsymbol{\rho})\Psi(\mathbf{x})}. \tag{4.11}$$

The insensitivity results concerning queueing networks can be used to analyze load balancing in data networks carrying elastic traffic. Optimization of flow-level balancing is a more difficult problem than the packet-level balancing problem. Instead of optimizing only the capacity allocation, also the routing probabilities need to be considered. The arrival process can be balanced either separately or jointly with the capacity allocation. Separate balancing is easier as the separate problems are smaller than the joint problem. On the other hand, separate balancing is more restrictive hence performance is worse than with jointly balanced allocation and routing.

When fixed routes or packet-level load balancing is studied, the capacity allocation policy maximizing the utilized capacity can be determined recursively one state at a time. The ratios of the capacities are given and the amount of allocated capacity is decided. The obvious policy choice is to maximize the allocated capacity in every state. When flow-level balancing is considered, the situation is not as simple. The amount of arriving traffic is given and the routing probabilities among the different routes need to be decided. We assume that the network has an access control mechanism that guarantees a minimum bit rate $\phi_k^{\min}$ for accepted class-k flows. We use the blocking probability as the optimization criterion. However, the formulations allow the use of other objective functions.

Separately balanced capacity allocation and routing First, we consider the case with separately balanced routing and capacity allocation. Similarly to the packet-level balancing problem with predefined routes, each traffic class has one or more possible routes. State vector $\mathbf{x}$ contains elements x_k^r denoting the number of active class-k flows on route r. The arrival intensity of class k is λ_k and it is stochastially divided between the routes. Class-k arrival intensity on route r in state $\mathbf{x}$ is denoted $\lambda_k^r(\mathbf{x})$. In order to retain insensitivity, the arrival intensities $\lambda_k^r(\mathbf{x})$ need to satisfy the balance condition (4.4).

The routing problem can be solved using the LP formulation of MDPs discussed in Section 3.2. State of the process consists of the network state and the routing decision. The routing vector is denoted $\mathbf{d} = (d_1, \ldots, d_K)^{\mathrm{T}}$, where $d_k = r$ if class-k traffic is directed to route r and $d_k = 0$ if the class is blocked. $\phi_k^r(\mathbf{x})$ is the (balanced) capacity allocated for class-k on route r. $\pi(\mathbf{x}, \mathbf{d})$ is a decision variable in the LP problem and corresponds to the probability that the network is in state $\mathbf{x}$ and routing $\mathbf{d}$ is used. In the ordinary LP formulation of the MDP theory, global balance conditions (2.1) appear as linear constraints on the decision variables. In order to retain insensitivity, we impose the stricter detailed balance conditions (4.10) as constraints.

The objective function of the LP problem is the total blocking probability. The probability that the system is in state $\mathbf{x}$ and blocking class-k traffic is $\sum_{d:d_k=0} \pi(\mathbf{x}, \mathbf{d})$ hence the total blocking probability is

$$\sum_k \frac{\lambda_k}{\lambda} \sum_{\mathbf{x}} \sum_{\mathbf{d}:d_k=0} \pi(\mathbf{x}, \mathbf{d}), \tag{4.12}$$

where $\lambda = \sum_k \lambda_k$ is the total arrival rate. Using this notation the MDP-LP formulation of the problem reads

$$\min_{\pi(\mathbf{x},\mathbf{d})} \sum_k \frac{\lambda_k}{\lambda} \sum_{\mathbf{x}} \sum_{\mathbf{d}:d_k=0} \pi(\mathbf{x}, \mathbf{d}), \tag{4.13}$$

$$\text{s.t. } \lambda_k \sum_{\mathbf{d}:d_k=r} \pi(\mathbf{x}, \mathbf{d}) = \frac{\phi_k^r(\mathbf{x} + \mathbf{e}_k^r)}{\sigma_k} \sum_{\mathbf{d}} \pi(\mathbf{x} + \mathbf{e}_k^r, \mathbf{d}) \quad \forall \mathbf{x}, k, r \in R_k, \tag{4.14}$$

$$\sum_{\mathbf{x}} \sum_{\mathbf{d}} \pi(\mathbf{x}, \mathbf{d}) = 1, \tag{4.15}$$

$$\pi(\mathbf{x}, \mathbf{d}) \geq 0 \quad \forall \mathbf{x}, \mathbf{d}, \tag{4.16}$$

where (4.14) is the detailed balance condition ensuring insensitivity and $\mathbf{e}_k^r$ is the unit vector corresponding to class-k flows on route r.

When the problem has been solved, the obtained values $\pi(\mathbf{x}, \mathbf{d})$ can be used to analyze the optimal policy. For example, the state probabilities are

$$\pi(\mathbf{x}) = \sum_d \pi(\mathbf{x}, \mathbf{d}), \tag{4.17}$$

and the class-k arrival rates at the different routes in state $\mathbf{x}$ are

$$\lambda_k^r(\mathbf{x}) = \frac{\sum_{\mathbf{d}:d_k=r} \pi(\mathbf{x}, \mathbf{d})}{\sum_{\mathbf{d}} \pi(\mathbf{x}, \mathbf{d})} \lambda_k. \tag{4.18}$$

Next, we consider the same load balancing setting but the routing is balanced jointly with capacity allocation.

Jointly balanced capacity allocation and routing In the previous section, capacity allocation was assumed to be separately balanced and fixed in advance and only routing was optimized. Better results can be obtained if routing and capacity allocation are balanced jointly.

The problem can be formulated and solved as an MDP-LP problem. In the jointly balanced problem, the decisions consist of capacity allocation and routing decisions. For each active route in the network, the constraining link is identified. Let C^r be the maximum feasible capacity on route r, i.e. the capacity of the link with the lowest capacity along the route. The allocation decisions are modeled with a binary vector $\mathbf{b} = (b_1^1, \ldots, b_1^{|R_1|}, \ldots, b_K^1, \ldots, b_K^{|R_K|})^{\mathrm{T}}$, where R_k is the set of routes available for class k. $b_k^r = 1$ if capacity C^r is allocated to class k on route r and 0 if no capacity is allocated.

The decision variable in the LP problem, $\pi(\mathbf{x}, \mathbf{d}, \mathbf{b})$, corresponds to the probability that the system is in state $\mathbf{x}$, routing vector $\mathbf{d}$ is used, and capacity allocation is $\mathbf{b}$. When routing is considered, the accepted traffic may not exceed the offered traffic. The problem formulation takes this constraint into account implicitly. When capacity allocation is considered, the allocated capacity on any link may not exceed the capacity of the link. In addition to the detailed balance constraints, the capacity constraints need to be added explicitly to the problem. Additional constraints are also needed to guarantee the minimum bit rate $\phi_k^{\min}$ for the accepted flows. The MDP-LP formulation of the problem reads

$$\min_{\pi(\mathbf{x},\mathbf{d},\mathbf{b})} \sum_k \frac{\lambda_k}{\lambda} \sum_{\mathbf{x}} \sum_{\mathbf{d}:d_k=0} \sum_{\mathbf{b}} \pi(\mathbf{x}, \mathbf{d}, \mathbf{b}), \tag{4.19}$$

$$\text{s.t. } \lambda_k \sum_{\mathbf{d}:d_k=r} \sum_{\mathbf{b}} \pi(\mathbf{x}, \mathbf{d}, \mathbf{b}) = \frac{C^r}{\sigma_k} \sum_{\mathbf{d}} \sum_{\mathbf{b}:b_{d,r}=1} \pi(\mathbf{x} + \mathbf{e}_k^r, \mathbf{d}, \mathbf{b}) \quad \forall \mathbf{x}, k, r \in R_k \tag{4.20}$$

$$\sum_k \sum_{r \in R_k : l \in r} C^r \sum_{\mathbf{d}} \sum_{\mathbf{b}:b_k^r=1} \pi(\mathbf{x}, \mathbf{d}, \mathbf{b}) \leq C_l \sum_{\mathbf{d}} \sum_{\mathbf{b}} \pi(\mathbf{x}, \mathbf{d}, \mathbf{b}) \quad \forall \mathbf{x}, l, \tag{4.21}$$

$$x_k^r \phi_k^{\min} \sum_{\mathbf{d}} \sum_{\mathbf{b}} \pi(\mathbf{x}, \mathbf{d}, \mathbf{b}) \leq C^r \sum_{\mathbf{d}} \sum_{\mathbf{b}:b_k^r=1} \pi(\mathbf{x}, \mathbf{d}, \mathbf{b}) \ \forall \mathbf{x}, k, r \in R_k, \tag{4.22}$$

$$\sum_{\mathbf{x}} \sum_{\mathbf{d}} \sum_{\mathbf{b}} \pi(\mathbf{x}, \mathbf{d}, \mathbf{b}) = 1, \tag{4.23}$$

$$\pi(\mathbf{x}, \mathbf{d}, \mathbf{b}) \geq 0 \quad \forall \mathbf{x}, \mathbf{d}, \mathbf{b}, \tag{4.24}$$

where C_l is the capacity of link l, (4.20) represents the detailed balance condition, (4.21) the link capacity constraints, and (4.22) the minimum bit rate constraints.

The optimal $\pi(\mathbf{x}, \mathbf{d}, \mathbf{b})$ values can be used to analyze the capacity allocation and routing policy. State probabilities are

$$\pi(\mathbf{x}) = \sum_{\mathbf{d}} \sum_{\mathbf{b}} \pi(\mathbf{x}, \mathbf{d}, \mathbf{b}). \tag{4.25}$$

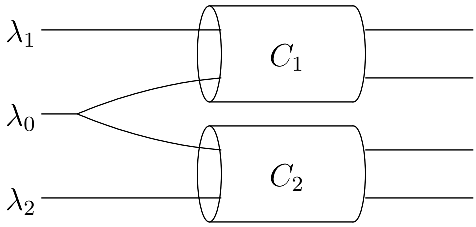


Figure 4.5: Example network

In state $\mathbf{x}$, class-k arrival intensity on route r is

$$\lambda_k^r(\mathbf{x}) = \frac{\sum_{\mathbf{d}:d_k=r} \sum_{\mathbf{b}} \pi(\mathbf{x}, \mathbf{d}, \mathbf{b})}{\sum_{\mathbf{d}} \sum_{\mathbf{b}} \pi(\mathbf{x}, \mathbf{d}, \mathbf{b})} \lambda_k \tag{4.26}$$

and the capacity allocated for class k on route r is

$$\phi_k^r(\mathbf{x}) = \frac{\sum_{\mathbf{d}} \sum_{\mathbf{b}:b_k^r=1} \pi(\mathbf{x}, \mathbf{d}, \mathbf{b})}{\sum_{\mathbf{d}} \sum_{\mathbf{b}} \pi(\mathbf{x}, \mathbf{d}, \mathbf{b})} C^r. \tag{4.27}$$

While the problem can be formulated as an LP problem, the computational burden of solving the problem limits the flow-level approach to small instances. In contrast, packet-level balancing discussed earlier can be solved recursively allowing analysis of bigger networks. Next, we present numerical results and compare the different methods.

Numerical Results In this section, we compare the different insensitive load balancing methods in a simple network illustrated in Figure 4.5. There are two parallel links with capacities C_1 and C_2 and three traffic classes. An adaptive traffic class can utilize both the links while the links also receive dedicated background traffic.

The offered arrival rate of the adaptive class is λ_0 and the rates of the background traffic classes are λ_1 and λ_2. The number of active background flows on link i is x_i and the number of adaptive flows is x_0. In addition to the total number of adaptive flows, the number of flows in the individual links are needed when static or flow-level load balancing is used. The number of adaptive flows on link i is denoted $x_{0,i}$.

The allocated bit rate on link i is $\phi_{0,i}(\mathbf{x})$ for the adaptive class and $\phi_i(\mathbf{x})$ for the background traffic. If capacity is allocated according to balanced fairness, the capacity of a link is equally shared between all the active flows utilizing that link. The balance function is

$$\Phi(\mathbf{x}) = \frac{\binom{x_1+x_{0,1}}{x_1}\binom{x_2+x_{0,2}}{x_2}}{C_1^{x_1+x_{0,1}} C_2^{x_2+x_{0,2}}} \tag{4.28}$$

and the capacities are $\phi_{0,i}(\mathbf{x}) = \frac{x_{0,i}}{x_{0,i}+x_i} C_i$ and $\phi_i(\mathbf{x}) = \frac{x_i}{x_{0,i}+x_i} C_i$.

We compare different policies using different traffic loads. Each background class is assumed to make up 10% of the total load and the mean

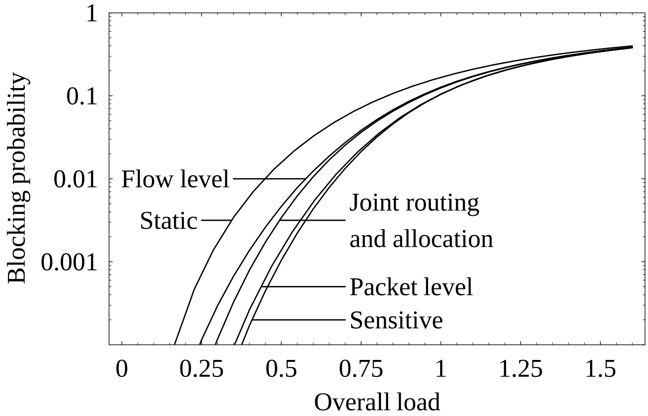


Figure 4.6: Blocking probabilities

flow sizes σ_0, σ_1 and σ_2 are assumed identical. We assume unit link capacities and the minimum bit rate $\phi^{\min}$ is taken to be 0.2 for all the traffic classes.

We compare the blocking probabilities of insensitive static load balancing, flow-level balancing with BF, flow-level balancing with jointly balanced routing and capacity allocation, and packet-level balancing. In order to estimate the performance penalty caused by the insensitivity requirement, we also determined the performance utilizing the optimal sensitive flow-level balancing policy assuming exponentially distributed flow sizes. The optimal sensitive policy accepts all traffic and routes an arriving adaptive class flow to the link that provides more capacity.

Figure 4.6 illustrates the blocking probabilities as a function of the total offered traffic load. Static load balancing has the worst performance as expected. Flow-level balancing with separately balanced routing and capacity allocation outperforms static routing but is the least efficient dynamic policy. If routing and allocation are jointly balanced, slightly better results are achieved. Packet-level balancing is significantly better than flow-level balancing and it performs almost as well as the sensitive flow-level policy. A key factor is the more efficient capacity usage. When packet-level balancing is used, an adaptive flow can utilize the capacity of both the links while with flow-level balancing only one link is used. Especially with low loads this has a significant effect. If an adaptive flow arrives in empty network, it utilizes twice the capacity when compared to flow-level routing. While the packet-level approach utilizes the capacity more efficiently, the sensitive flow-level balancing outperforms it in this example. This is due to the minimum bit rate constraint. The maximum number of flows depends on the capacity allocation policy and in this case the sensitive flow-level policy allows more concurrent flows leading to lower blocking probabilities.

4.5 Summary

In this chapter, we discussed load balancing among multiple resources. Such models have numerous applications in many fields, e.g., in computer and telecommunication systems. We studied two settings: parallel DPS queues and fixed networks with balanced capacity allocation.

When load balancing between multiple DPS queues is studied, value extrapolation can be used to approximate flow-level performance using heavily truncated state spaces. When applied to a single DPS queue, value extrapolation yields exact results, as shown in Section 3.3, and hence accurate results can also be expected in systems consisting of multiple DPS queues. Value extrapolation was used jointly with policy iteration to approximate the optimal performance of such systems. The optimal performance was compared to several heuristic policies and to policies obtained with one iteration round starting from static policies. Numerical experimentations illustrated that the policy obtained with one round of policy iteration with a good initial static policy outperforms the heuristic policies used. As the relative values of DPS systems with static policies can be easily determined, such policies can be defined without heavy computations making them useful for practical applications.

We also studied load balancing in fixed networks using balanced fairness. On the basis of the results of Bonald and Proutière [BP02], we studied insensitive load balancing in data networks executed on either the packet or flow level. In contrast to the DPS model, which is only applicable to parallel servers, the BF approach can be used in networks.

When packet-level balancing is used, the optimal balancing policy can be determined recursively, facilitating the analysis. The balanced fairness recursion can be solved as an LP problem, assuming either a set of fixed routes for each traffic class or arbitrary routing. We used the asymptotic throughput method in performance analysis.

When flow-level balancing is used, an access control mechanism was assumed to guarantee a minimum capacity for all accepted flows and to reject arriving flows if the capacity constraint cannot be satisfied. In addition to balanced capacity allocation, the routing also needs to be balanced in order to obtain insensitive results. The optimal routing policy can be solved either separately or jointly with the capacity allocation. The flow-level balancing problem was formulated using the linear programming formulation of the MDP theory. The global balance equations of the problem were replaced with the more constraining detailed balance equations, thus ensuring insensitivity. The size of the LP problem grows rapidly as the size of the network is increased. In practice, the computational complexity limits the use of the method to small toy networks.

We compared the performance of the different insensitive load balancing methods in a toy network. Flow-level balancing is the least efficient dynamic insensitive policy. The performance is improved if capacity allocation and routing are jointly balanced and optimized. Packet-level balancing outperforms flow-level balancing, regardless of the network load. Still, even in this case, some performance penalty has to be paid for insensitivity.

5 FLOW-LEVEL PERFORMANCE ANALYSIS OF WIRELESS NETWORKS

5.1 Introduction

In this chapter, we model and analyze wireless networks carrying elastic data traffic. We discuss two different types of wireless networks. In cellular networks, users send data to and receive data from fixed base stations and there are no direct transmissions between the users. In wireless multihop networks, there is no fixed infrastructure and the users relay each other's traffic towards the receiving user.

In wireless networks, interactions between simultaneous transmissions make the analysis considerably more difficult than in fixed networks. The capacities of links, i.e. the transmission capacity between two nodes, depend on the state of other links. A transmitting node interferes with the other nodes, reducing their link capacities. Interference phenomena can be modeled in various ways, ranging from simple Boolean models [JPPQ05] to more complex SINR models [Sha49].

We are interested in the flow-level performance of the systems being studied. The performance metrics we use are the mean flow throughput and the mean number of active transmissions (which is proportional to the mean transmission duration). The flow-level time scale is assumed to be longer than packet or medium access scales. The flow-level throughput is assumed to be constant, while on a shorter time scale it may fluctuate. For example, in Aloha networks the throughput at any given moment is stochastic but the expected value can be used in the flow-level analysis. We also assume that when the state of the network changes, the transient phase is short when compared to the flow scale, and hence the capacity changes can be assumed to be instantaneous in the flow-level analysis.

We analyze the performance of wireless networks carrying elastic data traffic. First, we discuss a simple cellular network with link adaptation consisting of two base stations and customers located on a line between them. We model the system and analyze the performance using different capacity allocation policies (Publication 7).

We also study wireless multihop networks using two different MAC schemes and balanced fairness capacity allocation. On the basis of earlier work by Penttinen et al. in [PVJ06], we apply the asymptotic throughput analysis to multihop networks using the STDMA MAC protocol (Publication 2). We also study multihop networks with random access, assuming that the transmission probabilities can be adapted upon flow arrivals and departures (Publication 8). We derive the exact throughput in the two-class scenario. In the general network case, we present an algorithm for optimizing the transmission probabilities and compare the throughput behavior of flow-optimized random access against the throughput obtained by optimal scheduling.

The structure of this chapter is as follows. In Section 5.3, we discuss a simple cellular system with two base stations and link rate adaptation, presented in Publication 7, and compare the flow-level performance of different capacity allocation policies. In Section 5.4, we model and analyze the performance of multihop networks using the spatial time division multiple access (STDMA) protocol [NK85], as discussed in Publication 2. Finally, multihop networks with random medium access are analyzed in Section 5.5 (Publication 8).

5.2 Related Research

Most prior analytical work on wireless networks assume a static user model (see, e.g., [LCS03, Alt02b]), i.e. the number of transmissions is fixed. In typical wireless scenarios, however, a dynamic model is needed to capture the user-level performance. The processor sharing paradigm has recently been applied to wireless networks (see, e.g., [PV02, BP03b, Bor05]) allowing analysis in the dynamic setting.

Aloha networks were first introduced in 1970 [Abr70] and a lot of seminal work on the subject was published already in the 70s and 80s. Kleinrock and Silvester have analyzed the capacity of multihop Aloha networks in many interesting scenarios ([KS78, SK83]). For an example of more recent interest in random access, Baccelli et al. [BaM06] introduced a decentralized random access protocol which attains density of progress equal to the well-known upper bound of wireless networks derived by Gupta and Kumar [GK00].

In [BG92] the authors propose a simple iterative approach for solving the link transmission probabilities to achieve given (feasible) link capacities. However, they do not provide a solution how to determine the maximum feasible link capacities with given capacity proportions, which is needed in performance analysis and one of the contributions of this thesis.

More recently, Wang and Kar [WK06] have considered a rate control problem in random access networks. They determine jointly the end-to-end flow rates and the corresponding transmission probabilities so that the flow rates are proportionally fair. They propose two algorithms which can also be implemented in a distributed manner. However, the authors do not consider the performance of such a scheme in a realistic setting, where file transfers arrive randomly and depart upon completion. Our work can be seen as an approximation of the flow-level performance if such a proportional fair rate control algorithm were applied in the dynamic scenario. The advantage of balanced fairness is that it allows mathematical throughput analysis in the dynamic scenario, which would be overwhelmingly difficult for proportional fairness.

5.3 Cellular Network with Road Topology

First, we study an example cellular system consisting of two base stations. While the system is simple, the methods used are applicable to more complex systems. Two base stations, A and B, are used to serve elastic traffic, or file downloads, destined to users located on a road between the base stations, cf. Figure 5.1. The base station nearest to the user is always used for a connection. The base stations are assumed to work coordinatedly, i.e. the base stations schedule their transmissions in cooperation in order to

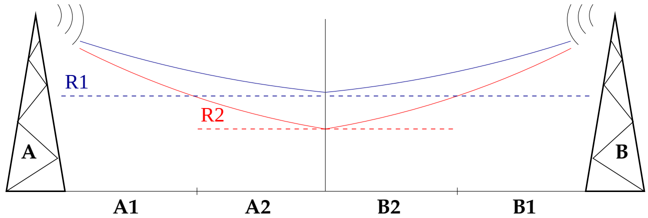


Figure 5.1: Example system with two base stations. Areas A1 and A2 are served by the station A and areas B1 and B2 are served by the station B. If both the stations are active simultaneously the maximum rate at A2 or B2 decreases from R1 to R2 due to interference.

optimize the performance of the whole system.

Link adaptation is modeled as follows. Close to the base stations (in areas A1 and B1) the total downlink rate is always R_1 irrespective of the state of the other station. Further away (in areas A2 and B2) the capacity remains at R_1 only if the other station is not active simultaneously, otherwise the rate decreases to R_2 due to interference.

We describe the system state by the vector $\mathbf{x} = (x_1, x_2, x_3, x_4)^{\mathrm{T}}$, giving the number of active flows in each area A1, A2, B2, B1, respectively. We use the term flow class interchangeably with the term area. The state space of the system is given by $\mathcal{S} = \{\mathbf{x} \mid \mathbf{x} \geq \mathbf{0}\}$. In computations we use a truncated state space which is denoted by $\tilde{\mathcal{S}}$.

As the system state evolves dynamically, we need to fix a policy defining how the network resources are used in any given state. In each state of the system, a policy defines a rate allocation which corresponds to a rate vector $\mathbf{r} = (r_1, r_2, r_3, r_4)$ giving the total rate for each class. The total rate is shared evenly among the flows which belong to a same class by time sharing.

The set of feasible allocations is determined as follows. Let $\mathbf{R}$ be the matrix comprising of column vectors each of which specifies the link capacities of an instantaneous "operation mode" under the constraints described above:

$$\mathbf{R} = \begin{pmatrix} R_1 & 0 & R_1 & 0 & 0 & 0 \\ 0 & R_2 & 0 & R_2 & R_1 & 0 \\ 0 & R_2 & R_2 & 0 & 0 & R_1 \\ R_1 & 0 & 0 & R_1 & 0 & 0 \end{pmatrix}. \tag{5.1}$$

For example, the first column of $\mathbf{R}$ represents the mode where both base stations serve the flows in the nearest class (areas A1 and B1 are being served), the second column represents the mode where both the stations are active and serve the traffic in the center-most areas. The policy rate vector $\mathbf{r}$ can take the form of any column of $\mathbf{R}$ and, additionally, any convex combination of the columns. These are available through time multiplexing, which is assumed to take place on a fast small time scale compared to flow durations. Thus, the available allocations are defined by the convex hull spanned by the column vectors of $\mathbf{R}$.

Alternatively, one may determine the hyperplanes that jointly constrain the feasible values of $\mathbf{r}$. This is just another way to describe the convex hull and can be done using the standard gift wrapping algorithm [Ski97]. The feasible rate vectors $\mathbf{r}$ are constrained by $\mathbf{Dr} \leq \mathbf{e}^{\mathrm{T}}$ ($\mathbf{e}$ is a vector of ones), where, assuming that $\frac{1}{2}R_1 < R_2 < R_1$,

$$\mathbf{D} = \begin{pmatrix} \frac{1}{R_1} & \frac{1}{R_1} & 0 & 0 \\ 0 & 0 & \frac{1}{R_1} & \frac{1}{R_1} \\ \frac{R_2}{R_1^2} & \frac{1}{R_1} & \frac{R_1-R_2}{R_1 R_2} & \frac{R_1-R_2}{R_1^2} \\ \frac{R_1-R_2}{R_1^2} & \frac{R_1-R_2}{R_1 R_2} & \frac{1}{R_1} & \frac{R_2}{R_1^2} \end{pmatrix}. \tag{5.2}$$

In the case that $0 < R_2 < \frac{1}{2}R_1$ the same matrix applies with the exception

$$d_{33} = d_{42} = \frac{1}{R_1}. \tag{5.3}$$

Next we describe the capacity set of the system by finding the maximum traffic load that the system can sustain. Let α and $1-\alpha$ denote the fraction of traffic in A1 and A2, respectively. Assuming symmetry the same fractions hold also for B1 and B2 and the system serves traffic at rate $\mathbf{r} = \frac{R}{2}(\alpha, 1-\alpha, 1-\alpha, \alpha)$, where R is the total capacity to be maximized.

$R_{\max}$ can be obtained by looking at the constraints $\mathbf{D}$, but can also be derived intuitively as follows. Assume that the load is high and all the classes have traffic.

First, let $0 \leq R_2 < \frac{1}{2}R_1$. Now it is advantageous to serve the traffic in areas A2 and B2 so that only one base station is active. Looking at the base station A; it serves area A1 the time α/R_1, area A2 the fraction of time $(1-\alpha)/R_1$ and, additionally, must remain quiet the fraction of time $(1-\alpha)/R_1$ when station B serves area B2. Due to symmetry

$$R_{\max} = 2\frac{\alpha + (1-\alpha)}{2(1-\alpha)+\alpha}R_1 = \frac{2R_1}{2-\alpha}. \tag{5.4}$$

Second, let $\frac{1}{2}R_1 \leq R_2 \leq R_1$. Now the traffic in areas A2 and B2 is served using both the base stations simultaneously. Looking again at the station A; it serves class A1 the time α/R_1, A2 the time $(1-\alpha)/R_2$, but with different rates. Again due to symmetry

$$\begin{aligned} R_{\max} &= 2\left(\frac{\alpha/R_1}{\alpha/R_1 + (1-\alpha)/R_2}R_1 + \frac{(1-\alpha)/R_2}{\alpha/R_1 + (1-\alpha)/R_2}R_2\right) \\ &= \frac{2R_1R_2}{R_1 - R_1\alpha + R_2\alpha}. \end{aligned} \tag{5.5}$$

The performance of the system depends on how the available capacity is divided between the customers. Next we discuss and compare different capacity allocation policies.

Performance under different operational policies

We analyze the performance of the system with different capacity allocation policies. We use three different dynamic policies and compare them to the static policy corresponding to the fixed rates at the capacity limit. The considered policies are:

- The optimal policy maximizing the system throughput
- Max-min fairness
- Balanced fairness

The flow-level performance of the policies are compared. We measure the performance of the system with the mean number of active flows which is proportional to the mean flow duration.

The first studied policy optimizes the system performance and can be determined by utilizing the MDP theory. The policy minimizing the mean file transfer time can be found using policy iteration. When the optimal policy is known, value extrapolation is used to determine the performance. In order to avoid problems caused by the boundary of the state space, the policy iteration is executed using a larger state space than with value extrapolation. Policy iteration is computationally demanding as the Howard equations need to be solved multiple times.

While the policy obtained with MDP maximizes the throughput of the system, it does not explicitly consider how the resources are shared among the users. In practice, this may lead to situations where some users efficiently use up all the system capacity while others are left without an acceptable level of service. To avoid such situations, one may impose additional constraints on the resource sharing to guarantee "fair" capacity allocation. We use max-min fairness as an example policy that takes fairness into account. The customers are treated more evenly than with the system optimal policy. In each state, the bandwidth of the flow with least bandwidth is maximized. The performance of the max-min policy is evaluated using value extrapolation.

The third policy we study is balanced fairness. BF recursion (2.8) can be formulated using the constraint matrix $\mathbf{D}$ (5.2) as

$$\Phi(\mathbf{x}) = \max_i \left\{ (\mathbf{D}\tilde{\mathbf{\Phi}})_i \right\} , \tag{5.6}$$

where $\tilde{\mathbf{\Phi}} = (\Phi(\mathbf{x}-\mathbf{e}_1), \ldots, \Phi(\mathbf{x}-\mathbf{e}_4))^{\mathrm{T}}$. The equilibrium distribution is given by

$$\pi(\mathbf{x}) = \frac{1}{G(\boldsymbol{\rho})}\Phi(\mathbf{x})\rho_1^{x_1}\rho_2^{x_2}\rho_3^{x_3}\rho_4^{x_4}, \tag{5.7}$$

where $G(\boldsymbol{\rho})$ is the normalization constant and ρ_i is the amount of class-i traffic (bit/s). Performance is evaluated using state probabilities of a truncated state space. The mean flow number can then be determined by summing over the truncated state space. BF is significantly faster to evaluate than the value extrapolation method. A significantly larger number of states can be included directly in the analysis to replace the extrapolation need.

We illustrate the performance of the policies with parameter values $R_1 = 5$ and $R_2 = 1$. The traffic intensities are assumed equal in all four regions. When value extrapolation method is used, the state space is truncated so that the maximum number of flows in each class is 6. With BF, the corresponding limit is 30. These limits result in accurate results with the traffic loads used.

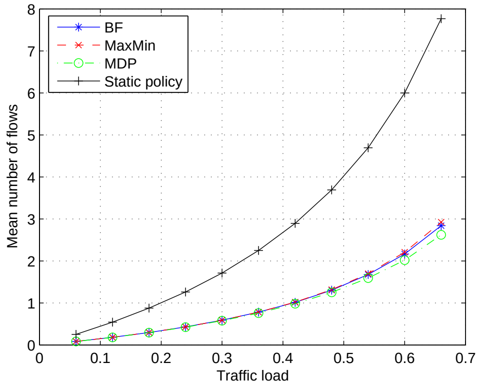


Figure 5.2: Mean number of active flows with parameters $R_1 = 5$ and $R_2 = 1$

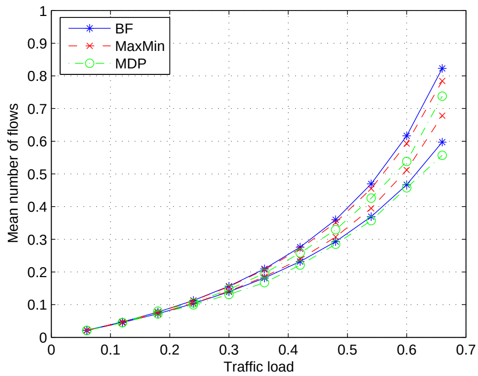


Figure 5.3: Mean number of active flows in classes 1 (lower curves) and 2 (upper curves) with parameters $R_1 = 5$ and $R_2 = 1$

Figure 5.2 illustrates the mean number of active flows with different system loads. The MDP policy has the best performance but the other dynamic policies are almost equal. The static policy of allocating equal bandwidth to all the classes regardless of the system state is significantly worse than the dynamic policies.

The mean number of active flows in traffic classes 1 and 2 are illustrated in Figure 5.3. While MDP policy ensures best service for both the classes, the differences between the policies are relatively small. Max-min allocation provides the most equal service to the classes as expected.

The illustrated results are produced with parameter values $R_1 = 5$ and $R_2 = 1$ which fall into region $0 < R_2 < \frac{1}{2}R_1$. In this case, it is beneficial to turn one base station off while the other one serves the flows in the middle region. In addition, we analyzed a scenario with values $R_1 = 5$ and $R_2 = 4$ when it is more efficient to serve the middle classes simultaneously. While the policies differ from the illustrated scenario, the performance is relatively similar.

Next, we consider wireless multihop networks using time division multiple access MAC-layer. Capacity set of a wireless network is generally difficult to determine due to various physical constraints and the effects of interference in the network. In the following, the wireless MAC layer protocol known as spatial time division multiple access (STDMA) [NK85] is studied. The modeling approach and the corresponding balanced fairness recursion was introduced by Penttinen et al. in [PVJ06]. We apply asymptotic analysis in the same setting.

The STDMA setting is modeled using an approach resembling the one used in the cellular example in the previous section. Sets of links with specified transmission parameters are scheduled on a fast time scale to produce constant (virtual) link capacities on the flow time scale. Each such set, called a transmission mode, thus defines, which links are active and which transmission power is used on each active link. For each transmission mode there is a capacity vector which defines the capacity of each link when that particular mode is active. Assume that these vectors form the columns of rate matrix $\mathbf{R}$. When the modes are scheduled on a fast time scale, the resulting link capacities available for flows are defined by the convex combinations of the columns of $\mathbf{R}$. The capacity set in the space spanned by link capacities (referred to as the link space) is thus given by $\mathcal{C}_{\text{link}} = \{\mathbf{c} = \mathbf{R}\mathbf{t} : \mathbf{e}^{\mathrm{T}}\mathbf{t} \leq \mathbf{1}, \mathbf{t} \geq 0\}$. See [PVJ06] for a more extensive description of the model.

In principle, from the rate matrix $\mathbf{R}$ one can define the capacity set also as a collection of linear inequality constraints, but generally this approach is computationally infeasible. However, for any fixed link capacity proportions (i.e., a direction in the link space) one may compute the corresponding boundary point, which defines the maximum available link capacities with the predefined proportions. The boundary point to the direction of $\mathbf{b}$ is given by $\mathbf{c_b} = \mathbf{b}/\Upsilon(\mathbf{b})$, where the function $\Upsilon(\mathbf{b})$ is defined as the solution to the LP problem [PVJ06]

$$\begin{aligned} \Upsilon(\mathbf{b}) = \min_{\mathbf{q}} & \ \mathbf{e}^{\mathrm{T}}\mathbf{q}, \\ \text{s.t. } & \mathbf{R}\mathbf{q} \geq \mathbf{b}, \\ & \mathbf{q} \geq \mathbf{0}. \end{aligned} \tag{5.8}$$

Throughput Asymptotics

The throughput behavior of wireless networks at low loads can be straightforwardly characterized using the formulae (3.22). The link-flow incidence matrix is denoted $\mathbf{A}$, i.e. $A_{ji} = 1$ indicates the flow i uses link j, otherwise $A_{ji} = 0$. Given the traffic pattern $\mathbf{p}$, the boundary point in the flow space is

$$\hat{\boldsymbol{\rho}} = \frac{\mathbf{p}}{\Upsilon(\mathbf{A}\mathbf{p})} \tag{5.9}$$

and the BF recursion (with, as usual, $\Phi(\mathbf{0}) = 1$ and $\Phi(\mathbf{x}) = 0$ for all $\mathbf{x} \notin \mathbb{Z}_+^N$) is [PVJ06]

$$\Phi(\mathbf{x}) = \Upsilon(\mathbf{A}\tilde{\mathbf{\Phi}}(\mathbf{x})), \tag{5.10}$$

where $\tilde{\mathbf{\Phi}}(\mathbf{x})$ is a vector containing the balance function values $\Phi(\mathbf{x}-\mathbf{e}_k)$.

In order to apply the heavy traffic derivative formula (3.28), we need to identify the saturated constraints at $r = 1$. Consider first the constraints in the link space. At the boundary of the capacity set in the link space any saturated constraint is a hyperplane the equation of which is directly available via duality. To the link space direction $\mathbf{Ap}$ the vector $\mathbf{u}$ defining the constraining hyperplane $\mathbf{u}^{\mathrm{T}}\mathbf{c}_{\mathbf{Ap}} = \mathbf{1}$ is given by the dual of the scheduling problem, i.e., it is the vector that solves

$$\max_{\mathbf{u}} \mathbf{u}^{\mathrm{T}}\mathbf{Ap}, \tag{5.11}$$

$$\text{s.t. } \mathbf{R}^{\mathrm{T}}\mathbf{u} \leq \mathbf{e}, \tag{5.12}$$

$$\mathbf{u} \geq \mathbf{0}. \tag{5.13}$$

In the case that the solution is not unique, i.e. several constraints are saturated simultaneously, one enumerates the spanning vectors of the solution space. This can be done by solving the dual problem by the simplex method, storing the optimal basis and carrying out, e.g., a depth-first search using simplex iterations over all other solutions that do not change the value of the objective function.

After the link space constraints have been determined, they need to be translated to the flow space. Let $\mathbf{u}_l$ be the lth extremal solution of the dual problem. Now, $b_{li} = \mathbf{a}_i^{\mathrm{T}}\mathbf{u}_l$ in (3.28), where $\mathbf{a}_i$ stands for the ith column of $\mathbf{A}$. Thus, the heavy load derivative for classes k such that $\gamma_k(1) = 0$ is given by

$$\gamma_k'(1) = -\frac{1}{\sum_j \mathbf{a}_k^{\mathrm{T}}\mathbf{u}_j}. \tag{5.14}$$

Example

Consider a wireless mesh network shown in Figure 5.4 (left), with two access points (AP), two relays (R) and 18 traffic classes. As the interference model, we assume that no node can participate in more than one transmission at a time. In other words, a feasible transmission mode is a matching on the network graph. We assume that each link has unit capacity when active. The traffic pattern is $p_i = i$ (without normalization) for each class i. We study the throughput of class 9 (marked with thick line in the left figure) which has the load $\hat{\rho}_9 = 3/40$ at the capacity limit. Figure 5.4 (right) shows the asymptotic throughput behavior of class 9. The throughput between the extremes is sketched using a cubic polynomial fitted to the end points and to the first two derivatives at $r = 0$ neglecting the derivative at $r = 1$. As seen in the figure, the low load derivative provides reasonable results while the heavy load derivative does not fit the rest of the curve well, hence better results are obtained when the heavy load derivative is omitted from the fitting.

5.5 Multihop Networks with ALOHA

In Publication 8, we consider a synchronized slotted time random access network (such as slotted Aloha), where the nodes access the channel randomly and independently of each other in each time slot. We assume that

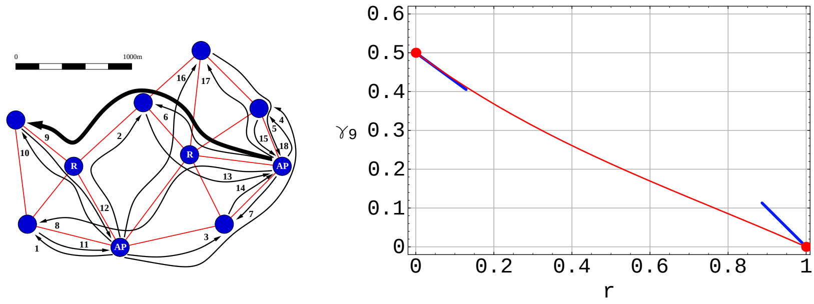


Figure 5.4: Left: Wireless mesh network and the flow classes. Right: Throughput asymptotics of the class denoted by the thick line (right).

the medium access time scale is much faster than the flow time scale and that the transmission probabilities can be adjusted whenever the state of the network changes. With this assumption, any given set of transmission probabilities results, from the flow level perspective, in a virtual network where the capacity of each link is the fraction of the nominal capacity that corresponds to the probability of successful transmission on that link in a randomly chosen slot.

The outgoing links of node i are identified by their receiver end nodes j belonging to the set L_i. We denote the transmission probability of link (i,j), $j \in L_i$, by p_{ij}. Note that the link transmissions in a single node are mutually exclusive and $\sum_{j\in L_i} p_{ij} = P_i$, where P_i is the overall transmission probability of node i. Denote the vector of all p_{ij} by $\mathbf{p}$ and the vector of the P_i by $\mathbf{P}$. Each link may be interfered by other transmitting nodes. We denote the set of nodes that (in case they are transmitting) interfere with link (i,j) by N_{ij} with the convention that N_{ij} includes the receiving node j itself.

Assuming unit nominal capacity, the capacity of the link (i,j) is the probability that there is a transmission on that link and no other node interferes with the reception,

$$c_{ij}(\mathbf{p}) = p_{ij} \prod_{k\in N_{ij}} (1 - P_k). \tag{5.15}$$

Thus the capacity set of the network can be written as

$$\mathcal{C} = \{\boldsymbol{\phi} \,:\, \mathbf{A}\boldsymbol{\phi} \le \mathbf{c}(\mathbf{p}),\, \mathbf{P} \le \mathbf{1}\},$$

where $\mathbf{A}$ is the routing matrix, $\mathbf{c}(\mathbf{p})$ is the vector of $c_{ij}(\mathbf{p})$, and as explained above $\mathbf{P}$ is related to $\mathbf{p}$ by $\sum_{j\in L_i} p_{ij} = P_i$.

First, we discuss capacity sets in a single resource case where all the links interfere with each other. Then this interference assumption is relaxed and capacity sets of multihop networks with spatial reuse are considered.

Single resource system

First we consider a simple example proposed already by Kleinrock [Kle76] but give it a slightly different treatment. Assume n nodes compete for a common channel. Node i attempts to use the channel independent of the others with the probability P_i. We wish to determine the P_i such that the throughput s_i of node i is $\sigma_i s$, where the σ_i are given parameters defining the traffic profile and the total throughput s is maximized. The throughput of node i is given by

$$s_i = P_i \prod_{j \neq i} (1 - P_j) = \frac{P_i}{1 - P_i} q, \tag{5.16}$$

where $q = \prod_i (1 - P_i)$. Now, by setting $s_i = \sigma_i s$, we get

$$1 - P_i = \frac{q}{q + \sigma_i s} \quad \Rightarrow \quad q = \prod_i (1 - P_i) = \prod_i \frac{q}{q + \sigma_i s}, \tag{5.17}$$

which defines s as a function of q. By derivation with respect to q and noting that $s' = 0$ at the maximum we obtain the relation

$$\sum_i \frac{1}{1 + \sigma_i a} = n - 1, \tag{5.18}$$

where we have denoted $a = s/q$. The parameter a can be solved (at least numerically) from (5.18) which immediately gives the optimal solution:

$$s = a \prod_i \frac{1}{1 + \sigma_i a} \quad \text{with} \quad P_i = \frac{\sigma_i a}{1 + \sigma_i a}. \tag{5.19}$$

Note that the boundary of the maximum obtainable link capacities satisfies

$$\sum_i P_i = \sum_i \frac{\sigma_i a}{1 + \sigma_i a} = \sum_i \left(1 - \frac{1}{1 + \sigma_i a} \right) = n - (n - 1) = 1, \tag{5.20}$$

which is the same optimality condition as obtained in [Kle76] by considering the determinant of the Jacobian matrix.

Example When $n = 2$ equation (5.18) reads

$$\frac{1}{1 + \sigma_1 a} + \frac{1}{1 + \sigma_2 a} = 1, \tag{5.21}$$

from which

$$a = \frac{1}{\sqrt{\sigma_1 \sigma_2}}. \tag{5.22}$$

Substitution in (5.19) gives

$$s = \frac{1}{(\sqrt{\sigma_1} + \sqrt{\sigma_2})^2}, \quad \text{with} \quad P_i = \frac{\sqrt{\sigma_i}}{\sqrt{\sigma_1} + \sqrt{\sigma_2}}, \tag{5.23}$$

and recalling that $s_i = \sigma_i s$ the equation for the boundary becomes

$$\sqrt{s_1} + \sqrt{s_2} = 1. \tag{5.24}$$

Multihop random access network

The previous example assumed that all links interfere with each other. With spatial reuse of the channel the determination of the maximal link capacities with given proportions becomes more challenging.

Denote a traffic profile by $\boldsymbol{\sigma}$, where σ_{ij} gives the relative capacity of link (i, j). Thus, $\boldsymbol{\sigma}$ is a vector defining a direction in the link capacity space. The maximum sustainable link capacities in this direction are $s\,\boldsymbol{\sigma}$, where s is the throughput coefficient to be solved.

Consider first a single node i. Locally the throughput coefficient, denoted by s_i, solves

$$\begin{cases} p_{ij} \prod_{k\in N_{ij}} (1 - P_k) = s_i \sigma_{ij}, \quad \forall j \in L_i, \\ \sum_{j\in L_i} p_{ij} = P_i, \end{cases} \tag{5.25}$$

which clearly fixes the link transmission probabilities and leaves only the node transmission probabilities as unknowns. The local throughput is then given by

$$s_i = P_i \left(\sum_{j\in L_i} \frac{\sigma_{ij}}{\prod_{k\in N_{ij}} (1 - P_k)} \right)^{-1}, \tag{5.26}$$

with the link transmission probabilities fixed by $\mathbf{P}$,

$$p_{ij} = \frac{\sigma_{ij}}{\prod_{k\in N_{ij}} (1 - P_k)} P_i \left(\sum_{j\in L_i} \frac{\sigma_{ij}}{\prod_{k\in N_{ij}} (1 - P_k)} \right)^{-1}. \tag{5.27}$$

In the global network-wide view, we attempt to increase all s_i as much as possible. Thus, parameter s becomes

$$s = \max_{\mathbf{P}} \min_i P_i \left(\sum_{j\in L_i} \frac{\sigma_{ij}}{\prod_{k\in N_{ij}} (1 - P_k)} \right)^{-1}. \tag{5.28}$$

We propose solving the problem (5.28) by the following water-filling argument (see Appendix A in Publication 8 for a more detailed description); we adjust $\mathbf{P}$ so that all s_i gain approximately equal increments. The actual algorithm consists of two alternating phases; (i) progressing to a direction where all the s_i initially grow at the same pace and (ii) equalization of the s_i to a common value in order to correct for the unequality due to finite step size. The idea of the algorithm is illustrated in Figure 5.5 which shows the contour of $\min_i s_i$ in a two-link scenario where the links interfere and $\sigma_1 = \frac{3}{4}$ and $\sigma_2 = \frac{1}{4}$. The progression of $\mathbf{P}$ in the algorithm is visualized using lines. The phases of the algorithm are clearly visible, every first step is a step towards increasing all s_i and every second step sets s_i-values equal (boundary $s_1 = s_2$ is shown as a dashed line in the figure).

Throughput analysis

Next, we analyze the flow throughput of various Aloha setting assuming balanced fairness capacity allocation. In order to determine the exact flow

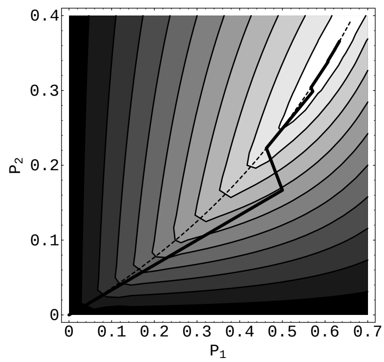


Figure 5.5: Illustration of the algorithm in 2-class Aloha system with $\sigma_1 = \frac{3}{4}$ and $\sigma_2 = \frac{1}{4}$

throughputs (2.11), the state sum (2.7) needs to be solved. First, we derive an exact result for the throughput in a single channel system with two traffic classes, which is a nice addition to the set of exactly solvable problems under BF. We also present an exact analysis of linear networks with only end-to-end traffic. In general, the throughput cannot be found in closed form, but one has to resort to numerical methods. We use asymptotic analysis to approximate the performance of general multihop random access networks.

Single resource system with two classes It is well known (see, e.g., [BP03a]) that when two classes compete for a common resource of unit capacity then under balanced fairness the system behaves as a single PS queue, each flow getting an equal share of the common resource. The balance function of the system is

$$\Phi_{\mathrm{PS}}(\mathbf{x}) = \binom{x_1 + x_2}{x_1,\, x_2} \tag{5.29}$$

and the corresponding allocations satisfy

$$\phi_1(\mathbf{x}) + \phi_2(\mathbf{x}) = 1. \tag{5.30}$$

The state sum is also well known,

$$G_1(\rho_1, \rho_2) = \sum_{x_1, x_2} \Phi_{\mathrm{PS}}(\mathbf{x}) \rho_1^{x_1} \rho_2^{x_2} = \frac{1}{1 - \rho_1 - \rho_2}. \tag{5.31}$$

From the balance function (5.29) one can derive new systems. In particular using $\Phi_{\mathrm{PS}}(\mathbf{x})^\alpha$ as a new balance function leads to the allocations

$$\phi_k^{(\alpha)}(\mathbf{x}) = \phi_k(\mathbf{x})^\alpha, \quad k = 1, 2 \tag{5.32}$$

satisfying

$$\phi_1^{(\alpha)}(\mathbf{x})^{1/\alpha} + \phi_2^{(\alpha)}(\mathbf{x})^{1/\alpha} = 1. \tag{5.33}$$

For $\alpha = 2$ the capacity set defined by (5.33) is precisely the one of a 2-class Aloha system, where two stations are accessing the same channel

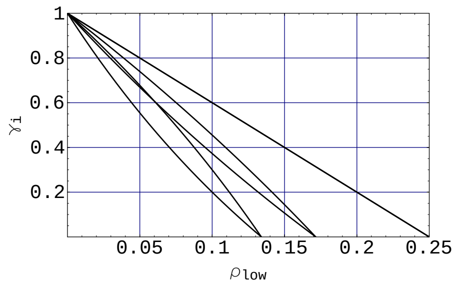


Figure 5.6: Throughput of the two classes with load ratios 3, 2, and 1, respectively, as a function of the load of the low-load class. For uneven loads, the upper curve corresponds to the class having higher load and the lower curve to the low-load class.

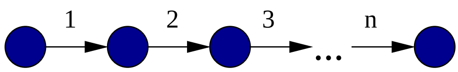


Figure 5.7: Linear network with n links.

(5.24). So the balance function of this system is $\Phi_{\text{PS}}(\mathbf{x})^2$. In order to calculate the throughputs of different classes we have to determine the state sum

$$G_2(\rho_1, \rho_2) = \sum_{x_1, x_2} \Phi_{\text{PS}}(\mathbf{x})^2 \rho_1^{x_1} \rho_2^{x_2} = \sum_{x_1, x_2} \binom{x_1 + x_2}{x_1,\ x_2}^2 \rho_1^{x_1} \rho_2^{x_2}. \tag{5.34}$$

It is shown in Publication 8 that the sum results in

$$G_2(\rho_1, \rho_2) = \frac{1}{\sqrt{1 - 2(\rho_1 + \rho_2) + (\rho_1 - \rho_2)^2}}. \tag{5.35}$$

The throughputs can now be calculated by (2.11)

$$\gamma_i = \frac{G_2(\rho_1, \rho_2)}{\frac{\partial}{\partial \rho_i} G_2(\rho_1, \rho_2)} = \frac{1 - 2(\rho_1 + \rho_2) + (\rho_1 - \rho_2)^2}{1 \mp (\rho_1 - \rho_2)}, \tag{5.36}$$

where the minus sign is for $i = 1$ and plus sign for $i = 2$.

The throughputs are shown in Figure 5.6 for load ratios 1, 2, and 3 as a function of the load of the low-load class. The diagonal line corresponds to the case $\rho_1 = \rho_2$.

Linear networks with end-to-end flows Consider a linear network with unidirectional links and only end-to-end traffic as show in Figure 5.7. As there is only one traffic class, BF capacity allocation reduces to processor sharing.

Assume that the interfering nodes of a link are the receiver node of that link and the subsequent node to the receiver. Then the probabilities of

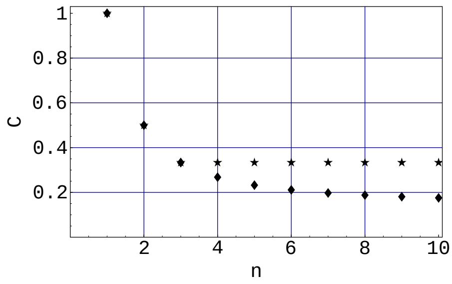


Figure 5.8: Capacity of a linear network with n links. Lower dots correspond to flow-optimized random access and upper to fully coordinated MAC.

successful transmissions of link i, s_i, are

$$\begin{cases} s_1 = p_1(1-p_2)(1-p_3) \\ \vdots \\ s_{n-2} = p_{n-2}(1-p_{n-1})(1-p_n) \\ s_{n-1} = p_{n-1}(1-p_n) \\ s_n = p_n. \end{cases} \tag{5.37}$$

It is obvious that in an optimal system all the success probabilities are equal. Thus, we get a set of equations

$$\begin{cases} p_1(1-p_2)(1-p_3) = p_2(1-p_3)(1-p_4) \\ \vdots \\ p_{n-4}(1-p_{n-3})(1-p_{n-2}) = p_{n-3}(1-p_{n-2})(1-p_{n-1}) \\ p_{n-3}(1-p_{n-2})(1-p_{n-1}) = p_{n-2}(1-p_{n-1}) \\ p_{n-2}(1-p_{n-1}) = p_{n-1}. \end{cases} \tag{5.38}$$

The maximal capacity C is obtained with $p_1 = 1$, hence the set of equations can be solved. For example, for $n = 2$ the solution to the equations gives $p_1 = 1$ and $p_2 = \frac{1}{2}$. For $n = 3$, $p_1 = 1$, $p_2 = \frac{1}{2}$ and $p_3 = \frac{1}{3}$. The resulting maximal capacities, $\frac{1}{2}$ and $\frac{1}{3}$, are exactly the same as in a fully coordinated MAC. If $n > 3$ the network capacity becomes slightly worse than that of coordinated MAC. As $n \to \infty$ the capacity approaches $\frac{4}{27} \approx 0.148$, with $p_i = \frac{1}{3}$ for all $i > 1$. In comparison, the capacity of a coordinated line network with the assumed interference model is $\frac{1}{3}$. In this scenario, BF capacity allocation corresponds to processor sharing, hence the mean flow throughput is $\gamma = C - \rho$ and the throughput curve is a line from point $(0, C)$ to $(C, 0)$. The maximal capacities C are illustrated in Figure 5.8 and compared to fully coordinated MAC. The longer the network, the greater the difference between the approaches.

Multihop networks Finally, we analyze the throughput in general multihop random access networks. For a given state $\mathbf{x}$, the balanced fairness

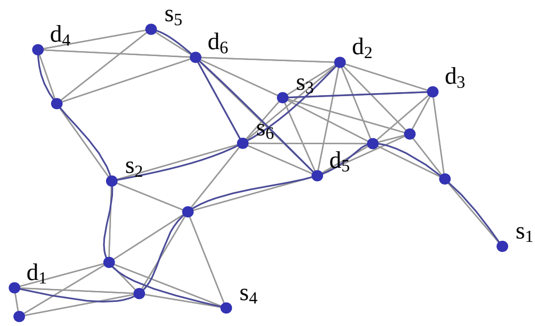


Figure 5.9: Wireless network with 20 nodes

recursion step (2.8) can be written as

$$\Phi(\mathbf{x}) = \frac{1}{s} = \min_{\mathbf{P}} \max_{i} P_i^{-1} \sum_{j \in L_i} \frac{\sigma_{ij}(\mathbf{x})}{\prod_{k \in N_{ij}} (1 - P_k)}, \tag{5.39}$$

where the traffic profile is given by

$$\boldsymbol{\sigma}(\mathbf{x}) = \mathbf{A}\tilde{\boldsymbol{\Phi}}(\mathbf{x}). \tag{5.40}$$

The balance function values can thus be solved using the optimization algorithm related to equation (5.28).

Though we were able to derive the throughputs in a two-class single-channel system in closed form, an exact analysis is not feasible in more complex networks and hence approximative methods are needed. Again, we use throughput asymptotics in the analysis.

We denote by $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_K)^{\mathrm{T}}$ the traffic profile of the network, where ρ_i is the proportion of class-i traffic and $\sum \rho_i = 1$. Load line is the line segment from zero to the boundary of the capacity set $\mathcal{C}$ in the direction of vector $\boldsymbol{\rho}$. The boundary point $\hat{\boldsymbol{\rho}} = s\,\boldsymbol{\rho}$, where s is solved from (5.28) with $\boldsymbol{\sigma} = \mathbf{A}\boldsymbol{\rho}$. The load line can then be parameterized as $r\hat{\boldsymbol{\rho}},\ r \in [0, 1]$.

Given both the end points and derivatives at zero load, the throughput curves can be sketched. In the single channel system with two classes, the throughput curve (5.36) is a rational function, hence it can be exactly extrapolated using an appropriate rational extrapolation function. In the general case the correct functional form is not known. In our numerical experiments, we use polynomial interpolation functions.

Numerical example In order to demonstrate the asymptotic analysis, we study the network illustrated in Figure 5.9. The network has 20 nodes and 98 links. A transmitting node is assumed to interfere all receiving nodes within the transmission distance. We assume 6 traffic classes and denote class-i source node s_i and destination node d_i. All the traffic is routed using the shortest paths as illustrated in the figure.

Assuming an equal load on all traffic classes, asymptotic analysis is used to approximate the flow-level performance. Figure 5.10 illustrates the throughputs of classes 1, 2 and 3. The extrapolation is done using third order polynomial functions and the first two derivatives at low load regime. The results are compared to the performance of optimal fully coordinated MAC discussed in Section 5.4.

Class-3 and class-2 routes consist of one and two hops, respectively, hence low load performance of random access is close to the coordinated operation. As the load increases, the coordination gives a greater benefit and random access results in worse throughputs. Class 1 has a longer route consisting of six hops. In this case, the performance difference is significant also with low loads.

5.6 Summary

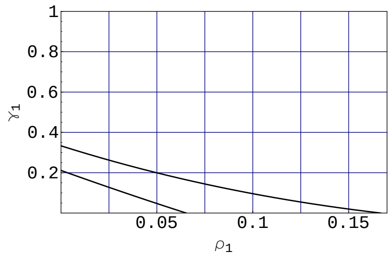
In this chapter, we have studied three different wireless data network models on the flow level. New transmissions arrive and depart dynamically and we analyzed the mean performance of the system. First, a cellular network with two base stations was modeled and analyzed. We also discussed multihop networks using either TDMA or ALOHA medium access methods.

When the cellular scenario was considered, we assumed that the base stations were coordinated and that a simple link rate adaptation scheme was used. We studied the effect of the operating policy on the performance of the system using both value extrapolation and balanced fairness. The effects of link adaptation and transmission coordination are apparent, even in this simplified model. There is a significant gap between the flow-level performance resulting from dynamic policies and that from static policies. Irrespective of operational policies, the flow-level performance analysis of such systems remains a difficult task. With BF one can go through the state space recursively and no matrix inversions are needed. Furthermore, the results are insensitive to the traffic details, thus making the results more robust. With value extrapolation, the performance of arbitrary policies can be approximated with heavily truncated state spaces. BF is computationally the lighter of the two approaches and is a reasonable approximation of the other capacity allocation policies studied.

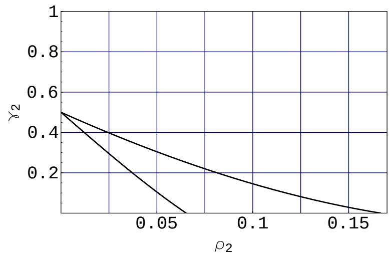
We also analyzed flow-level performance in multihop networks. The resources are shared according to balanced fairness, which serves as an approximation of common fairness policies such as proportional fairness. On the basis of the earlier results in [PVJ06], we used asymptotic analysis to analyze the flow-level performance of multihop STDMA networks.

We also analyzed flow-optimized random access in the same setting. In particular, we derived the flow throughput analytically in the two-class case, where the classes share a single resource. For arbitrary flow-optimized random access networks we provided a general scheme that enables the throughput to be evaluated. The scheme also entails a novel algorithm to determine the maximum link capacities with given proportions.

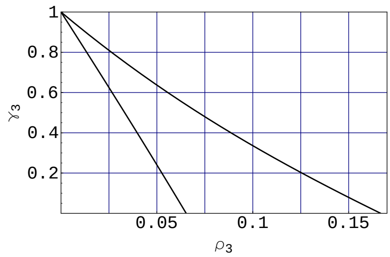
We compared the throughput behavior of flow-optimized random access against the throughput obtained by optimal STDMA scheduling. When the network is lightly loaded, the performance of the flow-optimized random access is comparable even to access-level optimal scheduling, but the performance deteriorates when the capacity limit is approached.



(a) Class 1



(b) Class 2



(c) Class 3

Figure 5.10: Asymptotic analysis of traffic classes 1, 2 and 3. The upper curves correspond to coordinated MAC and the lower curves to Aloha.

6 SUMMARY

In this thesis, we studied the performance analysis of data networks. Most telecommunication systems are dynamic in nature. The state of the network changes constantly as new transfers start and finish. In order to capture the behavior of such systems and to realistically evaluate their performance, it is essential to use dynamic models in the analysis. We studied networks carrying elastic traffic, i.e. file transfers or flows which are characterized by the file size and whose performance can be measured with the transmission duration which depends on the available capacity. Packet-level phenomena, e.g. packet delay or jitter, can be ignored when the flow-level performance of such networks is being analyzed. We assumed that medium access control and the capacity allocation mechanism operate on a faster time scale than flow level and that transient phases after flow arrivals and departures are short, and hence the capacities allocated for the flows can be considered constant between flow arrivals and departures.

The assumptions allow us to model the systems at flow level. Networks are modeled as stochastic queueing systems in which the flows correspond to customers and the network acts as a server and allocates a share of capacity to each concurrent flow. Many interesting performance measures can be derived from the equilibrium state distribution of the system. In general, queueing systems are sensitive, i.e., the equilibrium state distribution depends on detailed system characteristics such as the arrival process and flow size distribution. In order to facilitate the analysis, restricting assumptions are usually needed. For example, Poissonian flow arrivals and exponential flow size distributions make a system more tractable. The equilibrium state distribution of a network can be approximated by truncating the state space and by solving the state probabilities from the global balance equations. Such analysis is computationally heavy and becomes intractable when the size or load of the network is increased, and hence more efficient approximative methods are needed to study more realistic settings.

In this thesis, we used two performance approximation methods: value extrapolation and balanced fairness, specifically asymptotic throughput analysis. In addition, we introduced an approximation method based on balanced fairness and the Monte Carlo method for evaluating large sums that can be used to estimate the performance of systems with low or medium loads. Instead of the state space being recursively gone through in order, it is sampled randomly and the average throughput is calculated.

Value extrapolation is a novel approximative method developed during this work and based on the theory of MDPs. It can be used to approximate performance metrics that can be formulated as the expected value of a function of the system state. In this thesis, we used it to approximate the mean number of flows (which is proportional to the mean transfer time) in data networks carrying elastic traffic. Instead of first solving the state probabilities using the global balance equations, the performance measure is determined directly using relative values of the states and the Howard equations. The idea of value extrapolation is that the relative values outside the truncated state space can often be accurately extrapolated using a polynomial function, allowing more accurate results to be obtained without any significant computational penalty. As our numerical studies showed, value extrapolation works well when queueing networks are studied, allowing the state space to be heavily truncated while still leading to accurate results.

Balanced fairness is a recent capacity allocation scheme introduced by Bonald and Proutière. BF is the most efficient insensitive capacity allocation policy, i.e., the equilibrium distribution depends only on the traffic amounts as long as session arrivals are Poissonian. Balanced fairness simplifies performance analysis as the equilibrium distribution can be determined recursively state by state. The performance of balanced fairness is typically relatively similar to other fair capacity allocation policies, e.g. proportional or max-min fairness, and can be used to approximate these other schemes. In [BPV06], Bonald et al. introduced asymptotic throughput analysis, a method for approximating the mean flow throughputs in a network using balanced fairness. The mean throughput of a class can be approximated as a function of the system load. The mean throughput and its derivatives can be determined at zero load, allowing throughput to be extrapolated as the load of the system increases. If a traffic class is saturated at the capacity limit, the end point of the curve is also known, thus making more accurate interpolation possible. In some cases, the derivative of the throughput at the capacity limit can also be determined. In the numerical experimentations conducted in this thesis, throughput asymptotics worked well in various network settings. The results seemed reasonable and matched well with the results obtained using other methods. As already discussed in [BPV06], the heavy load derivative fits the throughput curve only at loads very close to the capacity limit. Our numerical results further confirmed this, because the derivative often differed significantly from the approximated curves.

The performance analysis methods are applied in two settings: load balancing and wireless networks. The purpose of load balancing is to divide the traffic load efficiently among the network resources in order to improve the performance. We discussed insensitive load balancing in data networks and load balancing among parallel discriminatory processor sharing queues.

On the basis of the insensitivity results of Bonald and Proutière, we studied both packet- and flow-level balancing in fixed data networks. When packet-level balancing is considered, a flow can be divided among multiple routes. We formulated the balanced fairness recursion step as a linear programming problem. When flow-level balancing is studied, an arriving flow is assigned to a route, which is used until the flow is finished. In contrast to other applications of balanced fairness in this thesis, flow-level balancing problems cannot be solved recursively state by state, but an LP problem corresponding to the whole state space needs to be solved, making the approach feasible only for small toy networks. Our numerical examples illustrated that, as expected, packet-level balancing outperforms flow-level balancing and that flow-level balancing performs better if the routing is balanced jointly with the capacity allocation.

Next, we studied load balancing among multiple parallel discriminatory processor sharing queues. Such a system can be used to model, e.g., telecommunication or computer systems. We used value extrapolation to compare different load-balancing policies. We also showed that value extrapolation can be used to obtain the exact relative values related to the queue lengths of a DPS queue with Poissonian arrivals and Cox distributed service requirements. This result can be used to derive so-called first iteration policies by taking only the first iteration round of policy iteration without solving the Howard equations, and hence we were able to derive policies outperforming the heuristic policies without significant computations.

Finally, we analyzed the performance of wireless networks carrying elastic data traffic in two settings: a simple cellular setting with road topology and multihop networks. Wireless networks are gaining more and more popularity as their advantages, such as easy deployment and mobility, outweigh their downsides.

The first wireless setting studied was a simple cellular network with two base stations and link adaptation, where users were located along a line between the base stations. We compared different capacity allocation policies using value extrapolation. While value extrapolation facilitates the analysis of arbitrary policies, balanced fairness is computationally lighter and approximated the performance of the other policies well. The effects of link adaptation and transmission coordination were apparent even in this simplified model. Irrespective of operational policies, the flow-level performance of the example system remains a difficult task.

In contrast to most previous analytical work related to wireless multihop networks using static traffic models, we studied networks in a dynamic setting where new flows appear and depart. We analyzed two different MAC schemes. Penttinen et al. modeled multihop STDMA networks in [PVJ06] and formulated the balanced fairness recursion step as an LP problem. On the basis of their work, we conducted asymptotic throughput analysis. The other scheme studied was multihop networks with random access. We assumed that the transmission probabilities can be optimized between flow arrivals and departures. We compared the throughput behavior of flow-optimized random access against the throughput obtained by optimal scheduling assuming balanced fairness capacity allocation. In particular, we derived the flow throughput analytically in the two-class case, where the classes share a single resource. For arbitrary flow-optimized random access networks we provided a general scheme that enables the throughput to be evaluated. The scheme also entails a novel algorithm to determine the maximum link capacities with given proportions. When the network is lightly loaded, the performance of the flow-optimized random access was comparable even to access-level optimal scheduling, but the performance deteriorated when the capacity limit was approached.

7 AUTHOR'S CONTRIBUTION

Publication 1 This paper is a joint work of the authors. The idea of value extrapolation is by prof. Jorma Virtamo but it was generalized by the present author. The numerical studies were conducted and the paper was written by the present author.

Publication 2 This paper is a joint work of the authors. Section 3 is by the present author and sections 4 and 5 by Dr. Aleksi Penttinen.

Publication 3 This paper is an independent work of the author.

Publication 4 This paper is a joint work of the authors. The paper was written by the present author.

Publication 5 This paper is a joint work of the authors. The numerical studies were conducted and the paper was written by the present author.

Publication 6 This paper is a joint work of the authors. The numerical studies were conducted and the paper was written by the present author.

Publication 7 This paper is a joint work of the authors. Numerical studies in Section IV B were conducted by the present author.

Publication 8 This paper is a joint work by the authors except the appendices. The algorithm in appendix A is by Prof. Jorma Virtamo and Dr. Aleksi Penttinen and the proof in appendix B by Prof. Jorma Virtamo.

REFERENCES

[AAA06] E. Altman, K. Avrachenkov, and U. Ayesta. A survey on discriminatory processor sharing. *Queueing Systems*, 53(1–2):53–63, 2006.

[Abr70] N. Abramson. The Aloha system - another alternative for computer communications. In *Proceedings of Fall Joint Computer Conference, AFIPS Conference*, page 37, 1970.

[AJK04] E. Altman, T. Jimenez, and D. Kofman. DPS queues with stationary ergodic service times and the performance of TCP in overload. In *Proceedings of INFOCOM 2004*, pages 975–983, 2004.

[Alt02a] E. Altman. Applications of Markov decision processes in communication networks: A survey. In *Handbook of Markov Decision Processes*, pages 489–536. Kluwer, 2002.

[Alt02b] E. Altman. Capacity of multi-service cellular networks with transmission-rate control: A queueing analysis. In *Proceedings of ACM Mobicom*, pages 205–214, 2002.

[AMO93] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms and Applications*. Prentice Hall, 1993.

[ASSC02] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, 2002.

[BaM06] F. Baccelli, B. Błaszczyszyn, and P. Mühlethaler. An aloha protocol for multihop mobile wireless networks. *IEEE Transactions on Information Theory*, 52(2):421–436, 2006.

[BCG05] R. Bruno, M. Conti, and E. Gregori. Mesh networks: commodity multihop ad hoc networks. *IEEE Communications Magazine*, 43(3):123–131, 2005.

[BCMP75] F. Baskett, K. Mani Chandy, R. Muntz, and F. Palacios. Open, closed, and mixed networks of queues with different classes of customers. *Journal of the ACM*, 22(2):248–260, 1975.

[BG92] D. Bertsekas and R. Gallager. *Data networks, 2nd edition*. Prentice-Hall, 1992.

[BJP04] T. Bonald, M. Jonckheere, and A. Proutière. Insensitive load balancing. In *Proceedings of Sigmetrics/Performance 2004*, pages 367–377, 2004.

[BMPV06] T. Bonald, L. Massoulié, A. Proutière, and J. Virtamo. A queueing analysis of max-min fairness, proportional fairness and balanced fairness. *Queueing Systems: Theory and Applications*, 53(1–2):65–84, 2006.

[Bon06a] T. Bonald. Insensitive queueing models for communication networks. In *Proceedings of ValueTools'06*, 2006.

[Bon06b] T. Bonald. Throughput performance in networks with linear constraints. In *Proceedings of 40th Annual Conference on Information Sciences and Systems (CISS)*, Mar. 2006.

[Bor05] S. Borst. User-level performance of channel-aware scheduling algorithms in wireless data networks. *IEEE/ACM Transactions on Networking*, 13(3):636–647, 2005.

[BP02] T. Bonald and A. Proutière. Insensitivity in processor-sharing networks. *Performance Evaluation*, 49:193–209, 2002.

[BP03a] T. Bonald and A. Proutière. Insensitive bandwidth sharing in data networks. *Queuing Systems*, 44(1):69–100, 2003.

[BP03b] T. Bonald and A. Proutière. Wireless downlink data channels: user performance and cell dimensioning. In *MobiCom '03*, pages 339–352, 2003.

[BP04] T. Bonald and A. Proutière. On performance bounds for balanced fairness. *Performance Evaluation*, 55:25–50, 2004.

[BPC+07] P. Baronti, P. Pillai, V. W. C. Chook, S. Chessa, A. Gotta, and Y. Fun Hu. Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards. *Computer Communications*, 30(7):1655–1695, 2007.

[BPRV03] T. Bonald, A. Proutiére, J. Roberts, and J. Virtamo. Computational aspects of balanced fairness. In *Proceedings of the 18th International Teletraffic Congress (ITC-18)*, 2003.

[BPV06] T. Bonald, A. Penttinen, and J. Virtamo. On light and heavy traffic approximations of balanced fairness. In *Proceedings of SIGMETRICS/Performance*, pages 109–120, 2006.

[BV04] T. Bonald and J. Virtamo. Calculating the flow level performance of balanced fairness in tree networks. *Performance Evaluation*, 58(1):1–14, 2004.

[CH98] A. K. Choudhury and E. L. Hahne. Dynamic queue length thresholds for shared-memory packet switches. *Networking, IEEE/ACM Transactions on*, 6(2):130–140, 1998.

[DKS89] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. In *Proceedings of SIGCOMM '89*, pages 1–12, 1989.

[Erl09] A. Erlang. The theory of probabilities and telephone conversations. *Nyt Tidsskrift for Matematik B*, 20, 1909.

[Erl17] A. Erlang. Solution of some problems in the theory of probabilities of significance in automatic telephone exchanges. *Elektroteknikeren*, 13, 1917.

[es05] IEEE 802.16e standard. Physical and medium access control layers for combined fixed and mobile operation in licensed bands, 2005.

[Eva96] S. Evans. Analysing system behaviour on different time scales. In *Stochastic Networks: Theory and Applications*, pages 231–246. Oxford University Press, 1996.

[EVW80] A. Ephremides, P. Varaiya, and J. Walrand. A simple dynamic routing problem. *IEEE Transactions on Automatic Control*, 25:690–693, 1980.

[FMI80] G. Fayolle, I. Mitrani, and R. Iasnogorodski. Sharing a processor among many job classes. *Journal of the ACM*, 27(3):519–532, 1980.

[FP01] S. Floyd and V. Paxson. Difficulties in simulating the internet. *IEEE/ACM Transaction on Networking*, 9(4):392–403, 2001.

[GK00] P. Gupta and P. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, 2000.

[HCMSS04] F. Hernandez-Campos, J. S. Marron, G. Samorodnitsky, and F. D. Smith. Variable heavy tails in internet traffic. *Performance Evaluation*, 58(2–3):261–284, 2004.

[HLN97] D. P. Heyman, T. V. Lakshman, and Arnold L. Neidhardt. A new method for analysing feedback-based protocols with applications to engineering web traffic over the internet. In *Proceedings of the 1997 ACM SIGMETRICS*, pages 24–38, 1997.

[Jai82] N. K. Jaiswal. Performance evaluation studies for time-sharing computer systems. *Performance Evaluation*, 2(4):223–236, 1982.

[Jon06] M. Jonckheere. Insensitive versus efficient dynamic load balancing in networks without blocking. *Queueing Systems: Theory and Applications*, 54(3):193–202, 2006.

[JPPQ05] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu. Impact of interference on multi-hop wireless network performance. *Wireless Networks*, 11(4):471–487, 2005.

[JV05] M. Jonckheere and J. Virtamo. Optimal insensitive routing and bandwidth sharing in simple data networks. In *Proceedings of ACM SIGMETRICS*, pages 193–204, Banff, Canada, 2005.

[Kel76] F. Kelly. Networks of queues. *Advances in Applied Probability*, 8:416–432, 1976.

[Ken53] D. G. Kendall. Stochastic processes occurring in the theory of queues and their analysis by the method of the imbedded Markov chain. *Annals of Mathematical Statistics*, 24(3):338–354, 1953.

[KK04] J. Kim and B. Kim. Sojourn time distribution in the $M/M/1$ queue with discriminatory processor-sharing. *Performance Evaluation*, 58(4):341–365, 2004.

[Kle64] L. Kleinrock. *Communication nets; stochastic message flow and delay*. McGraw-Hill, 1964.

[Kle67] L. Kleinrock. Time-shared systems: A theoretical treatment. *Journal of the ACM*, 14(2):242–261, 1967.

[Kle76] L. Kleinrock. *Queueing systems, vol II: Computer applications*. John Wiley & Sons, 1976.

[KM07] W. Kiess and M. Mauve. A survey on real-world implementations of mobile ad-hoc networks. *Ad Hoc Networks*, 5(3):324–339, 2007.

[KMT98] F. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49:237–252, 1998.

[Koo98] G. M. Koole. The deviation matrix of the M/M/1 and M/M/1/N queue, with applications to controlled queueing models. In *Proceedings of the 37th IEEE CDC*, pages 56–59, Tampa, 1998.

[KRD06] S. Kumar, V. S. Raghavan, and J. Deng. Medium access control protocols for ad hoc wireless networks: A survey. *Ad Hoc Networks*, 4(3):326–358, 2006.

[KS78] L. Kleinrock and J. Silvester. Optimum transmission radii for packet radio networks or why six is a magic number. In *Proceeding of IEEE National Telecommunications Conference*, 1978.

[LCS03] X. Liu, E. K. P. Chong, and N. B. Shroff. A framework for opportunistic scheduling in wireless networks. *Computer Networks*, 41(4):451–474, 2003.

[Moy98] J. Moy. OSPF version 2. IETF RFC 2328, 1998.

[MR99] L. Massoulié and J. W. Roberts. Arguments in favour of admission control for TCP flows. In *Proceedings of ITC 16*, pages 33–44, 1999.

[MR00] L. Massoulié and J. Roberts. Bandwidth sharing and admission control for elastic traffic. *Telecommunication Systems*, 15(1):185–201, 2000.

[NK85] R. Nelson and L. Kleinrock. Spatial TDMA: A collision-free multihop channel access protocol. *IEEE Transactions on Communications*, 33(9):934–944, September 1985.

[NMM98] M. Nabe, M. Murata, and H. Miyahara. Analysis and modeling of world wide web traffic for capacity dimensioning of internet access lines. *Performance Evaluation*, 34(4):249–271, 1998.

[ORS93] A. Orda, R. Rom, and N. Shinkin. Competitive routing in multiuser communication networks. *IEEE/ACM Transactions on Networking*, 1:614–627, 1993.

[PM04] M. Pióro and D. Medhi. *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[PV02] R. Prakash and V. V. Veeravalli. A time-scale separation technique for the analysis of random access systems with incremental redundancy. In *Proceedings of IEEE International Symposium on Information Theory*, page 15, 2002.

[PVJ06] A. Penttinen, J. Virtamo, and R. Jäntti. Performance analysis in multi-hop radio networks with balanced fair resource sharing. *Telecommunications Systems*, 31(4):315–336, 2006.

[rgppG] 3rd generation partnership project (3GPP). http://www.3gpp.org.

[Rob00] J. Roberts. Engineering for quality of service. In K. Park and W. Willinger, editors, *Self-Similar Network Traffic and Performance Evaluation*. Wiley, 2000.

[Rob04] J. W. Roberts. A survey on statistical bandwidth sharing. *Computer Networks*, 45:319–332, 2004.

[RS96] K. M. Rege and B. Sengupta. Queue-length distribution for the discriminatory processor-sharing queue. *Operations Research*, 58(4):653–657, 1996.

[RTF+01] E. Rosen, D. Tappan, G. Fedorkow, Y. Rekhter, D. Farinacci, T. Li, and A. Conta. MPLS label stack encoding. IETF RFC 3032, 2001.

[RVC01] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol label switching architecture. IETF RFC 3031, 2001.

[Ser99] R. Serfozo. *Introduction to Stochastic Networks*. Springer-Verlag, New York, 1999.

[Sha49] C. E. Shannon. Communicating in the presence of noise. In *Proceedings of the Institute of Radio Engineers (IRE)*, volume 37, pages 10–21, 1949.

[SK83] J. Silvester and L. Kleinrock. On the capacity of multihop slotted ALOHA networks with regular structure. *IEEE Transactions on Communications*, 31(8):974–982, 1983.

[Ski97] S. Skiena. *The Algorithm Design Manual*. Springer-Verlag, 1997.

[sta99] IEEE 802.11 standard. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, 1999.

[sta01] IEEE 802 standard. IEEE standard for local and metropolitan area networks: Overview and architecture, 2001.

[Tij94] H. Tijms. *Stochastic Models: An Algorithmic Approach*. John Wiley & Sons, Chichester, England, 1994.

[Tim03] V. Timonen. Simulation studies on performance of balanced fairness. Master's thesis, Helsinki University of Technology, 2003.

[TPSC92] D. Towsley, D. Panayotis, P. Sparaggis, and C. Cassandras. Optimal routing and buffer allocation for a class of finite capacity queueing systems. *IEEE Transactions on Automatic Control*, 37:1446–1451, 1992.

[vKNB05] G. van Kessel, R. Núñez-Queija, and S. Borst. Differentiated bandwidth sharing with disparate flow sizes. In *Proceeding of IEEE INFOCOM 2005*, pages 2425–2435, 2005.

[vU03] M. van Uitert. *Generalized Processor Sharing Queues*. PhD thesis, Eindhoven University of Technology, 2003.

[Whi86] W. Whitt. Deciding which queue to join: some counterexamples. *Operations Research*, 34:55–62, 1986.

[WK06] X. Wang and K. Kar. Cross-layer rate optimization for proportional fairness in multihop wireless networks with random access. *IEEE Journal on Selected Areas in Communications*, 24(8):1548–1559, 2006.

[YY07] S. F. Yashkov and A. S. Yashkova. Processor sharing: A survey of the mathematical theory. *Automation and Remote Control*, 68(9):1662–1731, 2007.