Helsinki University of Technology Networking Laboratory
Teknillinen korkeakoulu Tietoverkkolaboratorio
Espoo 2006 Report 5/2006

ALGORITHMS AND PERFORMANCE EVALUATION METHODS FOR WIRELESS NETWORKS

Aleksi Penttinen

Dissertation for the degree of Doctor of Science in Technology to be presented with due permission for public examination and debate in Auditorium S4 at Helsinki University of Technology (Espoo, Finland) on the 29th of September, 2006, at 12 o'clock noon.

Helsinki University of Technology
Department of Electrical and Communications Engineering
Networking Laboratory

Teknillinen korkeakoulu
Sähkö- ja tietoliikennetekniikan osasto
Tietoverkkolaboratorio

Distribution:
Helsinki University of Technology
Networking Laboratory
P.O.Box 3000
FIN-02015 TKK
Tel. +358-9-451 2461
Fax. +358-9-451 2474

ISBN 951-22-8329-8
ISSN 1458-0322

Picaset Oy
Espoo 2006

HELSINKI UNIVERSITY OF TECHNOLOGY
Networking Laboratory

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

Name of the dissertation
Algorithms and Performance Evaluation Methods for Wireless Networks

Date of manuscript 28th of August 2006	Date of the dissertation 29th of September
☐ Monograph	☒ Article dissertation (summary + original articles)

Department	Department of Electrical and Communications Engineering
Laboratory	Networking Laboratory
Field of Research	Teletraffic Theory
Opponent	Professor Sem Borst (Eindhoven University of Technology, the Netherlands)
Supervisor	Professor Jorma Virtamo (Helsinki University of Technology)

Abstract

The performance of wireless networks depends fundamentally on the characteristics of the radio resource. In this thesis we study methods that can be used to improve performance of wireless networks. We also study methods that can be used to analyze the performance of such networks.

In the first part of the thesis, we propose algorithms for multicast routing and max-min fair link scheduling in wireless multihop networks. The multicast routing problem is to find a minimum-cost sequence of transmissions which delivers a message from a given source node to a set of destination nodes. We propose three efficient multicast routing algorithms for certain common instances of the problem. The first algorithm assumes fixed tranmission costs and constructs an efficient multicast tree in a centralized fashion. The second algorithm can be used to minimize only the number of transmissions in the multicast tree, but it has a distributed implementation. The last algorithm is applicable in scenarios where the network nodes can control their transmission range and the objective is to minimize the power consumption of the multicast tree. In the max-min fair link scheduling problem one attempts to assign transmission rights to flows in a wireless multihop network so that the long-term flow rates become max-min fair. We present a low-complexity, low-overhead distributed algorithm for the problem.

The second part comprises of the flow-level performance analysis of elastic data traffic in wireless networks. The network is modeled in a dynamic setting, in which flows (e.g., file transfers) arrive stochastically and depart upon completion. We discuss how a recently introduced resource allocation concept, balanced fairness, can be applied to wireless networks and devise an efficient computational scheme for solving the resulting joint problem of scheduling and resource allocation. Additionally, we propose an alternative method to approximate the flow throughput under balanced fairness in arbitrary networks. Finally, we adapt balanced fairness to a model where flows are indexed by a continuous variable. The model can capture, e.g., location-dependent features of flows.

Keywords	Wireless communication networks, ad hoc networks, multicast routing, link scheduling, elastic traffic, performance, balanced fairness		
ISBN (printed)	951-22-8329-8	ISSN (printed)	1458-0322
ISBN (pdf)	951-22-8331-X	ISSN (pdf)	
ISBN (others)		Number of pages	74 p. + app. 80 p.
Publisher	Networking Laboratory / Helsinki University of Technology		
Print distribution			

☒ The dissertation can be read at http://lib.tkk.fi/Diss/

HELSINKI UNIVERSITY OF TECHNOLOGY
Networking Laboratory

TEKNILLINEN KORKEAKOULU P.O.BOX 1000, FIN-02015 TKK http://www.tkk.fi/	VÄITÖSKIRJAN TIIVISTELMÄ
Tekijä: Aleksi Penttinen	

Väitöskirjan nimi
Algoritmeja ja suorituskykyarvioinnin menetelmiä langattomille verkoille

Käsikirjoitus jätetty 28. elokuuta 2006	Väitöstilaisuus 29. syyskuuta 2006
☐ Monografia	☒ Yhdistelmäväitöskirja

Osasto	Sähkö- ja tietoliikennetekniikan osasto
Laboratorio	Tietoverkkolaboratorio
Pääaine	Teleliikenneteoria
Vastaväittäjä	Professori Sem Borst (Eindhovenin teknillinen korkeakoulu, Alankomaat)
Valvoja	Professori Jorma Virtamo (Teknillinen Korkeakoulu)

Tiivistelmä

Langattomien verkkojen suorituskyky riippuu olennaisesti radiokanavan ominaisuuksista. Tässä väitöskirjassa tutkitaan menetelmiä, joilla verkkojen suorituskykyä pystytään parantamaan. Toisaalta tutkitaan myös menetelmiä, jotka mahdollistavat verkkojen suorituskyvyn analysoimisen.

Väitöskirjan ensimmäisessä osassa esitetään algoritmeja ryhmälähetysten reititykseen ja max-min-reiluun linkkien vuoronjakoon langattomissa monihyppyisissä verkoissa. Ryhmälähetysten reititysongelmassa etsitään joukkoa perättäisiä lähetyksiä, jotka välittävät viestin lähettäjältä joukolle vastaanottajia siten, että lähetyskustannusten summa minimoituu. Esitämme kolme tehokasta reititysalgoritmia tiettyihin ryhmälähetysongelman erikoistapauksiin. Ensimmäisessä algoritmissa oletamme lähetyskustannusten olevan mielivaltaiset, joskin kiinnitetyt, ja pyrimme keskitetysti minimoimaan lähetyspuun kokonaiskustannusta. Toinen algoritmi soveltuu pääasiassa lähetysten lukumäärän minimoimiseen, mutta se voidaan toteuttaa hajautetusti. Kolmas algoritmi on sovellettavissa tilanteissa, joissa verkon solmut voivat säätää lähetyssädettään ja tavoitteena on minimoida lähetyspuun tehonkulutus. Max-min-reilussa linkkien vuoronjako-ongelmassa pyritään jakamaan lähetysvuoroja verkossa oleville voille siten, että voiden pitkän aikavälin siirtonopeudet toteuttavat kyseisen reiluuskriteerin. Esitämme vähän kontrolliliikennettä vaativan, hajautetun algoritmin ongelman ratkaisemiseksi.

Väitöskirjan toinen osa koostuu ns. elastisen dataliikenteen vuotason suorituskykyanalyysistä langattomissa verkoissa. Verkko on mallinnettu dynaamisessa tilassa, missä voita (esim. tiedostonsiirtoja) saapuu verkkoon satunnaisen prosessin mukaisesti ja ne poistuvat verkosta siirron päättyessä. Käsittelemme kuinka vastikään kehitetty resurssienjakomenetelmä, tasapainotettu reiluus (balanced fairness), voidaan mukauttaa langattomiin verkkoihin ja kehitämme tehokkaan laskennallisen menetelmän mukautuksen vaatimaan lähetysvuoronjaon ja resurssienjaon yhteisoptimointiin. Tämän lisäksi esitämme vaihtoehtoisen tavan approksimoida voiden läpäisyä mielivaltaisissa verkossa tasapainotetun reiluuden mukaisen resurssienjaon voimassaollessa. Lopuksi sovellamme tasapainotetun reiluuden konseptia mallissa, jossa vuot on indeksoitu jatkuvalla muuttujalla. Tämä malli mahdollistaa esimerkiksi voiden paikkariippuvien ominaisuuksien mallintamisen.

Avainsanat	langattomat tietoverkot, ad hoc -verkot, ryhmälähetysten reititys, lähetysvuoronjako, elastinen liikenne, suorituskyky, tasapainotettu reiluus (balanced fairness)		
ISBN (printed)	951-22-8329-8	ISSN (printed)	1458-0322
ISBN (pdf)	951-22-8331-X	ISSN (pdf)	
ISBN (others)		Sivumäärä	74 s. + liit. 80 s.
Julkaisija	Tietoverkkolaboratorio / Teknillinen Korkeakoulu		
Print distribution			

☒ Tämä väitöskirja on saatavissa elektronisesti osoitteesta http://lib.tkk.fi/Diss/

PREFACE

This thesis is a result of the work which was started in the AHRAS project in 2001. I had just finished my M.Sc. thesis under the supervision of Prof. Jorma Virtamo when he asked whether I would be interested in Ph.D. studies on ad hoc networks in this project, which had just recently started. As I already had good experiences on working in the teletraffic theory group, I felt comfortable enough to take on the challenge. In retrospect, it was a good decision.

The work has been funded by the Graduate School of Electronics, Telecommunications and Automation (GETA) of the Academy of Finland and Finnish Defence Forces Technical Research Center. I would also like to acknowledge the generous financial support of the Nokia Foundation and the TES Foundation. Funding from the Euro-NGI Network of Excellence made possible my three-month visit to the University of Thessaly in the fall of 2004.

First of all, I wish to thank Prof. Jorma Virtamo for his considerable efforts as the supervisor and the instructor of this thesis. His active role in the research work has been invaluable and also extremely inspiring. I would also like to thank my co-authors Dr. Iordanis Koutsopoulos, Prof. Leandros Tassiulas, Dr. Riku Jäntti, M.Sc. Juha Leino and Dr. Thomas Bonald, for the fruitful co-operation we have had.

Again, the people at the Networking Laboratory have been of significant help in both the practical and theoretical problems I have encountered. They have also managed to create an enjoyable working atmosphere, both at the office and on the floorball court. For the pleasant time I spent in warm and hospitable Greece I wish to express my gratitude to Professor Leandros Tassiulas and his team.

I would like to thank my friends and parents for their company and support during the work. Finally, I want to thank my fiancée Heidi for her love, support, and understanding.

CONTENTS

Preface 1

Contents 3

List of Publications 5

1 Introduction 7
- 1.1 Wireless networks . . . 7
- 1.2 A networking perspective . . . 8
- 1.3 Algorithm development for wireless multihop networks . . . 8
- 1.4 Analyzing the performance of wireless networks . . . 10
- 1.5 Outline of the thesis . . . 11

2 Routing of multicast messages in wireless multihop networks 13
- 2.1 Introduction . . . 13
- 2.2 Motivation . . . 13
- 2.3 Wireless multicast tree problem . . . 14
- 2.4 Review of existing research . . . 17
- 2.5 Contributions . . . 18
- 2.6 Multicast algorithm for type I model . . . 18
- 2.7 Distributed algorithm for type II model . . . 20
- 2.8 Algorithm for multicasting in type III model . . . 23
- 2.9 Summary and conclusions . . . 25

3 Max-min fair link scheduling 27
- 3.1 Motivation . . . 27
- 3.2 Review of existing research . . . 28
- 3.3 Problem statement . . . 28
- 3.4 Outline of the algorithm . . . 29
  - Greedy scheduler . . . 29
  - Implementation of weighted matching algorithm for max-min fairness . . . 30
- 3.5 Summary and conclusions . . . 31

4 Performance analysis of elastic traffic in wireless networks 33
- 4.1 Introduction . . . 33
- 4.2 Balanced fairness . . . 35
  - Assumptions . . . 35
  - Resource allocation . . . 36
  - Properties of balanced fairness . . . 37
- 4.3 Research on balanced fairness . . . 38
- 4.4 Contributions . . . 39
- 4.5 Capacity set of wireless multihop networks . . . 40
- 4.6 Modeling wireless networks . . . 41
  - Effects of transmission power . . . 41

Modeling MAC with a binary constraint model 42
Interference graph example 43
4.7 Computational aspects of balanced fairness in wireless multihop networks 45
4.8 Analysis of a case study 46
System description 46
Value extrapolation 48
Performance of the system 50
4.9 Approximative methods in balanced fairness 51
Flow throughput under light traffic 51
Flow throughput under heavy traffic: polytope capacity set . 53
Flow throughput under heavy traffic: capacity set with a smooth boundary 54
Approximating throughput by interpolation 55
4.10 Balanced fairness with continuous class indices 57
General formulation 58
Low load expansion 59
Example: a two-cell network 60
4.11 Summary and conclusions 62

5 Author's contribution **65**

References **67**

LIST OF PUBLICATIONS

[1] Aleksi Penttinen. Minimum cost multicast trees in ad hoc networks. In *Proceedings of IEEE ICC 2006*, June 2006. 6 pages.

[2] Aleksi Penttinen. Efficient multicast tree algorithm for ad hoc networks. In *Proceedings of IEEE MASS 2004*, pages 519–521, October 2004.

[3] Aleksi Penttinen and Jorma Virtamo. Improving multicast tree construction in static ad hoc networks. In *Proceedings of IEEE LCN 2003*, pages 762–765, October 2003.

[4] Aleksi Penttinen, Iordanis Koutsopoulos, and Leandros Tassiulas. Low-complexity distributed fair scheduling for wireless multi-hop networks. In *Proceedings of RAWNET 2005*, April 2005. 6 pages.

[5] Aleksi Penttinen and Jorma Virtamo. Performance of wireless ad hoc networks under balanced fairness. In *Proceedings of Networking 2004*, pages 235–246, May 2004.

[6] Aleksi Penttinen, Jorma Virtamo, and Riku Jäntti. Performance analysis in multi-hop radio networks with balanced fair resource sharing. *Telecommunications Systems*, 31:315–336, 2006.

[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*, June 2006. 6 pages.

[8] Thomas Bonald, Aleksi Penttinen, and Jorma Virtamo. On light and heavy traffic approximations of balanced fairness. In *Proceedings of ACM SIGMETRICS/Performance 2006*, pages 109–120, June 2006.

1 INTRODUCTION

1.1 Wireless networks

Wireless networks are telephone or computer networks that use radio as their carrier on the physical layer [WP]. In most well-known wireless networks only the last link utilizes wireless technology to provide mobile access to the core network for users. Examples of such systems include, e.g., GSM [3GP] and WLAN [Sta05] networks.

Recently, another paradigm of wireless networking has also attracted considerable attention. In wireless multihop networks datagrams are transferred over multiple wireless links to their destinations. Such networks are applied in situations where there is no existing infrastructure and the communication platform needs to be established within a short time and/or with low costs. Examples of such networks are mobile ad hoc networks [IET], sensor networks [ASSC02], and mesh networks [AW05].

A wireless link has essentially two advantages over a cabled link: it allows at least limited mobility on the part of the end points and avoids the infrastructure costs and effort required for cabling between the terminals. However, radio transmissions are also subject to many constraints which severely limit the performance of the communication. For example:

- *Scarcity of radio bandwidth.* Bandwidth availability is controlled by regulatory authorities. Suitable bandwidths are also limited from the application point of view; high frequencies cause strong signal absorption, which significantly reduces the communication range and potential application environments. On the other hand, low frequencies suffer from low link capacities.

- *Shared nature of radio bandwidth.* Simultaneous transmissions on the same frequency band may interfere with each other, thus reducing the available data rates or completely preventing successful reception.

- *Irregularities in radio bandwidth.* The received signal depends on various propagation effects, such as fading and multipath propagation, which cause fluctuations in the quality of the radio link.

- *Limited capabilities of the network devices.* There are application-dependent limits to the cost, size, and weight of a (mobile) network terminal and to the accompanying battery unit, which have a direct effect on the transmission/reception capabilities of the device.

Despite the challenges of the radio resource, there is a constant need for higher data rates in wireless networks. The evolution of handsets and terminals has made new applications possible and the convergence of networking technologies is leading towards a situation where the resource-intensive applications of wireline networks can be accessed via wireless links. On top of increasing data volumes in wireless networks, the density of different wireless terminals is also growing. This results in decreased transmission resources per terminal as the bandwidth resource remains constant.

Although multiple-antenna technology and programmable radios may make possible significant improvements to the current wireless link rates in the future, the performance of a wireless *network* depends fundamentally on the co-ordination of the radio resource usage. It is important to gain an understanding on the effects of the wireless medium on network performance. It is important to develop methods for the efficient utilization of the radio bandwidth.

1.2 A networking perspective

Teletraffic theory studies quantitatively the three-way relationship between the communications system, the offered traffic and the performance. In this thesis we adopt a networking perspective on wireless networks, especially wireless multihop networks. We study how a group of devices with given individual properties can be efficiently co-ordinated so as to utilize the shared transmission medium in a task the performance of which can only be measured on the network level.

We consider a wireless network as an optimization problem: we set a performance goal for a given network service which depends on the actions of individual network nodes. These actions form the set of variables of the problem. The actions are constrained by radio interference. We are interested in what the optimized performance is and, on the other hand, by which actions the optimized performance is achieved.

Radio channel interference, however, is a challenging constraint. The vast majority of optimization problems arising in wireless networks are extremely complex, which makes accurate solutions, if not impossible, at least impractical to reach. Correspondingly, algorithm development, e.g., in routing or scheduling, for wireless networks often resorts to heuristic algorithms. Similarly, in deriving performance measures for complex wireless networks only approximations of the performance may be available.

In this thesis we model wireless networks and propose methods for solving some of the central optimization problems of a wireless network. The presentation can be divided into two parts. The first part focuses on the development of efficient algorithms for routing and scheduling in wireless multihop networks. The second part is devoted to analyzing the performance of wireless networks.

1.3 Algorithm development for wireless multihop networks

Irrespective of the technology of the network devices, the network level performance, such as the available bandwidth or the delay experienced on a given route, depends on the co-ordination of several nodes participating in the transfer. The co-ordination is a difficult task because of the interference effects which create interdependencies among different concurrent activities in the network.

Assuming that the underlying technology is given, most of the important network level performance problems typically come down to resolving two particular questions: which nodes transmit (the question of routing) and when the nodes transmit (the question of scheduling). These sometimes interrelated tasks are dealt with in the following two chapters. Later in the thesis, in Chapter 4, we refine our approach to the question of what data are being transmitted in a given transmission. This refined viewpoint introduces the question of resource sharing, which makes it possible to analyze performance on the flow level, as the users experience it.

Optimizing the network level performance by routing and scheduling can usually be formulated as a form of a standard integer programming problem. However, the problems tend to become computationally intractable with a large number of variables and this approach is generally not feasible for practical network operations. Gathering the relevant data, solving the optimization problem, and, finally, disseminating control commands back to the network is too time-consuming for most control problems in wireless multihop networks. Furthermore, the associated communication costs may degrade the network performance to such an extent that the optimization process becomes counterproductive as a whole. Thus, in most practical scenarios, optimization should be carried out by simple and preferably distributed algorithms, even though the results remain suboptimal.

Algorithm development is also challenged by the rich variety in the potential applications of wireless multihop networks. The communications needs of different environments cannot be efficiently satisfied by any single algorithm. Routing and scheduling algorithms need to be tailored for different applications. For example, mesh networks are static and can be partially pre-designed, which allows many performance optimization tasks related to routing and scheduling to be carried out offline in the design phase. On the other hand, a mobile ad hoc network of, e.g., quickly maneuvering military vehicles prevents almost any form of optimization as the network topology changes with node mobility, active interference and node malfunction or even destruction. In extreme cases routing and scheduling are based on simple protocols such as flooding and unsynchronized random access protocols, which leave little room for performance optimization. Between these extreme scenarios we have a region of features such as moderate mobility, unreliable terminals or limited processing power, all of which have their characteristic performance problems. It is quite clear that no single algorithm can meet all these challenges.

In this thesis we develop efficient algorithms for two different problems of wireless multihop networks. First, we consider the routing of messages from one source to several destinations. In this case, the optimization problem is as follows. Each transmission has a cost and an attempt is made to minimize the sum of the costs in delivering a message from a source node to all of the destination nodes.

Second, we study the problem of the max-min fair scheduling of single-link flows in a synchronized slotted-time wireless network. In this problem we maximize the bandwidth of all links, with the constraints that the long-term bandwidth shares are approximately max-min fair and that in any time slot a node may only participate in one transmission, either as a transmitter or as a receiver. The maximization is carried out by selecting the set of transmitting flows in each time slot appropriately based on transmission history.

Both these problems are approached by means of simple high-performance heuristic algorithms, accounting also for the distributed nature of wireless multihop networks. In particular, the multicast algorithms developed in this thesis are of great practical importance because of the performance, simplicity, and adaptivity of the algorithms.

1.4 Analyzing the performance of wireless networks

The performance analysis of wireless (multihop) networks constitutes the second part of the thesis. Analytical performance analysis provides a cost-effective means for network dimensioning and understanding the relationship between the system, traffic, and performance. Compared to the alternative methods of prototyping and process simulation, analytic performance analysis makes possible quick results and also provides insights into the behavior of the system. The difficulties of the method arise in the modeling phase; how can a model to be constructed which describes the given performance measure adequately but still remains tractable?

In the contemporary IP-networks, such as the Internet, the vast majority of the traffic is controlled by the TCP-protocol. A TCP-flow typically comprises the transfer of a document, file or message. The transmission can use all the bandwidth that is available but can also adapt the transmission speed to share the bandwidth with other concurrent flows. We refer to this kind of traffic as *elastic traffic*. In this thesis we particularly study the performance of communications networks under elastic traffic conditions.

The performance of elastic traffic is observed on the flow level. In a typical example, a user transferring a file experiences the network performance in the duration of the transfer. In order to model the performance experienced by such a user, we need to characterize the network resources and model how other traffic present in the system interacts with the transfer, i.e. how much contention there is about the resources and how the contention is resolved by resource allocation. However, one must also account for the dynamic nature of the flows. As flows come and go, the resource allocation is also subject to change, which has a fundamental effect on the performance experienced by the flows.

We utilize the concept of balanced fairness [BP03] for the flow-level performance analysis of wireless multihop networks. Balanced fairness is a recently-proposed resource allocation scheme which is especially suitable when analyzing the flow-level performance of communications networks. Balanced fairness considerably simplifies the solution process of the problem, even making explicit formulas possible in certain cases. The performance of the system under balanced fairness is insensitive to traffic details beyond the traffic intensity. Third, the performance under balanced fairness can be used to approximate that of other fair sharing schemes such as proportional fairness and max-min fairness, which are generally intractable in dynamic settings.

We devise novel methods for the efficient computation and approximation of practically interesting performance metrics such as flow throughput.

We also give a formulation of balanced fairness in those cases where the flows are indexed by a continuous variable. We believe that the proposed computational tools and approximation scheme for balanced fairness are important contributions not only in the context of wireless networks but in the performance analysis of communications networks in general.

1.5 Outline of the thesis

The rest of the thesis consists of three independent, yet closely related, chapters. Chapter 2 summarizes the work published in Publication 1, Publication 2, and Publication 3. We describe the multicasting problem in wireless multihop networks and present state-of-the-art multicast routing algorithms for three different instances of the problem. In Chapter 3 we present the results of Publication 4. The chapter describes the max-min fair link scheduling problem and introduces a distributed scheduling algorithm that approximates max-min fair resource sharing. Chapter 4 presents recent developments in the performance analysis of wireless networks under dynamic elastic traffic. In particular, we study the resource allocation concept known as balanced fairness. We develop novel computational and approximate methods for balanced fairness and apply this concept to wireless networks. The publications associated with this chapter are Publications 5, 6, 7, and Publication 8. Finally, Chapter 5 concludes the thesis with a detailed list of the author's own contributions to the reported publications.

2 ROUTING OF MULTICAST MESSAGES IN WIRELESS MULTIHOP NETWORKS

We describe the multicasting problem in wireless multihop networks and present state-of-the-art multicast routing algorithms for three different instances of the problem.

2.1 Introduction

The multicast tree problem in a wireless multihop network is defined as follows: Find a sequence of transmissions which delivers a message from a given source node to one or more destination nodes so that the sum of the transmission costs is as small as possible.

We describe the advantages of multicast routing in wireless multihop networks and introduce the wireless multicast tree problem in a general form. Our contribution consists of three novel multicast routing algorithms providing state-of-the-art solutions to certain common instances of the problem.

Section 2.6 presents an efficient solution to the problem in networks with simple radios. We assume that each transmission of a node is received by all of its neighbors. Thus each node has only one transmission cost. The algorithm attempts to minimize the total cost of the multicast tree.

Section 2.7 assumes a similar model to the above and provides a distributed solution to minimize the number of transmitting nodes in the multicast tree (i.e., a special case in the domain of the algorithm in Section 2.6, where all the transmission costs are equal).

The third algorithm in Section 2.8 addresses the situation where each node can additionally choose the neighborhood that receives the transmission. We assume that for any two given alternative neighborhoods, one with the lower transmission cost is included in the neighborhood with the higher transmission cost. This situation arises, e.g., if the nodes may increase their transmission power to reach more nodes and the transmission costs are related to the power.

2.2 Motivation

Ad hoc networks are typically deployed by a group of people (or vehicles, computers etc.) that has set out to complete a task in an environment where no existing network infrastructure is available. These tasks, such as emergency rescue operations, battlefield missions, or shared desktop meetings, are likely to contain applications (walkie-talkie, live video streams, surveillance data) where a considerable amount of data is delivered to several destinations at the same time.

In mesh networks, where wireless stations interconnect to form an access network, software delivery or streaming services create a similar need.

Efficient query delivery or reporting to several sink nodes simultaneously lengthen the list for a sensor network, where smaller messages can also challenge the communication capability of the network. This one-to-many communications scenario is referred to as multicasting.

In ad hoc networks multicasting has two different interpretations. The group communication viewpoint typically presents multicast communication as a two-directional mailing list. In this context the existing studies (cf. overview in [HDL+02, GM04]) address primarily the problems related to distributed group membership maintenance, i.e. how to add and remove members of the multicast group.

The other interpretation of multicasting, the approach we adopt in this thesis, treats multicasting as an efficient one-directional delivery tree in the network, the sole purpose of which is to minimize the transmission costs of the delivery of the message from a source to several destinations.

Multicast trees make possible cost optimization of one-to-many communications in static or slowly varying ad hoc networks, mesh networks, sensor networks, and multihop extensions of base station-based wireless networks. The costs may represent factors related to, e.g., the load incurred, energy consumption, queuing or processing delay, security, reliability, or detectability. Using a suitable routing tree, the corresponding data transfer typically has substantially lower costs as compared to the two alternatives, sequential unicast transmissions or flooding. The advantages of multicast routing become increasingly evident if either the transferred data are large or the resources are scarce.

Trees can be constructed on operational time scales to a route certain message (or to find a path) or off-line, e.g., to provide solutions to multicast routing subproblems in cross-layer optimization schemes. For example, it is possible to explicitly maximize the lifetime of a network (cf. [CT00a, CT00b]) that uses multicast communications [Vae01].

The use of multicast trees is recognized as an important consideration in traditional fixed networks [Ram96]. The key difference between the multicast tree problems in wireless and traditional wireline networks further underlines the advantage of multicast trees in the wireless setting: in each transmission all the neighbors of the transmitting node receive the same data packet simultaneously; cf. Figure 2.1. Compared to the alternative of sequential unicast transmissions, all but one of the neighboring nodes receive the packet "free of charge". This is the multicast advantage of wireless networks. Note that the advantage applies only to multicast routing: unicast routing in wireless multihop networks is optimized using standard routing algorithms [BG92].

Before exploring the literature in detail, we state our general network model and define the wireless multicast tree problem, WMTP for short.

2.3 Wireless multicast tree problem

Assume a multihop network where nodes communicate with each other using (possibly) one-directional wireless links. In order to describe the problem unambiguously, we define the concepts of neighborhood and transmission tree.

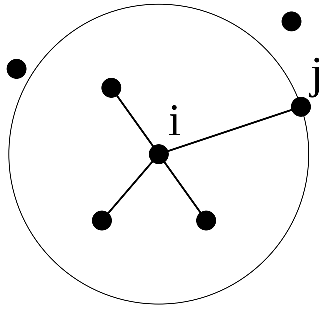

Figure 2.1: Transmission from i to j reaches all nodes within the range in this simplified model. There is only one transmission-related cost, but all the neighboring nodes receive the message.

Each node may have a selection of transmission parameters available. The parameters may be related to, e.g., transmission power, antenna pattern, or to coding and modulation. For each different parameter combination there is a neighborhood, which is defined as follows:

Definition 2.1 *A neighborhood of a node is a set of nodes that receive a message successfully transmitted by the node with certain transmission parameters. Each neighborhood may have a different cost.*

Note that a node may have several neighborhoods with different costs, but each neighborhood belongs to only one node which we refer to as the corresponding node of the neighborhood. A transmission tree is a sequence of transmissions (neighborhoods) that originate from a source node.

Definition 2.2 *A transmission tree is a set of neighborhoods such that each neighborhood either has a parent in the transmission tree or is the source node. A neighbourhood "A" is a parent of neighborhood "B" if the corresponding node of "B" belongs to "A" and there is a recursive sequence of parents from "A" to the source node.*

This is just a restatement of the fact that a node must have received a message before it can forward it. Finally, we may define WMTP as follows.

Definition 2.3 ***(Wireless multicast tree problem)*** *INPUT: the neighborhood costs of all nodes, the source node and a set of destination nodes.*
OUTPUT: A minimum cost transmission tree, which originates from the source node in such a way that all destination nodes belong to at least one neighborhood in the tree.

This neighborhood relation sets rather loose constraints on the network and can be efficiently used to model the effects of transmission parameters. The simplest model of a wireless multihop is a unit disk graph. In this case each node has only one neighborhood: the nodes within the unit disk centered at the node. The definition of WMTP here is more general than in Publication 1.

Although the general definition of the problem is feasible for almost any model of a network, the solution methods of the problem differ considerably. In this thesis we consider three common types of WMTP with the following modeling assumptions:

I only one neighborhood per node. Transmission costs may be arbitrary.

II special case of Type I, with all the transmission costs equal.

III arbitrary number of neighborhoods per node. Transmission costs may be arbitrary, but with an additional constraint: we assume that any neighborhood of a node contains all the lower-cost neighborhoods of the same node.

The type I model reflects a situation where the nodes use simple radios with fixed transmission parameters or only one transmission cost can be maintained per node. Most of the ad hoc network routing protocols [IET] assume this model.

Assuming fixed transmission parameters, WMTP also has a simple graph theoretical interpretation: we attempt to find a minimum weight tree (in a directed graph) which is rooted in a given source vertex and contains a set of destination vertices. The weights are associated with vertices rather than edges and only non-leaf tree vertices are accounted for in the total weight of the tree. In other words, we seek a minimum-weight connected set of nodes, which originates from the source node and dominates all the receiver nodes.

The type II model corresponds to the scenario of minimizing the number of transmissions in the tree. This is often a good rule of thumb in minimizing the load and delay of the multicast operation, especially in cases where the network dynamics prevent the use of more refined cost information.

The type III model arises in the context of minimum power multicasting [WNE00]. In short, the objective is to find the minimum power transmission tree in a scenario where a node may adjust its transmission range d freely with a power cost proportional to d^α, where $\alpha > 2$. High-power transmissions can deliver the message to a large number of nodes, making possible more direct routes towards the destinations but at extremely high cost.

An important special case of WMTP, irrespective of the network model, is the broadcast problem in which all nodes except the source belong to the destination set. In the type II model a node attempts to send a message to all the nodes in the network so that the number of retransmissions is minimized. This problem can be seen as a minimum connected dominating set problem (with the requirement that the source belongs to the set), which remains NP-hard even in unit disk graphs [AWF02], a fact which is a strong motivation for development of simple heuristics for the broadcast problem and for WMTP in general. In type III model the problem becomes even more difficult. In a wireline network the situation corresponds to the minimum spanning tree problem which can be solved in polynomial time using, e.g., Prim's algorithm [Pri57].

Note that the models discussed here do not assume the unit disk model, although it is used in simulations in Publication 1 and Publication 2. Nodes do not have to be identical and, for example, the antennas may have directional beams. Most of the analytical results in the field assume the unit disk model for tractability.

2.4 Review of existing research

In fixed networks the minimum-cost multicast tree problem is usually solved using Steiner tree heuristics [Ram96]. The wireline multicasting costs are related to links whereas the wireless multicasting costs occur at the nodes. This difference follows from the fact that several nodes may receive a single transmission simultaneously. It also renders the wireline multicast tree heuristics inefficient in wireless multihop networks, which thus constitutes an independent research field.

A widely-used model for wireless multihop networks is that each node has only one set of neighbors which receive all the transmissions by the node and the transmission costs are equal, i.e. the type II model. In this setting the broadcast problem, i.e. minimizing the number of transmissions to deliver the message to all nodes, has received a lot of attention in ad hoc networks; see [WC02] for an extensive overview and comparative study. More theoretical approaches to the broadcast problem can be found, e.g., in [SSZ02, LK00, AWF02, GPM03].

Despite the vast interest in the broadcast scenario, there are few references to the efficient multicast algorithms. The multicast routing research into wireless multihop networks has been very active [HDL+02, GM04], but the focus has been on the protocol aspects of multicasting. In one of the very few studies of efficient multicast algorithms (WMTP type I model), [GK98], the authors introduce two heuristic algorithms. Despite the proven worst-case approximation guarantees in unit disk graphs, the algorithms leave much room for improvement in practical applications. Practical approaches to the problem were suggested in [GM03] (type I model) and [CN02] (type II model), where the idea is to construct a virtual mesh that connects the multicast group members by means of unicast tunnels and to compute a source-based delivery tree on the mesh. Obviously, this approach lacks efficiency.

A closely-related problem from graph theory, the node-weighted Steiner tree, is addressed in [KR95] and [GK99]. The main difference between this problem and the corresponding instance of WMTP (type I) is that in WMTP we seek a connected set of nodes that dominates the receivers, rather than containing them.

The type III model originates from the minimum-energy broadcasting and multicasting problem in ad hoc networks. The problem was introduced by Wieselthier et al. in a series of papers [WNE99, WNE00, WNE01].

The main focus of the work was on broadcasting. In [WNE00], the authors developed an algorithm called Broadcast Incremental Power (BIP). In BIP a spanning tree is constructed as in the well-known Prim's algorithm, with the difference that each step considers finding the minimum *incremental cost* that is needed to connect the next node to the evolving spanning tree. After the spanning tree is constructed, redundant transmission are eliminated in BIP by means of a separate algorithm. For the type III multicast problem the authors proposed MIP (Multicast Incremental Power), which simply prunes a broadcast tree constructed by BIP so that the only leaves are the receiving terminals.

The minimum-energy broadcasting problem has been an active topic since its introduction. In [ČHE02] the broadcasting problem is proven to be NP-hard. In [EG01] the authors show that even finding the minimum spanning tree is NP hard in the type III model. The BIP algorithm was further studied in [WaLF01], where the authors derived analytical performance bounds for the algorithm, showing that the graph theoretical approximation ratio of BIP is between $\frac{13}{3}$ and 12. Applying Prim's algorithm directly to generate a link-based minimum spanning tree to approximate the broadcast tree, as also suggested in [WNE00], has an approximation ratio between 6 and 6.33 [Nav05]. The analytical results in the field assume the unit disk model for tractability. Algorithms with improved performance have been suggested in [ČHE02, Yua05] and in [CSS03] the authors present an algorithm that is based on local distance information only. Although many variants of the original broadcast problem formulation have been presented in the literature, the multicast tree construction has not been addressed since MIP.

2.5 Contributions

The contributions in Publication 1 are an efficient algorithm for type I problems (cf. Section 2.6) and an exact enumeration algorithm for the type II model, which significantly extends the size of the problems that can be solved exactly by enumeration. Publication 2 presents a fully-distributed solution for type II problems (cf. Section 2.7). Both the presented algorithms show excellent efficiency and are shown to outperform the existing or obvious approaches.

Publication 3 was motivated by the apparent lack of multicast algorithms in the field. It presents an algorithm designed to produce efficient trees in a type III model (presented in Section 2.8) for small receiver groups and suggests an simulated annealing (cf. [KGV83]) formulation to improve any given tree.

The following sections describe our proposed routing algorithms in detail.

2.6 Multicast algorithm for type I model

The algorithm described next is reported and analyzed in Publication 1. In the algorithm the multicast tree is constructed incrementally, starting from the source and adding new transmitting nodes so that the shortest path distances to non-covered destinations are maximally decreased.

Assumptions of the algorithm:

- Type I network model.
- Shortest-path matrix (where the element with the index (i, j) states

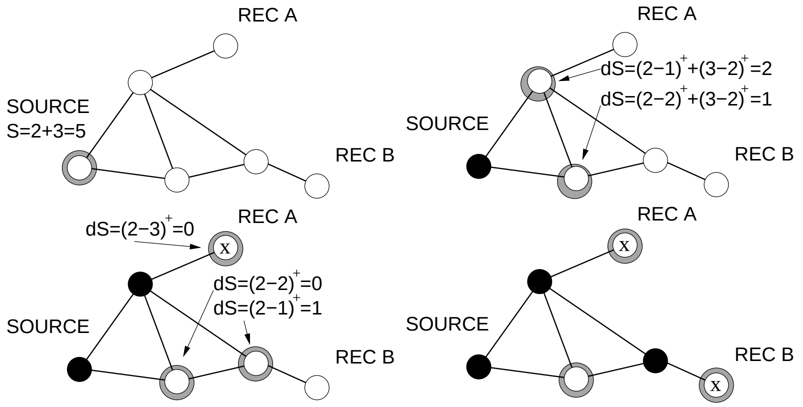

Figure 2.2: Example: Minimize the number of transmission needed to connect SOURCE to REC A and REC B. All node weights equal one, whence the shortest paths are defined in terms of hop count. Let S be the sum of minimum distances between transmitting nodes and non-covered receivers. To start, mark SOURCE as covered and initialize S (see top left figure). In the iteration step, compute dS, the decrement in S if the node was added to the transmitting nodes, for each covered node. Mark the node with maximum dS as transmitting and mark all its idle neighbors as covered. Iteration step is repeated until all receivers are covered. After the selection of the second transmitting node (see bottom left figure) REC A is covered and S contains only the distance to REC B.

the cost of shortest path from node i to node j) which is available in terms of transmission costs.

We may use the term node weight interchangeably with the term transmission cost. The node weight information is fully contained in the shortest-path matrix.

During the algorithm each node belongs to one of the following three sets: covered nodes, transmitting nodes, or idle nodes. The set of covered nodes contains all the non-transmitting nodes which have the data packet, i.e. which are neighbors of a transmitting node closer to the source. Idle nodes are not part of the multicast operation.

The algorithm is described as follows. Initialize all nodes "idle" and set the source node as "covered" as it is the origin of the message. The multicast tree construction proceeds with iteration rounds until all destinations have received the message. In each iteration round we select one node from the covered nodes, mark it as transmitting and mark all of its idle neighbors as covered. The selection is carried out in such a way that the sum of the minimum distances from the transmitting nodes to each of the non-covered multicast receivers is maximally decreased. The iteration stops as soon as all multicast receivers are covered. Figure 2.2 illustrates the algorithm.

Let D be the all-pairs shortest path matrix. Let D_i denote the i:th row of the matrix. The set of nodes that are already covered, but not transmitting, is denoted by K while the set transmitting nodes is denoted by M.

The set of multicast receivers that are not covered is denoted by T. Let $c(i)$ be the shortest distance from any of the nodes in M to node i and denote the vector of $c(i)$s by c. Finally, let $dS(i)$ be the decrement in the sum of shortest path costs from M to T (i.e. in $\sum_{i \in T} c(i)$) if node i was added to M. Using this notation, Algorithm 1 presents a pseudocode implementation of the proposed algorithm. In the algorithm $(\cdot)^+$ and min are applied

Algorithm 1 Centralized multicast tree construction

```
K ← {s}, M ← ∅
c ← D_s
while T ≠ ∅ do
    dS(i) ← Σ_{j∈T} (c(j) − D_i(j))^+, ∀ i ∈ K
    n ← arg max_{i∈K} dS(i)
    M ← M ∪ {n}
    K ← (K ∪ N(n))\M
    T ← T\N(n)
    c(T) ← min(c(T), D_n(T))
end while
Output M
```

component-wise. In each round we select the node which maximally reduces the sum of the shortest distances. In a network with N nodes and R multicast receivers, the worst-case complexity of the algorithm in this straightforward implementation is $O(N^2 R)$, since the selection of the next node has complexity $O(NR)$ and the outer loop will be repeated at most $N - 2$ times. Naturally, the shortest path information needs to be gathered separately.

2.7 Distributed algorithm for type II model

Algorithm 1 requires full topology information (the shortest path matrix) for routing decisions, which requires centralized operation. Distributed implementations of the algorithm pose a significant challenge. However, if we are interested only in minimizing the number of transmissions in the tree, i.e. all the transmission costs are equal, the following distributed algorithm can be applied efficiently. The algorithm is reported and analyzed in Publication 2.

Assumptions of the algorithm:

- Type II model.
- Each node knows the lowest hop counts from itself to the destinations. This information is typically available in the unicast routing tables.
- Each node knows the lowest hop counts from their neighbors to the destinations. This information can be communicated locally.

The main difference to the assumptions in Section 2.6 is that the node does *not* have to know, e.g., mutual distances of the destination nodes. Thus, the algorithm may rely on the underlying unicast routing information.

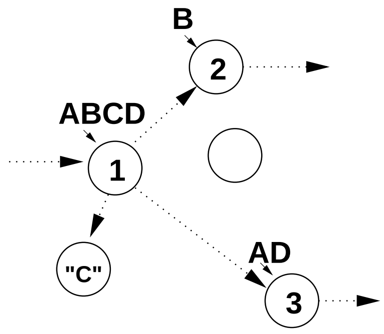

Figure 2.3: Node 1 has to find a multicast tree to the destinations A,B,C and D. A split algorithm at node 1 assigns node 2 to find a multicast tree (path) to B and node 3 to find a multicast tree to A and D. C was found directly from the neighbor list.

The algorithm consists of independent split operations at nodes to construct the tree. When a node needs to send a message to a set of destinations, it divides the routing problem, the destination set, into independent subproblems each of which is resolved by a selected neighboring node, as if it itself was the source of the subset problem. In other words, each node decides independently the next hop neighbor(s) in the tree, i.e. the forwarding nodes and their destination sets, based on its own destination set. We call this operation split. Figure 2.3 illustrates the idea.

The implementation of the split operation is a key factor with regard to the performance of the algorithm. Consider the split operation $\langle i, T \rangle$ in which node i is given the task of forwarding data/finding a route to the set of destinations (a.k.a. multicast receivers) T. Let N_i be the set of neighbors of node i.

The split algorithm at i consists of finding a set of next hop forwarding nodes and dividing the multicast receivers among them. During the algorithm the destinations receive assignments, either permanent or non-permanent, to neighboring nodes of i (which become forwarding nodes). Once made, a permanent assignment remains fixed, but a non-permanent assignment can later be replaced (only) by a permanent one.

Since a transmission of i reaches all the neighboring receivers they are first removed from T. Then the algorithm repeats the following split step until all receivers in T have non-permanent assignments or T becomes empty: Let $B_j(T) \subseteq T$ denote the set of remaining receivers, which can be reached from node $j \in N_i$ with one less hop than from i. In other words $B_j(T)$ is the set of receivers to which a shortest path from i goes via j. Select a forwarding node $m = \arg\max_{j \in N_i} |B_j(T)|$, i.e. the neighbor j for which the number of elements in the set $B_j(T)$ is the largest.

The assignment works as follows. The nodes (destinations) in $B_m(T)$ are assigned permanently to m and removed from T. After that any receiver j in T for which the distance from m to j is equal to the distance between i to j are assigned non-permanently to m without removal from T.

Now only the receivers for which the hop distance would have been increased if routed through m remain in T without an assignment. In case such nodes exist, we repeat the split step until all receivers in T have non-permanent assignments to a forwarding node or T becomes empty. As a result we have generated a set of split problems with forwarding nodes and their assigned destinations.

Note that the existing non-permanent assignments at the end of the split algorithm mean that the distances to the corresponding receivers are not decreased in this split. This is intentional and also important, otherwise the resulting tree would be only a shortest path tree, yet optimized. Following the procedure, the non-permanent assignments remain with the nodes which are responsible for a larger number of destinations, but only if no other forwarding node is on a shortest path to the corresponding multicast receiver.

Obviously, if there is only one multicast destination, the split algorithm selects the next hop node from one of the neighboring nodes on a shortest path to the destination. In practice, this elementary subproblem is readily solved by the underlying unicast routing.

The above algorithm for the split operation is summarized in a pseudo-code implementation in Algorithm 2. In this implementation the non-permanent assignments overlapping with permanent ones are removed in a separate loop.

Algorithm 2 Split operation, input: $\langle i, T \rangle$; forwarding node i, set of receivers T

$A,\ M,\ F_0 \leftarrow \emptyset$
$T \leftarrow T \backslash N_i$
/* $D_i(j)$ denotes the distance between i and j */
$B_j^0(T) \leftarrow \{t \in T \mid (D_i(t) - D_j(t)) = 0\},\ j \in N_i$
$B_j^1(T) \leftarrow \{t \in T \mid (D_i(t) - D_j(t)) = 1\},\ j \in N_i$
while $T \backslash F_0 \neq \emptyset$ **do**
  $m \leftarrow \arg\max_{j \in N_i} |B_j^1(T)|$
  /* add to beginning */
  $M \leftarrow \langle m, B_m^1(T) \cup (B_m^0(T) \backslash F_0) \rangle \cup M$
  $F_0 \leftarrow F_0 \cup B_m^0(T)$
  $T \leftarrow T \backslash B_m^1(T)$
end while
/* remove redundant non-permanent assignments */
for each $\langle m, T_m \rangle \in M$ **do**
  $T_m \leftarrow T_m \backslash A$
  $A \leftarrow A \cup T_m$
end for
Output M /* <node, destination list> - pairs */

The main difference of this algorithm to the centralized one presented in Section 2.6 is in the selection next forwarding nodes. Whereas in the centralized algorithm the selection of the next node to be added to the tree is global, the distributed algorithm forces each node to decide the local tree structure independently of other tree nodes.

2.8 Algorithm for multicasting in type III model

Minimum energy multicasting problem is addressed in Publication 3. The article presents an algorithm designed to produce efficient trees in the type III model for small receiver groups and suggests an simulated annealing (cf. [KGV83]) formulation to improve any given tree. Here we present the suggested algorithm.

Assumptions of the algorithm:

- Type III model
- Neighborhood costs of the nodes are given.

Let c_{ij} be the cost of node i transmitting to its neighborhood j. For concreteness, we adopt the minimum energy multicasting setting, where the neighborhood j is associated to node j; c_{ij} is the required transmission power for node i to transmit directly to node j. Correspondingly, the neighborhood j consists of all nodes which are reached with less transmission power than c_{ij}. All-pairs shortest path matrix can be computed from the neighborhood costs by standard means.

The proposed algorithm (referred to as Incremental Shortest Path Tree (ISPT) in Publication 3), starts with an initial tree and then grafts the receivers one by one to the tree using direct paths in a selected order. It is shown in Publication 3 that the algorithm is especially suitable in cases where the number of multicast receivers is fairly small. The three phases, tree initialization, grafting and sweep are presented next in detail.

The initial tree, trunk, is defined here to be a subtree which originates from the multicast source. In this presentation we choose the initial tree as the shortest path from the source to the most "distant" destination. The initial tree could also be a path from the source to some special node (cf. rendezvous point in PIM-SM for IP multicast) or simply the source node itself.

Starting with initial tree as the current tree, the multicast tree is then constructed incrementally by repeating the following grafting step: For each destination not yet in the current tree, determine the path from the tree to the destination which yields the smallest incremental path cost. Incremental path cost refers to the additional cost required to implement the path from the existing tree. For example, in the initial tree the source node s transmits, e.g., to neighborhood 2 with the cost c_{s2}. Now the *incremental cost* of node s to reach the node 5 (in its neighborhood 5) is simply $c_{s5} - c_{s2}$. The *incremental path cost* of a path from node s via 5 to k is $c_{s5} - c_{s2} + D_5(k)$, where $D_5(k)$ is the shortest path cost from 5 to k.

Having now one possible path for each destination, select the path which has the smallest cost (an intuitive, yet arbitrary choice for grafting order) and attach it to the tree to produce the current tree for the next iteration, see Figure 2.4 for illustration of the algorithm.

In addition to the above described model, the shortest incremental paths can be found by applying the Dijkstra's shortest path algorithm to a modified network. The modified network is the original network with the neighborhood costs in which all the nodes of the current tree are reduced

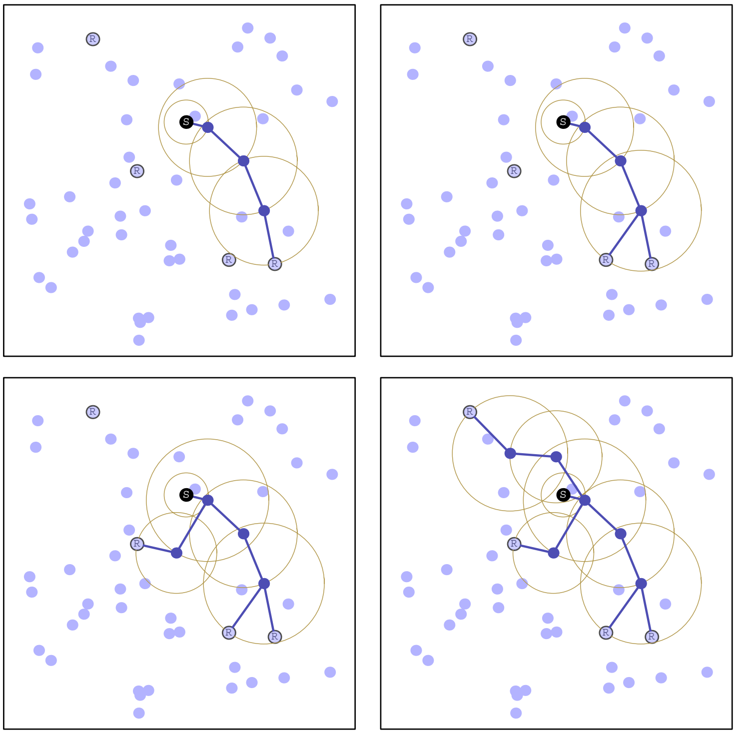

Figure 2.4: Example: Initial tree is the shortest path from the source to the furthermost receiver. The grafting is repeated iteratively by attaching a path connecting the tree to a receiver, for which the incremental path cost is lowest. Algorithm stops after all the receivers are connected.

to one node. The links from this node to its neighbors are determined according to the shortest incremental costs from the tree.

The worst case complexity of the multicast algorithm in a network with N nodes is $O(N^3)$, which corresponds to the spanning tree problem in a fully connected network. However, the computational complexity of the algorithm is roughly the product of the number of nodes in the tree, average number of neighboring nodes and number of receivers.

By different selections of the initial tree and the grafting order the algorithm can be applied to produce different efficient tree structures. After grafting, the resulting tree can be further improved by removing unnecessary transmissions. This operation is called the sweep [WNE00, Yua05]. Description of our sweep algorithm is included in the following summary of the whole approach.

The multicast algorithm is summarized in Algorithm 3 with the following notation. Let s_i be the *state* of node i which tells the current neighborhood assignment of node i. $s_i = 0$ denotes that the node is not transmitting. During the algorithm the transmission tree M is the set of nodes for which $s_i \neq 0$. Let k stand for the source node and set the destination distance list to $l_n = D_k(n)$, $\forall n \in T$, where T is the set of multicast destinations.

Algorithm 3 Type III algorithm operation

Tree initialization: e.g. $M \leftarrow$ shortest path to the receiver $\{\arg\max_n l_n\}$.
while there is a multicast receiver not in any of the neighborhoods of M **do**
  grafting, see Algorithm 4
end while
remove unnecessary transmissions by a sweep operation see Algorithm 5

Algorithm 4 Grafting algorithm

for each transmitting tree node i, for which the state s_i has changed **do**
  for each neighbor j of i **do**
    for each remaining destination n **do**
      if $(c_{ij} - c_{is_i})^+ + D_j(n) < l_n$ **then**
        set $l_n = (c_{ij} - c_{is_i})^+ + D_j(k)$
        associate l_n to (i, j, n)
      end if
    end for
  end for
end for
pick (i, j, n) for which l_n is smallest (or use selected grafting order)
increase the state of node i to c_{ij}
add the route $j \rightarrow k$ to the tree M, i.e. update the corresponding states.

Algorithm 5 Sweep operation

repeat
  list all the nodes V in breadth-first-search from the source node
  for each transmitting node $i \in M$ in the BFS-list **do**
    select $j = \arg\max_{j \in M} c_{ij}$ which does not yet have a predecessor
    set $s_i = j$
    set i to be the predecessor of neighbors $\{k | c_{ik} \leq c_{ij}\}$
  end for
until no changes in S (or use, e.g., two iterations)

2.9 Summary and conclusions

We have described the advantages of multicast routing in wireless multihop networks and introduced the wireless multicast tree problem in a general form. The contribution of this thesis consists of three novel multicast routing algorithms providing state-of-the-art solutions to certain common instances of the problem.

The WMTP optimization approach is limited to networks and applications where the cost information can be kept up-to-date. In highly dynamic scenarios resource optimization is hardly possible and one-to-many communications require less refined means, such as flooding variants [LK00].

In the discussion we treated WMTP as a message forwarding problem, but the same algorithms can also be applied in building multicast trees (virtual tree paths) over any directed graph where the costs are related to nodes rather than links.

Although the contributions presented here provide efficient solutions to the problem instances they are designed for, we identify several interesting open questions for future study:

- Efficient split algorithm to allow distributed multicast operations in the type I model.
- Efficient enumeration algorithm to compute optimal trees in the type I model.
- Analysis of the algorithms when the shortest path information is approximative or contains errors.
- Delay studies of multicast trees under different interference models.
- Efficient algorithm for general WMTP.

3 MAX-MIN FAIR LINK SCHEDULING

This chapter describes the max-min fair link scheduling problem in a wireless multihop network and introduces a distributed low-overhead scheduling algorithm that approximates max-min fair resource sharing. An inherent feature of the approach is its immunity to topology changes as well as to flow traffic variations.

3.1 Motivation

Although the traditional concept of quality of service (QoS) provisioning based on an explicit statement of QoS requirements by each user is well suited for session-based situations with continuous information flow, it requires connection admission control and resource allocation techniques and involves an additional signaling burden. Thus, such provisioning may turn out to be problematic in the presence of mobility and wireless medium dynamics. Alternatively, sessions can specify their satisfaction (utility) as a function of the allocated bandwidth. However, defining user satisfaction in terms of a utility function is not generally feasible. In those cases where users do not specify their resource requirements, an intuitive objective is to split the available resource equally to all sessions. Whenever users cannot utilize a portion of their allocated resource because of some constraint, the unused resources should also be distributed among other sessions. This objective is captured by max-min fairness.

Max-min rate fairness can be provided at the medium access control (MAC) or the network layer. At the MAC layer, fairness properties need to be ensured on a link basis, namely for single-hop flows. At the network layer, fair rates must be provided for end-to-end, multi-hop session flows and this clearly encompasses fairness in single-hop flows. The focus of this work is on fairness at the MAC layer.

In [TS02] Tassiulas and Sarkar introduce the problem of max-min fair link scheduling. The problem is to schedule one-link flows in a conflict-free manner so that the flows attain max-min fair rates. Scheduling has the constraint that no node can simultaneously take part in more than one transmission at any given time.

The approach proposed in [TS02], however, is centralized. Furthermore, the approach utilizes certain sufficient conditions for schedulability to compute max-min fair rates. As the given conditions are not necessary for general networks, the fair rates typically fail to utilize available capacity efficiently. Clearly, it would be desirable to devise a method that combines conflict-free scheduling with minimal node coordination and achieves a good approximation of max-min fair rates without wasting capacity. The method should ideally rely on readily-available local information and should involve minimal signaling load in the network.

We present a low-complexity, low-overhead distributed algorithm for *approximating* max-min fair rates in a wireless network of general topology. Our algorithm is based on the distributed computation of maximum weighted matching of the network graph with appropriately defined link weights. Apart from its simplicity and its low complexity, the algorithm does not require a time frame (however, it needs slot synchronization among nodes) and it can be applicable in the presence of arbitrary topology or channel quality variations and flow traffic demand changes.

3.2 Review of existing research

Existing work on MAC layer max-min fairness can be classified into two categories. The first one refers to single-channel systems with connectionless multiple access, where fairness properties are sought through a random access scheme. In [NKGB00], a framework for implementing fairness by maximizing the sum of user utility functions is proposed; this gives rise to distributed contention resolution methods in order to achieve the desired rates. Max-min fairness arises as an asymptotic case of a special utility function. Another work along the same lines appeared in [LLB00], where max-min fair rates can be achieved using appropriate the flow weights based on adaptation of a back-off timer. However, the algorithm requires the a priori computation of max-min fair rates in order to find the flow weights. More recently, the work in [WK04] shows that max-min fair rates can be attained in the context of Aloha with appropriate adjustment of the access probability of nodes in a distributed fashion. Although these distributed access methods require minimal coordination between nodes, they suffer from severe bandwidth loss as a result of unavoidable collisions. Moreover, fairness is guaranteed only in a probabilistic sense and is meaningful only for large-enough time scales.

The second category of studies comprises connection-oriented multiple access methods, where fairness is solicited with conflict-free link scheduling methods. The authors in [TS02] introduced the concept of max-min fair rate allocation and provide a scheduling policy for achieving max-min fair rate allocation for single-hop session flows and time-slotted systems. Each node assigns service tokens to adjacent links in a round-robin fashion and the weight of a link is the minimum of the stored tokens at the two end nodes. At each slot, the set of links that form the maximum weighted matching of the network graph are scheduled for transmission. This step makes the approach centralized. A distributed slot assignment algorithm that approximates max-min fair bandwidth sharing is presented in [Sal04]. The algorithm is based on local adjustments of link rates by reallocating time slots subject to conflict constraints in an effort to track the corresponding distributed fluid algorithm, which provably converges to max-min fair rates. The methods of this class require some amount of node coordination, but they guarantee collision-free access to resources.

3.3 Problem statement

We consider a time-slotted system with control and data time slots, where a number of control mini-slots precede one data slot. The duration of a mini-slot is much smaller than that of a data slot. A general, non-bipartite network topology graph $G = (V, E)$ is assumed, with vertices representing wireless nodes and links between pairs of nodes showing node connectivity. Let N be the number of nodes in the network. Network-wide slot synchronization is assumed.

There exist J flows in the network. Each flow that traverses a link is represented by a directed edge from the link end-node (the transmitter) to the other node (the receiver). Associated with a network topology graph is the network flow graph $G_f = (V_f, L_f)$, with the same nodes as in the topology graph ($V_f = V$) and edges between each pair of nodes, with each edge corresponding to a distinct flow between those nodes. Note that there may be any number of flows on the same link. We focus only on single-hop flows in this work. A flow is said to be active on a link if it transmits a packet on that link.

We consider nodes that possess a single transceiver, that is, one hardware unit that can be used to set up a distinct communication link. We assume that there exist only primary scheduling conflicts, so that the same node cannot transmit or receive simultaneously on more than one link. Given this assumption, the set of active flows in the network at a specific time instant must constitute a matching of the network flow graph. In the presence of secondary conflicts, i.e. when receiving nodes are interrupted if they hear more than one transmission simultaneously, the problem changes considerably. We do not address the issue in this thesis. The work in [CL87] presents a distributed algorithm for constructing a fixed-length TDMA-based link schedule under a certain fairness metric and secondary conflicts.

A single physical layer transmission rate corresponding to a certain modulation level and/or coding rate is used for every flow, so that time shares are mapped to bandwidth shares. Our model is quite generic as it incorporates the cases of: (i) arbitrary, time-varying packet arrival rates of flows and therefore time-varying bandwidth requirements of flows, and (ii) arbitrary, time-varying topology changes, due to inherent volatility of the wireless networks.

3.4 Outline of the algorithm

Greedy scheduler

Let each flow transmit at most one packet in each slot. Associate each flow j, $j = 1, \ldots, J$ with a weight $w_j = C^{n_j}$, where $C > J$ is an arbitrary number and n_j is the number of time slots that have elapsed since flow j transmitted a packet. When a packet of flow j arrives to an empty queue and waits to be transmitted, then $n_j = 1$. If flow j has no packets to transmit, we set $w_j = 0$. Scheduling conflicts reflect interference constraints and determine eligible sets of flows that are allowed to transmit in the same slot. Consider the collection of eligible flow sets in slot t and let $\mathcal{I}(t)$ denote the set of indices, with each index i corresponding to one such set I_i. This index set in turn depends on the presence of packets at the transmitter as well as on link availability that is affected by topology variations. Each eligible set of flows I_i is referred to as matching set of flows, since it is a matching of the network flow graph.

The algorithm employed by a centralized greedy scheduler is as follows. In an attempt to approximate max-min fair rates, the algorithm selects the matching set of flows with the maximum total weight for transmission at each time slot, namely it selects the set

$$i^* = \arg\max_{i \in \mathcal{I}(t)} (\sum_{j \in I_i} w_j(t)). \tag{3.1}$$

For performance of the greedy scheduler refer to Publication 4.

The best known solution to the maximum weighted matching problem in general graphs is of complexity $O(NJ + N^2 \log N)$ [Gab90]. Several approximations have also been developed for the problem that aim either at linear complexity ([DH03],[Pre99]), or at allowing for distributed implementation [WW03]. The centralized greedy method in which the edge with the largest weight is sequentially inserted in the matching and all conflicting edges are removed has complexity of $O(J \log N)$ (provided that the edges are sorted a priori). The algorithm results in a matching which has provably a weight of at least $1/2$ times the optimum. However, these approaches are not amenable to a distributed implementation of scheduling in wireless networks or involve significant burden of control messages.

The computation of maximum weighted matching also arises as the maximum throughput policy in scheduling in switches with packets queued in the switch input (e.g. [MAW96]), where link weights represent the number of packets waiting to be transmitted. Several variations to the basic approach have also been proposed (see [McK99],[TT03] and references therein). Being designed for input-queued switches, these algorithms are suited for bipartite graphs and are not amenable to distributed implementation, since they imply inter-port communication and involve steps that require centralized coordination such as node sorting.

Implementation of weighted matching algorithm for max-min fairness

We propose a distributed greedy algorithm for max-min fair link scheduling, which is parameterized to allow controlling the trade-off between the scheduling overhead and utilization rate.

A set of mini-slots preceding each data slot will serve the purpose of control information exchange in our approach. Since control messages themselves are subject to collisions, the control overhead is essentially the number of mini-slots required to exchange coordination information in a distributed *and* conflict-free manner. The proposed algorithm attempts to identify a matching with maximum (or at least, as large as possible) weight in a distributed fashion by using the notion of the greedy scheduler.

The key idea is to give priority to flows (edges) with larger weight. Each node is aware of n_j, the number of timeslots elapsed since last transmission of flow j, for each flow that corresponds to an adjacent edge in the flow graph. This determines the weight of the edge, w_j. Due to the fact that each node locally selects the flow with the largest weight, we can employ as the weight w_j any increasing function of n_j and thus we can assume that $w_j = n_j$ in the sequel.

The algorithm consists of R iteration rounds. In each round, each node selects the largest-weight incident flow and broadcasts its decision to its neighbors. This procedure takes place for each node in a control mini-slot. The neighbors that receive this information eliminate all other candidate flows destined to or originated from the node that made the decision. If both end-nodes of a flow select the same flow, this flow is added to the matching set of flows. Otherwise, if a node learns that the other end-node of its selected edge has picked another flow (which has a higher weight), the node becomes idle. The next iteration round is then performed only by the idle nodes. The parameter R controls the tradeoff between the number of required mini-slots and the maximality of the resulting matching. If R is small, some of the nodes may be unnecessarily barred from transmitting and if R is large the algorithm requires a lot of control overhead. The procedure is referred to as Algorithm 6 and its pseudo-algorithm is as follows.

Algorithm 6 Distributed matching algorithm, input graph $G_f = (V_f, E_f)$ and weights $w_e,\ e \in E_f$

$M \leftarrow \emptyset$ /* matching */
$\mathbf{p} \leftarrow$ a random permutation of numbers $\{1, \ldots, N\}$.
/* iteration loop */
for $i = 1$ to R **do**
    $G'_f \leftarrow G_f$
    for $n' = 1$ to N **do**
        $n \leftarrow p_{n'}$
        $S_n \leftarrow$ set of edges in G'_f connected to node n.
        if $S_n \neq \emptyset$ **then**
            $e_n \leftarrow \arg\max_{e \in S_n} w_e$. Ties are broken randomly.
            $w_{e_n} += 0.1$
            $G'_f \leftarrow e_n \cup (G'_f \backslash S_n)$.
            if e_n was already selected by its other end-node **then**
                $M \leftarrow M \cup e_n$
                Remove the end nodes of e_n and the attached links from G.
            end if
        end if
    end for
end for

3.5 Summary and conclusions

We studied the problem of approximating max-min fair rates in a wireless network without explicit node coordination and we present a greedy, low-complexity scheduling algorithm that serves this purpose. In Publication 4 we show that the algorithm outperforms a centralized algorithm of providing max-min fair rates, which in general topologies is based on sufficient conditions for schedulability. The proposed greedy scheduling provides only approximate max-min fair rates, but it is immune to topology or flow traffic variations and its overhead does not depend on the number of links or flows. When flow traffic demands and wireless link availabilities remain unchanged, the schedule is periodic. However, even in the presence of variations, the scheduled flows are continuously adjusted on-the-fly in order to maintain fairness.

There exist several directions for future study. The approach constitutes a first step towards the goal of fair scheduling in the presence of limited, local knowledge about system status. In that sense, it can be extended to more general resource models, such as that of orthogonal frequency division multiplexing (OFDM), which comprises two-dimensional resource allocation.

4 PERFORMANCE ANALYSIS OF ELASTIC TRAFFIC IN WIRELESS NETWORKS

This chapter reviews the concept of balanced fairness in the flow-level performance analysis of communications networks. We discuss how the concept can be applied to wireless networks and devise novel methods for the efficient computation and approximation of practically interesting performance metrics, such as flow throughput. We give a formulation of balance fairness in those cases where the flows are indexed by a continuous variable.

4.1 Introduction

Performance analysis of elastic traffic in wireless networks studies the network as observed by flows, i.e. file transfers. The analysis requires knowledge of network transmission resources and a resource allocation policy, and it needs to account for the dynamic nature of the flows. In other words, we need to know at what rates traffic can be transferred in different parts of the network, how the rates are shared among contending flows, and how the allocation behaves in time as flows come and go.

In a wireless network, the key element affecting the performance is interference. As a result of interference, typically only a fraction of the links can be active simultaneously at any given instant. Such a set of links (which can operate concurrently with rates > 0) is called a transmission mode. When different modes are switched on a fast time scale, the resulting network appears at flow level as a virtual network with links with capacities (in bits/s) which depend on the selection of modes and their respective time shares, i.e. on link scheduling. In wireless networks scheduling may thus be used to transfer network capacity from one part of the network to another, within certain limits.

Resource allocation policy is needed to resolve situations where two or more flows share the same resource, e.g., a link. Resource allocation has a fundamental effect on flow performance. For example, to maximize the total throughput in the network the optimal allocation policy would give a strict priority to flows which consume the least resources. In certain cases, this leads to a situation where flows with relatively high resource costs, e.g., flows traversing several links, end up with zero allocated bandwidth. Indeed, the fairness of bandwidth sharing has been recognized as an important consideration and several fairness concepts have been introduced and analyzed; cf. [MW00, MR02]. In wireless networks, resource allocation must be determined jointly with scheduling.

As flows come and go, the resource allocation changes with each arrival and departure. Naturally, any changes in the allocation during a flow lifetime have direct implications for flow-level performance. This creates an additional challenge for resource allocation and flow-level modeling. Although resource allocation could be made optimal in the sense of a utility function for any given fixed set of flows, this does not necessarily guarantee that the system converges to a steady state that is optimal in the dynamic scenario [BP03]. The analysis of the dynamic setting is generally difficult, even for the simplest network topologies [FdLFL$^+$01].

A new bandwidth sharing concept called balanced fairness (BF) has recently been introduced by Bonald and Proutière [BP03, BP02, BP04]. This is a very interesting notion, for two reasons. First, it leads to network performance which does not depend on any other traffic characteristics except for the traffic intensities on different paths. In other words, the performance under BF is insensitive. Second, systems with BF resource sharing are considerably simpler to analyze in the dynamic setting. For many simple systems, BF permits an explicit analysis.

Simulation studies have shown that in many cases the performance of a network under BF is similar to that under other fair allocation schemes, such as proportional fairness and max-min fairness [BMPV06]. BF therefore provides a useful approximation tool for evaluating network performance. However, BF itself does not represent a solution to a utility optimization problem or guarantee the Pareto efficient use of the resources.

In this chapter we review the basics of balanced fairness and wireless network modeling. We show how balanced fairness can be applied to flow-level performance analysis in a wireless network. The resulting joint problem of resource allocation and scheduling easily becomes computationally heavy. We devise an efficient computational scheme to minimize the per-state computations required for performance analysis. As an alternative performance analysis tool, we suggest approximating the flow throughput under BF. In this approximative scheme, we first determine the throughput at zero load and find the capacity limit, and additionally compute the throughput derivatives at zero load and at the capacity limit. These results are then utilized to interpolate the throughput curve over the load range. This method can be applied to wireless and non-wireless scenarios of arbitrary sizes. Finally, we give a formulation of balanced fairness in a case where flows are indexed by a continuous variable. This construction makes possible the modeling of, e.g., location dependent features of flows. Throughout the section several examples are explicitly worked out to illustrate the proposed schemes.

The organization of the rest of the chapter is as follows. Section 4.2 goes through the definition of balanced fairness. Section 4.3 surveys the related literature. Section 4.4 summarizes the primary contributions of the present work before delving into the details in the following sections. The modeling of wireless networks and the computational aspects of balanced fairness in the context, i.e. the contents of Publication 5 and Publication 6, are discussed in Sections 4.5 and 4.7. The results are then applied in an example from Publication 7, presented in Section 4.8, which also introduces a novel performance analysis method, referred to as value extrapolation. Section 4.9 presents a method for approximating the throughput performance of more complex systems and Section 4.10 gives a formulation of balanced fairness with continuous class indices. These last two sections are extracted from the work reported in Publication 8. Finally the chapter is concluded in Section 4.11.

4.2 Balanced fairness

The performance, measured, for instance, in terms of the average transfer time for a document of a given size, clearly depends on the dynamic behavior of the system and on how the bandwidth is shared between different flows. Therefore, it is necessary to study the system in a dynamic setting where new flows arrive at the network, are transferred across the network, and, upon completion, depart from the system.

The notion of balanced fairness was introduced by Bonald and Proutière in the context of wireline networks [BP03]. Under dynamic resource allocation defined by balanced fairness the dynamic flow-level model of a communications network becomes significantly easier to analyze. Furthermore, the performance of the system is then insensitive to traffic details, such as flow size distributions.

In this section we present the balanced fairness concept. For the most part, we follow the presentation in Publication 8 and define balanced fairness in the usual form for a finite set of discrete flow classes.

Assumptions

The traffic is modeled as flows which are categorized into flow classes. A flow class represents a set of similar flows in terms of network resource usage. In a wireline network, for instance, a flow class could be characterized by a set of links representing a path in the network. Each flow class receives dynamically allocated rates, which are constrained by the system, which is basically modeled as a set of inequalities. We model the flow as a continuous end-to-end stream of data and assume that any change on the rate of the stream takes place immediately throughout the network. In other words, at any flow arrival or departure the resource allocation is instantaneously updated everywhere in the network to correspond the new situation.

Denote the number of different flow classes by N. Let x_i be the number of class-i flows in progress and denote the network state by $x = (x_1, \ldots, x_N)$.

Class-i flows arrive stochastically and have finite, random sizes. Flows are part of *sessions* which consist of random number of flows possibly separated by phase type distributed think times. We assume only that the sessions arrive according to Poisson process – the flow process itself can have almost arbitrary structure, for example including correlations of successive flows, cf. [BP03, FBP$^+$01].

The model is parameterized by the traffic intensities of each flow class. The traffic intensity of class-i flows is defined as the product of the flow arrival rate λ_i and the mean flow size (in bit/s),

$$\rho_i = \frac{\lambda_i}{\mu_i}.$$

We denote by $\rho = (\rho_1, \ldots, \rho_N)$ the vector of traffic intensities.

In state x class-i flows are allocated the total rate of $\phi_i(x)$ bit/s, with the convention $\phi_i(x) = 0$ if $x_i = 0$. The flows in each class share evenly the rate allocated to the class. We denote by $\phi(x) = (\phi_1(x), \ldots, \phi_N(x))$ the vector of allocated rates.

In all states x, the allocation vector must belong to some coordinate convex capacity set $\mathcal{C}$ that represents the physical constraints of the network. In many practically interesting examples, the capacity set is a polytope, that is

$$\mathcal{C} = \{\phi, \phi A \le C\}, \tag{4.1}$$

for some $N \times L$-dimensional non-negative matrix A and L-dimensional positive vector C [BMPV06]. For wired networks with single-path routing for instance, the vector C gives the capacities of the L network links and the matrix A is the incidence matrix, that is the i, l-entry of A is equal to 1 if class-i flows go through link l and equal to zero otherwise.

Resource allocation

An allocation ϕ is said to be balanced if for all states x and all classes $i, j = 1, \dots, N$:

$$\phi_i(x)\phi_j(x - e_i) = \phi_i(x - e_j)\phi_j(x), \quad x_i > 0,\, x_j > 0, \tag{4.2}$$

where e_i represents the N-dimensional unit vector whose components are equal to 0 except for component i which is equal to 1. For Poisson flow arrivals and i.i.d. exponential flow sizes, this balance property is equivalent to the reversibility of the Markov process describing the evolution of the network state. The invariant measure of this Markov process is then given by:

$$\pi(x) = \Phi(x)\rho_1^{x_1} \dots \rho_N^{x_N}, \tag{4.3}$$

where $\Phi(x)$ is the inverse of the product of allocated rates along any direct path from state x to state 0, which in view of (4.2) does not depend on the considered path. This so-called balance function is recursively defined by $\Phi(0) = 1$, and for all states x such that $x_i > 0$,

$$\Phi(x) = \frac{\Phi(x - e_i)}{\phi_i(x)}, \tag{4.4}$$

with $\Phi(x) = 0$ if $x_i < 0$ for some i.

We emphasize that all results are valid under non-Markovian assumptions. The invariant measure (4.3) is *insensitive* to all traffic characteristics except for the vector of traffic intensities ρ provided flows are generated within sessions as discussed previously.

There is a continuum of allocations that satisfy the balance property (4.2), but a single allocation such that $\phi(x)$ belongs to the boundary of the capacity set $\mathcal{C}$ in every state $x \neq 0$. This unique allocation is balanced fairness. Under balanced fairness, the invariant measure (4.3) gives a well-defined state distribution, i.e. the system is stable, provided ρ belongs to the interior of the capacity set $\mathcal{C}$ [BMPV06, BP03]. When the capacity set $\mathcal{C}$ is a polytope (4.1), this is equivalent to the strict component-wise inequality $\rho A < C$.

We assume that this stability condition is satisfied and denote by π the equilibrium distribution of the network state:

$$\pi(x) = \frac{1}{G(\rho)}\Phi(x)\rho_1^{x_1} \dots \rho_N^{x_N}, \tag{4.5}$$

where $G(\rho)$ is the normalization constant,

$$G(\rho) = \sum_x \Phi(x)\rho_1^{x_1} \ldots \rho_N^{x_N}. \tag{4.6}$$

Rewriting (4.4) as:

$$\phi_i(x) = \frac{\Phi(x - e_i)}{\Phi(x)}, \tag{4.7}$$

we deduce that the balance function associated with balanced fairness is recursively defined by

$$\Phi(x) = \min\{\alpha : \frac{(\Phi(x - e_1), \ldots, \Phi(x - e_N))}{\alpha} \in \mathcal{C}\}, \tag{4.8}$$

with the convention $\Phi(0) = 1$ and $\Phi(x) = 0$ for all $x \notin \mathbb{Z}_+^N$. Note that this recursion defines a unique balance function, which in turn defines a unique allocation in view of (4.7). For a polytope capacity set, there are L capacity constraints and the recursion takes the following simple form [BMPV06]:

$$\Phi(x) = \max_{l=1,\ldots,L} \frac{1}{C_l} \sum_{i=1}^{N} \Phi(x - e_i)A_{il}. \tag{4.9}$$

Properties of balanced fairness

For any capacity set $\mathcal{C}$, let:

$$a_i = \max\{\alpha : \alpha e_i \in \mathcal{C}\}, \quad i = 1, \ldots, N. \tag{4.10}$$

By the coordinate convexity of $\mathcal{C}$, this is the maximum rate allocated to class-i flows. Since $\phi(x)$ belongs to the border of the capacity set, this is also the rate allocated to class-i flows when there are no other flows in the network.

Now consider the capacity set obtained by scaling the i-axis by the factor $1/a_i$. The associated balance function $\varphi(x)$, referred to as the scale-free balance function, is given by:

$$\varphi(x) = \Phi(x)a_1^{x_1} \cdots a_N^{x_N}, \tag{4.11}$$

where $\Phi(x)$ denotes the original balance function. The scale-free balance function satisfies:

$$\forall n \geq 0, \quad \varphi(ne_i) = 1, \quad i = 1, \ldots, N.$$

In view of (4.7) and (4.11), we have in all states $x \neq 0$:

$$\phi_i(x) = a_i \times \frac{\varphi(x - e_i)}{\varphi(x)}.$$

A key performance metric is the flow throughput, defined as the ratio of the mean flow size to the mean flow duration. This may be viewed as the *equivalent bandwidth* as perceived by users. By Little's result, the flow throughput γ_i of class-i flows is given by

$$\gamma_i = \frac{\rho_i}{\mathrm{E}[x_i]}.$$

In view of (4.5) and (4.6), we have:

$$\mathrm{E}[x_i] = \sum_x x_i \pi(x) = \frac{\rho_i}{G(\rho)} \frac{\partial}{\partial \rho_i} G(\rho).$$

We deduce:

$$\gamma_i = \frac{G(\rho)}{\frac{\partial}{\partial \rho_i} G(\rho)} \quad . \tag{4.12}$$

Recalling that a_i is the maximum bit rate allocated to class-i flows, we have $\gamma_i \leq a_i$ for all $i = 1, \ldots, N$.

4.3 Research on balanced fairness

An analysis of dynamic flow-level model is generally difficult. Modeling flow arrivals with a Poisson process and flow sizes with exponential distribution leads to a standard N-dimensional Markov system. Balanced fairness makes this system reversible [Kel79], which gives a recursive definition for the equilibrium distribution (in contrast to matrix inversion) and even closed form solutions in the simplest cases.

Balanced fairness has its origins in the article by Massoulié and Roberts [MR00], where the authors discovered that the Markov model for a simple linear network under proportional fairness [Kel97] has a reversible equilibrium distribution which depends only on traffic intensities. This insensitivity result was then further extended to homogeneous grid networks by Bonald and Proutière in [BM01].

Bonald and Proutière showed that the insensitivity to service time distribution is equivalent to reversibility in processor sharing networks [BP02]. This lead to the question, can one define a resource allocation which makes a general network reversible? To this end, Bonald and Proutière characterized in [BP03] the general class of bandwidth allocations having the insensitivity property by applying the properties of Whittle networks ([Ser99]). Balanced fairness was defined as the largest feasible of such allocations.

Insensitivity of balanced fairness to flow arrival process was motivated in [BP03] following the lines of the single resource case [FBP$^+$01]. Flows and the intervening think times are assumed to constitute sessions. The session structure is described by an open stochastic network, where the processor sharing nodes represent flows in resources and infinite servers the think times between the flows. By modeling the flow and think time durations by phase-type distributions, this construction makes possible practically arbitrary session structure and thus arbitrary flow arrivals within the session. Yet, the equilibrium distribution of the system is, by the theorems due to Kelly [Kel79], as if the flows constituted a Poisson process (given that the sessions arrive according to Poisson process).

In [BP03], the authors showed also that max-min fairness and proportional fairness are generally sensitive except in a few particular scenarios. The phenomenon has been also studied by means of simulation [Tim03]. Recently, in [BMPV06] the authors pointed out that if the resource requirements of the flows do not differ significantly, balanced fairness can be applied to approximate the throughput performance of the other two fair allocation schemes. In case the requirements do differ from each other (e.g., in a wireless network links with bad radio conditions can consume most of the time slots, if priority is given to smallest bit rates), the performance of balance fairness and proportional fairness remain fairly similar, while max-min fairness may lead to much lower throughput. The unified approach to the network modeling by the polytope model was also presented in [BMPV06].

Balanced fairness was originally proposed in the context of fixed networks with fixed routes. The generalization of the concept to wireless networks is presented in Publication 5 and the generalization into networks with traffic splitting was presented in [LV05].

Efficient recursive algorithms are known for certain practically interesting network types, notably for trees [BPRV03, BV04, BV05]. Despite the advances, analysis under balanced fairness in general remains difficult. Fortunately, the construction makes possible derivation of some simple performance bounds. Performance of balanced fairness is always better than transmitting the flows in store and forward fashion, which gives a lower bound for the throughput performance [BP04]. This store-and-forward bound was later improved in [Bon06].

4.4 Contributions

Publication 5 is the first extension of balanced fairness outside the fixed network domain. It extends the analysis to cover multihop wireless networks, where the link capacities depended on scheduling. Under certain conditions on the interference structure it gave two different forms of recursion for the balance function. These arise from two alternative ways to describe the capacity set of the network, as will be discussed in Section 4.7.

Publication 6 expands from the results of Publication 4. The paper explores a wide range of alternative network models of wireless multihop networks and gives a general recursion framework for the polytope model. The resulting problems are typically tedious to solve and the focus is put on computational efficiency. The paper presents an efficient algorithm to solve the balance function recursion, by combining the recursion with linear programming duality and the well-known simplex method. Several examples are explicitly worked out.

Publication 7 presents a simple example of the flow-level performance analysis, in a scenario where two base stations are used in a coordinated fashion to serve downloading users on a road between the stations. The performance of balanced fairness method is compared against other allocation policies within the traditional Markovian model. Balanced fairness is shown to outperform other competitors in computational efficiency. The paper presents also a novel approximation method called value extrapolation. Value extrapolation can be applied to approximate any performance measure in a Markov system expressed as the expected value of a random variable which is a function of the system state.

Despite the advances in computational performance analysis presented in Publication 6, computational complexity remains a difficult issue, especially when the number of flow classes is large. Publication 8 takes a different approach to the problem. By determining the throughput and its derivatives at zero load and at the capacity limit, performance of a networks can be interpolated with a reasonable accuracy without the need to solve the recursion. The approach makes possible an approximate analysis of practically any system. Publication 8 extends the notion of balanced fairness also to cases where the flows are indexed by a continuous variable. This situation arises, e.g., in modeling spatial characteristics of the flows.

4.5 Capacity set of wireless multihop networks

Originally, balanced fairness was presented for networks with fixed link capacities. Publication 5 extends the approach to wireless networks where the link capacities can be changed in certain limits by scheduling. In this section we discuss modeling of wireless networks and, in particular, show how the capacity set can be defined for wireless networks.

Even when the capacity set $\mathcal{C}$ of a system is known to be of the polytope form (4.1), the matrices A and C may be difficult to define. Wireless multihop networks (with certain exceptions) is a good example of such a case.

Capacity set of a network can be alternatively described using the instantaneously achievable rate vectors. At any given instant, the network operates in a single transmission mode τ, which defines the controllable communication parameters uniquely at the physical and access layers for each link. For instance, in a wireless network τ may define which links are active or what modulation schemes and transmission powers are used.

At a short time scale, with a fixed τ, the links of the network can be seen as fixed bit pipes, the capacities of which arise from the parameters defined by the mode. Denote the link capacities of mode τ by the $1 \times L$ vector r_τ, where L is the number of links. Without loss of generality we consider only those rates vectors that are maximal, i.e. rate vectors r_τ, only for τ such that there does not exist another τ' with $r_{\tau'} \geq r_\tau$. Let $\mathcal{T}$ denote the set of transmission modes corresponding to the maximal rate vectors and let R be a matrix consisting of the rows $r_\tau, \tau \in \mathcal{T}$.

As the flow-level operations occur at a longer time scale, also all the rates obtained by time multiplexing the vectors in $\mathcal{T}$ are available to flows. This provides an alternative method to define the capacity set. For the links we have the feasible capacity set

$$\mathcal{C}_{\text{link}} = \{r \, : \, r \leq tR, \; te^{\mathrm{T}} = 1, \; t \geq 0\}, \tag{4.13}$$

where $t = (t_1, \ldots, t_{|\mathcal{T}|})$ can be interpreted as the schedule with each component t_τ defining the fraction of time the transmission mode τ is used. In a wireline network there is only one transmission mode, and R has only one vector consisting of the fixed link capacities.

The polytope form (4.1) of the capacity set can be found by defining the convex hull of achievable rates for the links. To this end one enumerates all the instantaneous rate vectors and, additionally, all their axis projections (where any number of components of the vector is replaced by zero). For the resulting set of rate vectors, the facets of the convex hull, each of which characterizes an inequality constraint, can be found using the gift wrapping algorithm [Ski97].

Denote the convex hull of link rates resulting from the above construction by $rD \leq C$. Let A be the routing matrix with $a_{il} = 1$ if class i uses link l and zero otherwise. Now the constraints (4.1) defining the capacity set become simply

$$\mathcal{C} = \{\phi : \phi AD \leq C\}.$$

The gift wrapping algorithm has the complexity $O(m^{\lfloor L/2 \rfloor + 1})$, where m is the number of the rate vectors (instantaneous + axis projections) and L is the number of links. Unfortunately, this construction is not generally feasible. For a trivial example, a wireline network with 10 links consists of maximal rate vector giving the (fixed) link capacities and $2^{10} - 1$ axis projections. Thus, the complexity of finding the full polytope representation of the hypercube spanned by the rate vector would be $O(2^{60})$. Although there is no need to generate the inequalities in such a way in this context, the example illustrates the complexity of the approach. This is the motivation for developing efficient computational schemes directly for the instantaneous rate vector presentation in Section 4.7. The idea is to combine the facet enumeration of the capacity set with the balance function recursion. Before that, however, we discuss how the rate vectors can be found in various models for wireless networks.

4.6 Modeling wireless networks

When the link capacity set is described using the instantaneous rate vectors (4.13), the rate matrix R contains all the required information of the lower layer configurations. Hence, the modeling process comes down to determining the set of instantaneous rate vectors. Generally, one selects the set of lower layer parameters to be included in the model and then enumerates the set of feasible parameter configurations and the associated rate vectors. Finally, only the maximal vectors are stored. This section discusses examples on how the rate vectors spanning the capacity set are found within different network models.

Effects of transmission power

Assume that the link capacities are determined by signal-to-noise and interference ratios (SIR) at the nodes. Let g_{lk} denote the power gain between links l and k, that is the power gain between transmitter of link k and receiver of link l. SIR is defined as

$$\text{SIR}_l = \frac{g_{ll} p_l}{\sum_{k \neq l} g_{kl} p_k + \nu_l} , \tag{4.14}$$

where ν_l is the noise power at the receiving end and p_l the transmission power on link l.

Information theoretic bound. One of the most important channel models in the performance analysis of ad hoc networks is the well-known Shannon model, which, combined with the BF-resource sharing, yields the best possible insensitive performance in the given network scenario.

Shannon capacities [Sha49] give the theoretical limits for the link rates in an AGWN channel of bandwidth W, which depend explicitly on the other active links and noise. In this model the link rates are given by (for a given transmission power configuration τ)

$$r_l(\tau) = W \log_2 (1 + \mathrm{SIR}_l) \quad . \tag{4.15}$$

Although the set of different transmission modes cannot be enumerated as each p_l can have a continuous value between 0 and some maximum transmission power, the spanning rate vectors needed for R use only the extreme power settings for the links, as noted in [LA03]. Hence, in the information theoretic model it is sufficient to consider the transmission modes in which each link is either "off", or "on", the latter meaning that the transmitter is sending with full power. There are $2^L - 1$ candidates for the spanning vectors. For these the corresponding link capacities follow directly from (4.15) and (4.14).

Note that in this model we have assumed that all the links can be active simultaneously. In practice, hardware and software configurations of the network nodes may invalidate this assumption. For example, it may not be possible to transmit and receive simultaneously or to receive multiple concurrent signals.

Threshold model, STDMA. In the threshold model, a set of links (which satisfy the possible physical constraints) can operate simultaneously if the signal to noise ratio is adequate at each receiver. In the simplest case, the capacity of link l is constant c_l when SIR_l exceeds a given threshold s_l and zero otherwise. Such models are often used in the context of Spatial-TDMA (introduced in [NK85]), where the active links in each time slot must satisfy the above-mentioned condition [BVY03].

One may distinguish two cases in the threshold model, the ones with and without power control, as, e.g., in [JX04]. In both cases one starts the construction of R by enumerating all combinations of active links as in the Shannon model. Denote the set of link combinations by $\mathcal{T}'$.

If the links use fixed power, $p_l = p_{\max, l}$ for all active links l, R is constructed as follows. For each $\tau' \in \mathcal{T}'$ check whether all $\mathrm{SIR}_l \geq s_l$ when the active links use their respective maximum powers. If the condition holds, the rate vector is included in R.

If power control is assumed, $p_l \in [0, p_{\max, l}]$, nodes can adjust their transmission powers to minimum acceptable levels to avoid causing unnecessary interference ([Som02]). In this case, for each, $\tau' \in \mathcal{T}'$ check whether there exists a feasible power vector that attains $\mathrm{SIR}_l = s_l$ for all $l \in \tau'$, that is, whether the corresponding set of linear equations has a feasible solution. If the condition holds, the rate vector is included in R.

The model is also easily extended by introducing several thresholds for different rates. This model can be adapted to approximate closely any discrete rate system.

Modeling MAC with a binary constraint model

In some scenarios the link activity is constrained by physical limitations or access control protocols. Such constraints can often be accounted for using a pairwise link constraint model. In a common MAC-layer model two links can be simultaneously active either with some predefined capacities or not at all. This reflects the situation that the channel is locally reserved for one link among competing transmissions. On the other hand, if the attenuation (power gain) in the SIR formula (4.14) is a very steep function of distance, the threshold model can be approximated by the pairwise model.

In the literature, the pairwise link constraint models appear frequently with some differences in how the link constraints are defined. An elementary access model sets the following constraints to the links in the network. A node may not transmit and receive simultaneously and it cannot transmit or receive more than one packet at a time. In other words, all the links connected to a given node belong to different transmission modes. This is often used as an access model with the assumption that other transmissions in the vicinity of the node can operate without conflict using locally distinct frequencies ([HS88, TS02]).

A more detailed MAC model would entail that no two links can be simultaneously active if either of the receiving ends is interfered by the other transmission. In the model presented in [Ari84, NK85] a transmission can prevent reception everywhere within the transmission range, whereas in the widely applied protocol model by Gupta and Kumar [GK00] the interference depends on the locations of the transmitting node so that the closest (with a selected margin) transmission can be successfully received.

Obtaining the rate vectors in the constraint model. Assume that the constraints are defined for link pairs only. Such pairwise constraints can conveniently be handled using a link graph, cf. the flow contention graphs in [HB01, LLB00], conflict graph in [JPPQ05] or the compatibility matrix in [NK85].

Given a network and a set of flows with their routes, the corresponding link graph is constructed as follows: each directed link in the network (that is in use in the scenario) is mapped to a vertex and an edge connects two link graph vertices if the corresponding links cannot be active simultaneously. Conversely, an independent set on the link graph corresponds to a set of links that can be active simultaneously. Thus, the feasible transmission modes are obtained by enumerating the independent sets (cf., e.g, [BK73]) of the link graph and the corresponding rate vectors R result from associating the pre-defined capacities to the links in the independent sets.

This model can also be used in conjunction with the power control models to eliminate the infeasible parameter configurations. To generate the rate vectors in this hybrid case, one reduces the set of all combinations of active links into the set of feasible combinations of active links (where, e.g., active links do not share a node) by stating the feasibility constraints in the link graph and enumerating the independent sets. Then the corresponding link rates in the feasible combinations are determined using the SIR-based models.

Interference graph example

The following example of interference modeling is presented in Publication 5. We assume a six-node network with three flow classes and five active unidirectional links of unit capacity, as shown in Fig. 4.1. The interferences shown in the link graph result from the protocol model [GK00] and the link

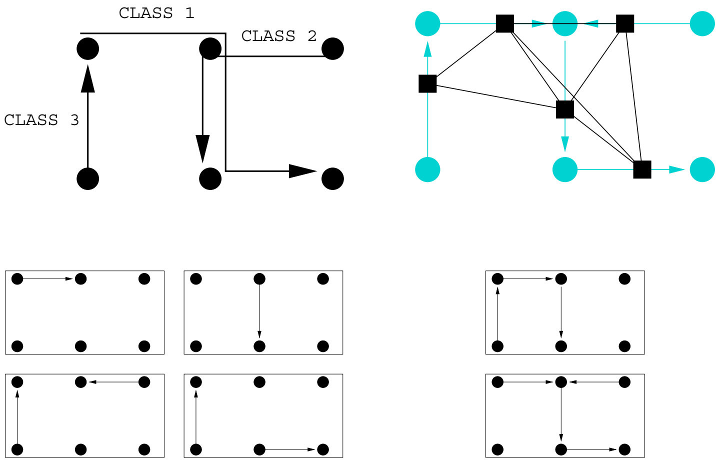

Figure 4.1: Flows, link graph, maximal independent sets and cliques of the interference graph example.

graph has four maximal independent sets and two maximal cliques. There are two alternative ways for defining the capacity set of the system. The first approach defines the spanning rate vectors of the capacity set. The capacity set of the system is given by (4.13) with

$$R = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix},$$

where R is the matrix consisting of the transmission modes (independent sets in the link graph). Routing matrix A maps the capacity constraints of the links to that of flows. In this case,

$$A = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

We note that in this example the link graph is triangulated (i.e. it contains no induced cycles other than triangles) and thus a perfect graph (cf. [Die00]). Publication 5 shows that in this case one obtains the inequality form of the capacity set by identifying the sets of mutually contending transmissions, i.e. sets of transmissions contending for a common time slot. Such transmissions constitute a clique in the corresponding link graph. Each clique q imposes a necessary condition on the bandwidth allocation, which is also sufficient in the perfect graph case. The most stringent set of conditions is set by the maximal cliques, i.e. cliques that are not a subset of another clique. Thus, we have the necessary conditions for a feasible bandwidth allocation

$$\sum_{l \in q} \sum_{i=1}^{3} \phi_i(x) A_{il} \leq 1 \ , \qquad \forall q \in \mathcal{Q} \ , \tag{4.16}$$

where $\mathcal{Q}$ denotes the set of all maximal cliques. The maximal cliques can be enumerated, e.g., by an algorithm from [BK73].

Equations (4.16) define thus the capacity set in form (4.1) with

$$A = \begin{pmatrix} 3 & 2 \\ 2 & 1 \\ 0 & 1 \end{pmatrix}, \qquad C = \begin{pmatrix} 1 & 1 \end{pmatrix}.$$

4.7 Computational aspects of balanced fairness in wireless multihop networks

In this section we adapt the concept of balanced fairness to wireless networks. We define the general form of recursion for capacity sets which are characterized by the instantaneous rate vectors. The recursion step takes the form of LP problem, the solution of which can be facilitated by utilizing the corresponding results from previously computed states.

If the capacity set is defined as the convex hull of available link rates, we may rewrite the recursion step (4.8) as the following LP problem:

$$\begin{aligned} \Phi(x)^{-1} &= \max_{t,\beta} \beta, \\ tR &\geq \beta\Theta(x), \\ te^{\mathrm{T}} &= 1, \\ t &\geq 0, \end{aligned} \tag{4.17}$$

where $\beta = 1/\alpha$ and $\Theta(x) = \{\Phi(x-e_1), \ldots, \Phi(x-e_N)\}A$. Here A is the routing matrix.

By dividing the constraints by β and changing the variable as $t/\beta \to q$, we get rid of the parameter β and can write the problem more compactly

$$\begin{aligned} \Phi(x) &= \min_q qe^{\mathrm{T}}, \\ qR &\geq \Theta(x), \\ q &\geq 0. \end{aligned} \tag{4.18}$$

Each component of the vector q represents here the duration an extremal rate vector (a given transmission mode) is used. One may imagine the vector $\Theta(x)$ in the space spanned by the link capacities and the LP problem is now to express Θ with the help of rate vectors R using the minimal total time.

Publication 6 suggests an efficient computational algorithm for the problem. The important observation is that only $\Theta(x)$ in (4.18) depends on the state of the system. Therefore, we suggest to apply the LP duality [BSS93] to devise an efficient computational method to solve the balance function.

The dual problem of (4.18) is given by

$$\begin{aligned} \Phi(\mathbf{x}) &= \max_y \Theta(x)y^{\mathrm{T}}, \\ yR^{\mathrm{T}} &\leq e, \\ y &\geq 0. \end{aligned} \tag{4.19}$$

The dual problem has the same optimum value as (4.18), see ([BSS93]), but its advantage is that the constraints do not depend on $\Theta(x)$. Hence, an optimal solution y^* to the problem in state x is also a feasible solution for the dual problem in any other state x'.

This leads us to propose integrating the recursion and the solution of the LP problem into a single problem. The idea of the approach is to calculate the new value of the balance function in the recursion by using a stored solution of (4.19) in some previously calculated state. If the solution is not optimal, we can update it by the standard simplex iterations (cf., e.g., [BSS93]) until the optimum is found.

Compared to the standard simplex method applied to the primal problem (4.18), this approach has two significant advantages.

- We need to find only one starting feasible solution for the whole recursion. In contrast, solving the LP problem by standard means would first require finding a feasible starting solution at each state.
- Given that the iteration is started from the optimal solution for a state which is close (possibly adjacent) to the current one, the simplex iteration converges in a very small number of steps to the optimum. The number of iterations the simplex method would require to converge to the optimum from a arbitrary feasible solution could be large.

Although the algorithm can be elegantly described using the simplex and the dual problem, it is often more efficient to apply the dual simplex algorithm ([Tah97]) directly to the primal problem (4.18), which is then supplemented with slack variables.

In the dual simplex algorithm we store the optimal bases (sets of rate vectors defining the solution) of the primal solutions instead of the solutions themselves. By duality, the bases satisfy optimality conditions but are not necessarily feasible. A dual simplex iteration first checks whether the current basis is feasible (and thus optimal), i.e. whether the corresponding facet of the capacity set is penetrated by the direction $\Theta(x)$, and if not, replaces one of the rate vectors in the basis by another thus defining a neighboring facet. Thus the algorithm searches the intersection point by "crawling" towards the point facet by facet on the surface of the capacity set. Publication 6 discusses additional computational aspects related to the formulation.

Although the computational difficulty of the recursion is generally dominated by the number of states instead of the per-state effort, significantly more complex systems to be analyzed numerically by the presented algorithm.

4.8 Analysis of a case study

We present the system described in Publication 7 here as an example of the modeling process involved in the balanced fairness approach. We compare the system performance under balanced fairness with two alternative resource allocation schemes: maximum throughput and max-min fairness. To facilitate the analysis under the alternative schemes we develop a novel approximation scheme referred to as value extrapolation.

System description

Two base stations, A and B, are used in a co-ordinated fashion to serve elastic traffic, or file downloads, destined to users located on a road between

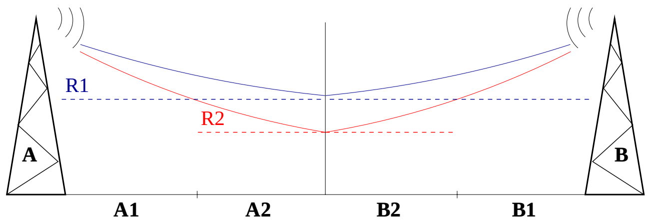

Figure 4.2: Example system with two base stations. Areas A1 and A2 are served by the station A and areas B1 and B2 are served by the station B. If both the stations are active simultaneously the maximum rate at A2 or B2 decreases from R1 to R2 due to interference.

the base stations, cf. Figure 4.2. The base station nearest to the user is always used for connection. 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 $x = (x_1, x_2, x_3, x_4)$, 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 $S = \{x_1, x_2, x_3, x_4 \mid x_i \geq 0, \forall i\}$. In computations we use a truncated state space which is denoted by 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 $\phi(x) = (\phi_1(x), \phi_2(x), \phi_3(x), \phi_4(x))$ 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 R be the matrix comprising of row vectors each of which specifies an instantaneous transmission mode under the constraints described above:

$$R = \begin{pmatrix} R_1 & 0 & 0 & R_1 \\ 0 & R_2 & R_2 & 0 \\ R_1 & 0 & R_2 & 0 \\ 0 & R_2 & 0 & R_1 \\ 0 & R_1 & 0 & 0 \\ 0 & 0 & R_1 & 0 \end{pmatrix}. \tag{4.20}$$

For example, the first row of R represents the mode where both base stations serve the flows in the nearest class (areas A1 and B1 are being served), the second row represents the mode where both the stations are active and serve the traffic in the center-most areas. The policy rate vector ϕ can take the form of any row of R and, additionally, any convex combination of the rows. These are available through time multiplexing, which is assumed to take place on a fast time scale compared to flow durations.

The alternative definition of the capacity set can be found by enumerating the bounding facets of the convex hull of available rates. The feasible rate vectors $\phi(x)$ are constrained by $\phi(x)A \le C$, where,

$$C = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}.$$

Assuming that $\frac{1}{2}R_1 < R_2 < R_1$,

$$A = \begin{pmatrix} \frac{1}{R_1} & 0 & \frac{R_2}{R_1^2} & \frac{R_1 - R_2}{R_1^2} \\ \frac{1}{R_1} & 0 & \frac{1}{R_1} & \frac{R_1 - R_2}{R_1 R_2} \\ 0 & \frac{1}{R_1} & \frac{R_1 - R_2}{R_1 R_2} & \frac{1}{R_1} \\ 0 & \frac{1}{R_1} & \frac{R_1 - R_2}{R_1^2} & \frac{R_2}{R_1^2} \end{pmatrix}.$$

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

$$a_{33} = a_{24} = \frac{1}{R_1}.$$

Value extrapolation

In order to compare the system performance under balanced fairness to that under max-min fairness and maximum throughput policy we assume that the system behaves like a Markov system, i.e. the flows arrive according to a Poisson process and the flow sizes are exponentially distributed. To facilitate the analysis of the other operational policies, we introduce a novel approximation scheme referred to as value extrapolation.

The idea of the value extrapolation is to consider the system in the MDP (Markov Decision Processes) setting, cf., e.g., [Tij94, Dzi97], and to solve the expected value of a performance measure from the Howard equations written for a truncated state space. Instead of a simple truncation, the relative values of states just outside the truncated state space are estimated using a polynomial extrapolation based on the states inside. This leads to a closed system and, unless the system is heavily loaded, allows one to obtain accurate results with remarkably small truncated state spaces. Here we give a formal definition of the method.

A policy ω specifies a feasible capacity allocation in each state. When a policy ω is given, the state transition intensities $q_{x,y}(\omega)$ are known. Assume now that the performance measure is described as a revenue rate $r_x(\omega)$ at state x and that we are interested in the expected performance of the system as it evolves in time, i.e. the mean revenue rate $r(\omega)$. Although this measure can be determined using the steady state probabilities as

$$r(\omega) = \sum_x r_x(\omega)\pi(x),$$

the probabilities are computationally tedious to obtain, as discussed above. An alternative characterization of $r(\omega)$ provides a way to approximate $r(\omega)$ with significantly higher accuracy.

Let $v_x(\omega)$ be the relative value of state x, i.e. the expected difference in cumulative revenue over infinite time horizon when starting from state x rather than from equilibrium:

$$v_x(\omega) = \mathrm{E}\left[\int_0^{\infty} (r_{X(t)}(\omega) - r(\omega))\, dt \;\middle|\; X(0) = x\right],$$

where $X(t)$ is the state process. With a given policy and performance measure, the steady state average revenue rate can be determined by solving the so-called Howard equations [Tij94],

$$r_x(\omega) - r(\omega) + \sum_{y \in S} q_{x,y}(\omega)(v_y(\omega) - v_x(\omega)) = 0, \quad \forall x. \tag{4.21}$$

In the truncated state space there are $|S'|$ equations and $|S'| + 1$ variables. The expected state values are fixed only up to an additive constant, because only the differences $v_y(\omega) - v_x(\omega)$ occur in the equations; hence we may set, e.g., $v_0(\omega) = 0$. Note that the mean revenue rate $r(\omega)$, our performance measure, is one of the unknown variables solved from this group of equations.

The idea of the value extrapolation method is to calculate $r(\omega)$ in a truncated state space, which essentially means that we assume something on the behavior of the relative values outside the truncated state space. The simplest truncation to some set S' is to set $q_{x,y} = 0\ \forall x \in S', y \notin S'$. Regarding to the relative values of the states, this corresponds to setting $v(\ldots, N+1, \ldots) = v(\ldots, N, \ldots)$ in the Howard equations, where N is the maximum number of flows in the truncated state space.

The truncation can be done more intelligently if the relative values of the states behave smoothly outside the truncated state space. More accurate results are achieved if the outside values are extrapolated using the values inside the region. First order polynomial extrapolation is

$$v(\ldots, N+1, \ldots) = 2v(\ldots, N, \ldots) - v(\ldots, N-1, \ldots),$$

and the second order extrapolation is

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

A strong motivation for this procedure is that the value extrapolation leads to exact results in certain simple cases. Consider for example an M/M/1-queue, with a policy that allows free entry to the system and with a cost (negative of the revenue) reflecting the total time in the system (which by Little's result is proportional to the mean queue length). The cost rate in a given state is then simply the number of customers in that state, i.e. the state index itself. Let arrival rate be λ, service rate μ and denote $\rho = \frac{\lambda}{\mu}$. Now the Howard equations can be written as

$$i - r + \lambda(v_{i+1} - v_i) + \mu(v_{i-1} - v_i) = 0, \quad \forall i > 0.$$

The equations are clearly solved by

$$r = \frac{\rho}{1-\rho}, \qquad v_{i+1} - v_i = \frac{i+1}{\mu - \lambda},$$

from which by setting $v_0 = 0$, we get

$$v_i = \frac{1}{2}\frac{i(i+1)}{\mu - \lambda}.$$

The behavior of the relative value is a simple quadratic polynomial of the state variable. Thus, extrapolating the relative value with the second order extrapolation yields exact value for $r(\omega)$ no matter how small the truncated space is. It can be reasoned that for any system with cost related to the time in system, the relative values of states are at least asymptotically (i.e. for higher states) quadratic functions of the state occupancy and therefore one can expect the second order extrapolation to work reasonably well.

The advantage of value extrapolation is that even a few states in the truncated state space may be enough to get relatively accurate estimates for the performance measure. The downside is that the Howard equations need to be solved. The efficiency of value extrapolation is demonstrated in an example in Publication 7.

Performance of the system

As an example we study the mean number of flows in the system without detailed class separation. The detailed analysis can be found from Publication 7. Note that in this analysis we assume Poisson arrivals and exponential flow sizes.

We compare the following operating policies:

- Static policy. The scheduling is fixed so that each class receives equal and constant rate irrespective of the number of flows in the classes.
- Maximum throughput policy. We solve the optimal policy for each state in minimizing the mean flow duration from the Howard equations by a method called the policy iteration [Tij94].
- Max-min fair policy. In each state we allocate the bandwidth to flows according to the max-min fair criterion: A rate allocation is max-min fair if and only if any flow rate cannot be increased at the expense of any higher rate.
- Balanced fairness. As described in previous section.

Figure 4.3 illustrates the mean number of active flows with different system loads. The MDP policy has the best performance but the dynamic policies are all 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.

This example illustrates clearly the advantages of balanced fairness. The computational effort is significantly lighter compared to the analysis with other policies. With BF one has to go the state space through only once and no matrix inversion is needed. Value extrapolation facilitates the analysis of other policies. We observe also that BF seems to be a reasonable approximation of the performance of the system in the dynamic setting. We reiterate that the results under BF remain insensitive to traffic details unlike under other policies.

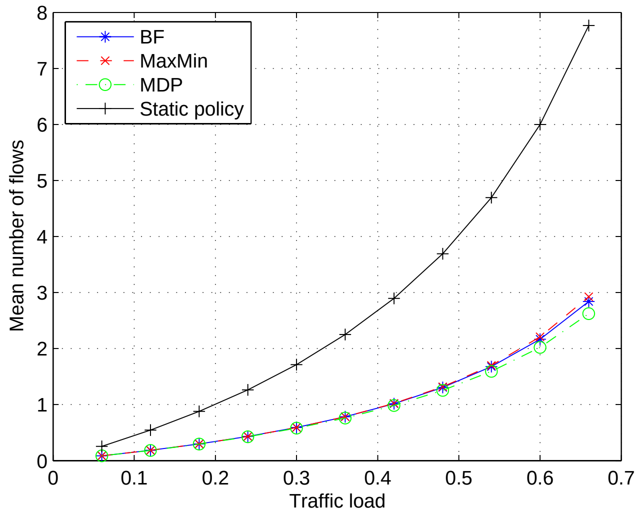

Figure 4.3: Mean number of active flows with parameters $R_1 = 5$ and $R_2 = 1$

4.9 Approximative methods in balanced fairness

While straightforward in principle, the numerical evaluation of practically interesting performance metrics like per-flow throughput is feasible for limited state spaces only, besides some specific networks where the results are explicit. Publication 8 takes another approach to the performance analysis, which will be presented in this section. We study the behavior of balanced fairness in light and heavy traffic regimes and show how the corresponding performance results can be used to approximate the flow throughput over the whole load range. The results apply to any network, with a state space of arbitrary dimension.

The organization of this section is as follows. First we show how the flow throughput and its derivative on low loads can be determined from the balance function. Then the corresponding analysis is carried out at the capacity limit of the system. We separate the cases when the capacity set is (a) a polytope and (b) arbitrary smooth boundary. Finally the light and heavy traffic regimes are combined in an interpolation scheme.

Flow throughput under light traffic

The light traffic regime (when $\rho \to 0$) can be seen as an empty system to which a single flow of studied class arrives. By (4.10), the throughput of the flow is a_i, but how does the throughput behave when we start increasing the load in given proportions?

To this end we determine the derivative of the throughput at zero load. We use the recursion (4.8) to calculate the normalization constant (4.6) up to the second order (i.e., only the states up to occupation $\sum_i x_i = 2$ are taken into account). Using the scale-free balance function $\varphi(x)$ discussed in Section 4.2, recalling that $\varphi(0) = \varphi(e_i) = 1$, and introducing the shorthand notation $\varphi_{ij} = \varphi(e_i + e_j)$, we have

$$G(\rho) = 1 + \sum_i \frac{\rho_i}{a_i} + \sum_{i \geq j} \varphi_{ij} \frac{\rho_i \rho_j}{a_i a_j} + \dots, \tag{4.22}$$

where $\varphi_{ii} = 1$ and, in view of (4.8),

$$\varphi_{ij} = \min\{\alpha : \frac{a_i e_i + a_j e_j}{\alpha} \in \mathcal{C}\} \quad \forall i \neq j. \tag{4.23}$$

The parameter φ_{ij} can be described as the contraction factor needed to bring an $a_i \times a_j$ rectangle inside $\mathcal{C}$, as depicted in Figure 4.4. Since $\mathcal{C}$ is convex, we have $1 \leq \varphi_{ij} \leq 2$ for all i, j.

In the case of a polytope capacity set (4.1), expressions (4.10) and (4.23) take the forms

$$\begin{cases} a_i = \min\limits_{l=1,\dots,L} \dfrac{C_l}{A_{il}}, \\ \varphi_{ij} = \max\limits_{l=1,\dots,L} \dfrac{a_i A_{il} + a_j A_{jl}}{C_l} \quad \forall i \neq j. \end{cases} \tag{4.24}$$

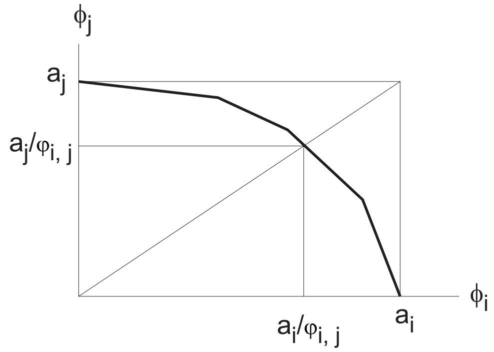

Figure 4.4: The contraction factor $\boldsymbol{\varphi_{ij}}$.

In order to study the behavior of the flow throughput we define a load line with a given traffic profile $p = (p_1, \dots, p_N)$, with $|p| = 1$. By this we mean the set of all $\rho \in \mathcal{C}$ such that the proportions of the loads in different classes are given by p. Let $\hat{\rho}$ be the end point of the load line on the boundary of $\mathcal{C}$. Then the load line consists of points $r\hat{\rho}$, where $r \in [0, 1]$. We wish to characterize the function $\gamma_i(r)$ on a given load line. Using (4.12) and (4.22) we easily find

$$\begin{cases} \gamma_i(0) = a_i, \\ \gamma_i'(0) = -\Big(\hat{\rho}_i + a_i \sum\limits_{j \neq i} \hat{\rho}_j \dfrac{\varphi_{ij} - 1}{a_j}\Big). \end{cases} \tag{4.25}$$

It is worth noting that in order to calculate this derivative what is essentially only needed are the values of the φ_{ij} and these can easily be found by the pairwise consideration of (4.23) as illustrated in Figure 4.4.

Flow throughput under heavy traffic: polytope capacity set

A polytope capacity set (4.1) arises when flows are contending for L resources with respective capacities $C_1, \ldots, C_L$, and each unit of bandwidth allocated to class-i flows requires A_{il} units of capacity from resource l. As shown in [BMPV06], this model describes a rich variety of communication networks.

Let $\hat{\rho}$ be a load vector on the boundary of the capacity set and denote by $\hat{\varrho} = \hat{\rho}A$ the corresponding resource demand vector. Note that $\hat{\varrho}_l = C_l$ for all l in some non-empty set $\mathcal{L}$ of resource indices (otherwise, $\hat{\rho}$ would not be on the boundary of the capacity set). As in the light traffic analysis, we denote by $\gamma_i(r)$ the class-i flow throughput on the load line $r\hat{\rho}$, where $r \in [0, 1]$. We are interested in the derivative $\gamma_i'(1)$ for those classes i such that $\gamma_i(1) = 0$.

A single saturated resource. If $\mathcal{L}$ reduces to a single resource l, we have $\gamma_i(1) = 0$ if and only if class-i flows use resource l, that is if $A_{il} > 0$. For these classes, the heavy traffic regime is then determined by that resource only, in the sense that:

$$\gamma_i(r) \sim \frac{C_l}{A_{il}}(1 - r), \tag{4.26}$$

when $r \to 1$, from which we deduce:

$$\gamma_i'(1) = -\frac{C_l}{A_{il}}.$$

This is the result one would obtain if there were no other resource constraint than l. Note that the derivative depends on the direction $\hat{\rho}$ through the saturated constraint l only.

The asymptotic result (4.26) is a consequence of the following general bounds [BP04]:

$$\max_{k=1,\ldots,L} \frac{A_{ik}}{C_k - \varrho_k} \leq \gamma_i^{-1} \leq \sum_{k=1}^{L} \frac{A_{ik}}{C_k - \varrho_k}. \tag{4.27}$$

Recall that γ_i^{-1} corresponds to the ratio of the mean duration to the mean size for class-i flows. The left-hand side inequality means that the mean flow duration is higher than if there were no other resource constraint than k, for all $k = 1, \ldots, L$; the presence of other resources makes data transfers longer. The right-hand side inequality means that the mean flow duration is less than the sum of the mean flow durations due to each individual resource constraint. We refer to this inequality as the store-and-forward bound since it was originally derived in [BP04] in the context of wired networks with single-path routing: the mean flow duration increases when flows are successively transmitted, in a store-and-forward way, on each link of their path in the network. While the bounds (4.27) are proved in [BP04] in the particular case where the elements of the matrix A are equal to 0 or 1, the same inequalities are satisfied for a general matrix A. The proof is essentially the same as in [BP04].

The resource demand vector is equal to $r\hat{\varrho}$ as a function of r, where $r \in [0,1]$. If $\mathcal{L}$ reduces to a single resource l, it follows from (4.27) that

$$\frac{A_{il}}{C_l(1-r)} \leq \gamma_i^{-1} \leq \frac{A_{il}}{C_l(1-r)} + \sum_{k \neq l} \frac{A_{ik}}{C_k - r\hat{\varrho}_k}.$$

where we used the fact that $\hat{\varrho}_l = C_l$. The asymptotic result (4.26) then follows from the fact that $\hat{\varrho}_k \neq C_k$ for all $k \neq l$.

Several saturated resources. When the set $\mathcal{L}$ consists of more than one resource, we have $\gamma_i(1) = 0$ if and only if class-i flows use at least one resource in $\mathcal{L}$, that is if $A_{il} > 0$ for some $l \in \mathcal{L}$. While the asymptotic values of the bounds (4.27) do not coincide, it is conjectured that the heavy traffic regime is given by the store-and-forward bound, namely

$$\gamma_i(r) \sim \frac{1-r}{\sum_{l \in \mathcal{L}} A_{il}/C_l},$$

from which we deduce:

$$\gamma_i'(1) = -\frac{1}{\sum_{l \in \mathcal{L}} A_{il}/C_l}. \tag{4.28}$$

This is suggested by the result of Schweitzer [Sch79] who proved that the throughputs in multiclass closed networks with a large number of customers, which correspond to the store-and-forward model in heavy traffic, are given by the proportional fair allocation and the results of Massoulié [Mas05] showing the asymptotic equivalence of proportional fairness and balanced fairness. A check with the examples presented in [BV04, BMPV06] confirms that the heavy traffic regime is indeed given by expression (4.28).

Again, the derivative (4.28) depends on the direction $\hat{\rho}$ through the set of saturated constraints $\mathcal{L}$ only, i.e. the derivatives have constant values on any given facet of the polytope, similarly on any given edge between two facets, etc. In particular, the flow throughput is not a continuous function of the load vector $\hat{\rho}$: in approaching the boundary of the capacity set, the flow throughput behaves differently depending on how many constraints are saturated at $\hat{\rho}$. Thus the directional derivative of the flow throughput has discontinuities at edges of the boundary where two or more facets of the polytope meet.

Flow throughput under heavy traffic: capacity set with a smooth boundary

When the capacity set is constrained by a smooth surface instead of a polytope, virtually no exact results are known. Since on a facet of a polytope the directional derivative depends only on the facet plane itself and any smooth surface can locally be approximated by its tangent plane, it is tempting to think that the directional derivative of the throughput at any point $\hat{\rho}$ on the boundary is determined as soon as the tangent plane at $\hat{\rho}$ is known. This thinking, however, is fallacious. The difficulty becomes obvious if one tries to approximate the boundary by a polytope and considers the limit when the polytope becomes denser. Depending on whether $\hat{\rho}$ is on a facet of the approximating polytope or at an edge of two or more facets, one gets a different result. There seems to be no simple way to resolve this ambiguity in the limit process.

There is, however, a special system with curved boundary that yields to exact analysis, viz. the case where the boundary of the capacity set is a N-dimensional L^α-ellipsoid. In Publication 8 we derive the directional derivative of throughput at the capacity limit for the corresponding scale-free L^α-sphere. Here we present the result in the general case of an ellipsoid, obtained by the scaling transformation, cf. Section 4.2.

The L^α-ellipsoid is defined by the surface

$$\left(\frac{\phi_1}{a_1}\right)^\alpha + \cdots + \left(\frac{\phi_N}{a_N}\right)^\alpha = 1, \quad \alpha > 1,$$

where the $a_i,\ i = 1, \ldots, N$, are positive scaling constants and α parameterizes the curvature. The asymptotic throughput is obtained by rescaling from the results in the Appendix of Publication 8,

$$\gamma_i = \frac{2\,a_i}{(N+1)\alpha - (N-1)} \, \frac{1 - ((\frac{\rho_1}{a_1})^\alpha + \cdots + (\frac{\rho_N}{a_N})^\alpha)}{(\frac{\rho_i}{a_i})^{\alpha-1}}. \tag{4.29}$$

Now consider the directional derivative of γ_i along the load line defined by the traffic profile $(p_1, \ldots, p_N)$, i.e.,

$$\hat{\rho}_i = \left(\frac{p_i^\alpha}{(\frac{p_1}{a_1})^\alpha + \cdots + (\frac{p_N}{a_N})^\alpha}\right)^{1/\alpha}.$$

Substituting $\rho = r\hat{\rho}$ into expression (4.29) and forming the derivative with respect to r yields

$$\gamma_i'(1) = -\frac{2\,\alpha\,a_i}{(N+1)\alpha - (N-1)} \left(\frac{(\frac{p_1}{a_1})^\alpha + \cdots + (\frac{p_N}{a_N})^\alpha}{(\frac{p_i}{a_i})^\alpha}\right)^{\frac{\alpha-1}{\alpha}}. \tag{4.30}$$

When $N = 2$, the result is useful as any smooth curve at a given point can be approximated up to second order by an L^α-ellipsoid, so that the two first derivatives match to those of the original curve, cf. Publication 8. In systems with smooth capacity set and $N > 2$, further research is required to find alternative methods of determining the derivative of the flow throughput at the capacity limit.

Approximating throughput by interpolation

In this section we demonstrate how the above light and heavy traffic results can be used to find an approximate throughput function along given load line, $\rho = r\hat{\rho}$, $r \in [0, 1]$.

There are two steps in our approach. First, we find the low-load throughput $\gamma_i(0)$ and its derivative $\gamma_i'(0)$ using (4.25) and the derivative at capacity limit $\gamma_i'(1)$ using (4.28), when the capacity set is a polytope. If the capacity set has a smooth boundary, the heavy traffic regime requires a different

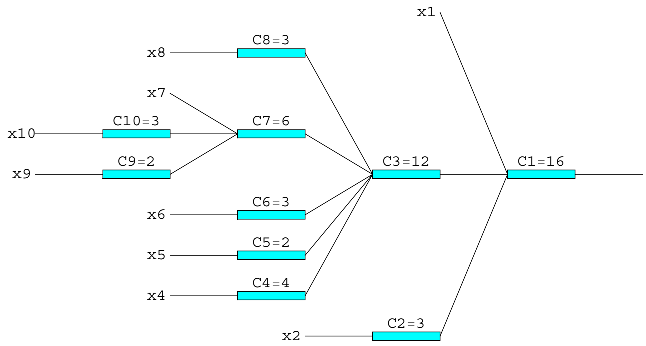

Figure 4.5: An access network with tree topology.

treatment. One can attempt to approximate the boundary with a system for which the derivative is known. In case $N = 2$, a good approximation is obtained by fitting the parameters a_1,a_2 and α of the L^α-ellipsoid so that $\hat{\rho}$ is on the ellipsoid and that the first and second derivatives of the ellipsoid surface match those of the boundary of the original capacity set. Then (4.30) yields the throughput derivative.

In the second step, having found $\gamma_i(0)$, $\gamma_i'(0)$ and $\gamma_i'(1)$, we fit a rational expression of the form

$$\tilde{\gamma}_i(r) = \gamma_i(0)(1-r)\,\frac{1+c\,r}{1+d\,r}, \tag{4.31}$$

where c and d are free parameters, to give the right derivatives $\gamma_i'(0)$ and $\gamma_i'(1)$. The form of this function is motivated by the observation that in all explicitly solvable cases with a polytope capacity set, e.g., trees [BV04], the throughput is given by a rational expression where the degree of the numerator is higher by one than that of the denominator.

Tree network example. As an example consider an access network of tree topology depicted in Figure 4.5. The network consists of 10 links and 9 flow classes (numbered according to the corresponding access link; class 3 does not exist). The capacity set is a polytope; the corresponding capacity vector C and incidence matrix A follow similarly as in previous examples.

$$\begin{aligned} C &= (16,3,12,4,2,3,6,3,2,3), \\ A &= \begin{pmatrix} 1&0&0&0&0&0&0&0&0&0\\ 1&1&0&0&0&0&0&0&0&0\\ 1&0&1&1&0&0&0&0&0&0\\ 1&0&1&0&1&0&0&0&0&0\\ 1&0&1&0&0&1&0&0&0&0\\ 1&0&1&0&0&0&1&0&0&0\\ 1&0&1&0&0&0&0&1&0&0\\ 1&0&1&0&0&0&1&0&1&0\\ 1&0&1&0&0&0&1&0&0&1 \end{pmatrix}. \end{aligned}$$

We wish to study the class-10 throughput along the load line with equal traffic loads. The link that first becomes saturated is link 3. This happens

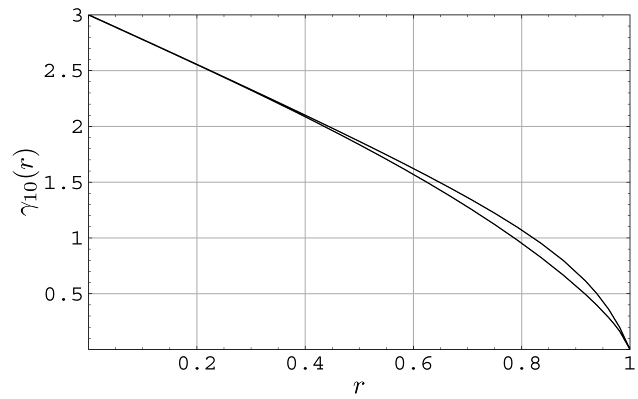

Figure 4.6: Approximate and exact class-10 throughput of the tree network with equal loads (upper and lower curves, respectively).

at $\hat{\rho}_i = \frac{12}{7}$ for all i. A mechanical calculation using Eqs. (4.24) and (4.25) then yields $\gamma_{10}(0) = 3$ and $\gamma'_{10}(0) = -\frac{141}{64}$.

At the capacity limit with $\hat{\rho}_i = \frac{12}{7}$ for all i, link 3 solely is saturated. Then (4.28) immediately gives $\gamma'_{10}(1) = -12$. The same fitting procedure as before yields the rational approximation shown in Figure 4.6 (upper curve). The lower curve represents the exact result that is available for all tree networks [BV04] but with a heavier computational procedure. The approximation is good, though there is a slight deviation in the heavy traffic regime.

The reason for this deviation is basically the fact that even though link 3 only is saturated at the capacity limit, also link 1 is close to saturation: its load is $9 \cdot \frac{12}{7} \approx 15.4$ while its capacity is $C_1 = 16$. In other words, the considered load line ends on a facet of the polytope at a point that is close to an edge. In such cases the directional derivative at the capacity limit is not a very good descriptor of the overall behavior of the throughput curve. Though the derivative given by (4.28) is correct at the very end of the curve, the curve bends rather sharply and soon, with a load slightly below the capacity limit, the derivative is closer to a value predicted by (4.28) assuming all the constraints active at the edge to be saturated. In the present example, if we do assume that also link 1 was saturated, then (4.28) gives $\gamma'_{10}(1) = -1/(\frac{1}{12} + \frac{1}{16}) = -\frac{48}{7}$. Using this value in the fitting renders the match with the exact curve almost perfect (not shown in the figure).

4.10 Balanced fairness with continuous class indices

In this section we formulate the concept of balanced fairness for networks with an infinite number of flow classes. We then give the low load expansion for the flow throughput, allowing us to calculate derivatives at zero load to any desired order.

General formulation

We consider here the balanced fairness concept in systems where the flow classes are indexed by a continuous variable $x \in \mathcal{A} \subseteq \mathbb{R}^d$. Flows are assumed to arrive according to a Poisson process with the arrival density $\lambda(x)$ and to have the mean size $1/\mu(x)$; the load density is denoted $\rho(x) = \lambda(x)/\mu(x)$. Whereas in the discrete case the state of the system is specified by a vector where an entry for each class gives the number of flows in that class, in the continuous index case we have to go to 'sparse matrix notation' and instead describe the locations of the flows present in the system. So the system state is specified by set of indices of the active flows $\xi = \{x_1, \dots, x_n\}$, where n is any non-negative integer. The set of all possible states is denoted by Ξ. Note that ξ is an unordered set of indices, i.e. any permutation of the indices in the set ξ gives just another, equivalent label for the same state. We also assume that $\lambda(x)$ has no atoms, whence the probability that any two arriving flows have exactly the same index is zero, so we can restrict ourselves to states where no two indices are equal.

As in the discrete case, the balance requirement is satisfied if the service rates are of the form

$$\phi(x, \xi) = \frac{\Phi(\xi \backslash \{x\})}{\Phi(\xi)}, \ \ \forall x \in \mathbb{R}^d, \ x \in \xi, \ \xi \in \Xi,$$

in which $\Phi(\cdot)$ is an arbitrary function. In this case the state probability density function reads

$$f(\xi) = \frac{1}{G}\Phi(\xi) \prod_{x_i \in \xi} \rho(x_i),$$

where G is the normalization constant.

Balanced fairness refers to the balanced allocation where the resource usage is as efficient as possible. Let $\mathcal{C}^\xi$ be the capacity set for the constellation ξ. Analogously with the discrete case, cf. (4.8), the balance function is defined recursively,

$$\Phi(\xi) = \min\{\alpha : \frac{(\Phi(\xi \setminus \{x_1\}), \dots, \Phi(\xi \setminus \{x_n\}))}{\alpha} \in \mathcal{C}^\xi\},$$

that is, we remove the active flows in any order until we reach $\Phi(\emptyset)$, which can be fixed arbitrarily, e.g., $\Phi(\emptyset) = 1$. The system is stable if and only if $(\rho(x_i), \dots, \rho(x_n)) \in \mathrm{int}(\mathcal{C}^\xi)$ for all $\xi \in \Xi$.

If the capacity set is a polytope, cf. (4.1), for all ξ, we may write

$$\Phi(\xi) = \max_{l=1,\dots,L^\xi} \frac{1}{C_l^\xi} \sum_{i=1}^{n} \Phi(\xi \setminus \{x_i\}) A_{il}^\xi, \tag{4.32}$$

where C^ξ is a vector of length L^ξ and A^ξ is an $n \times L^\xi$ matrix, both defined for the constellation ξ.

Typically, determination of A^ξ for all ξ is cumbersome and in numerical evaluation of the recursion (4.32) one uses an alternative formulation along the lines of (4.18). Applying this alternative representation, the balance function $\Phi(\xi)$ is evaluated recursively as follows:

1. Enumerate all m possible rate vectors for the constellation ξ. These vectors form the rows of the $m \times n$ rate matrix R^ξ. Only the spanning rate vectors need to be included, but it is not necessary to eliminate any non-spanning vectors.

2. Denote $\Theta = (\Phi(\xi \setminus \{x_1\}), \ldots, \Phi(\xi \setminus \{x_n\}))$, $e = (1, \ldots, 1)$ and $y = (y_1, \ldots, y_m)$. Now the value of the balance function is obtained from the LP problem

$$\Phi(\xi) = \min_y \; e \cdot y^{\mathrm{T}},$$
$$yR^\xi \geq \Theta,$$
$$y \geq 0.$$

Low load expansion

The normalization constant can be written as a series in terms of multiple integrals with progressively larger number of active flows,

$$\begin{aligned} G = 1 + \int_{x\in\mathcal{A}} \Phi(x)\rho(x)dx \, + \\ + \frac{1}{2!}\int_{x\in\mathcal{A}}\int_{y\in\mathcal{A}} \Phi(x,y)\rho(x)\rho(y)dxdy + \ldots . \end{aligned} \tag{4.33}$$

Here and hereafter, the concise notation $\Phi(x_1, \ldots, x_n) \stackrel{\text{def}}{=} \Phi(\{x_1, \ldots, x_n\})$ is adapted. The factorial in (4.33) compensates for the fact that in an n-fold integral over the full range each state (unordered set of indices) appears $n!$ times. Hereafter, we suppress the integration limits with the understanding that all the integrals are over the full range.

The throughput $\gamma(x)$ is defined in the usual way,

$$\gamma(x) = \lim_{dx\to 0} \frac{\rho(x)}{\mathrm{E}\left[X(x, x+dx)\right]/dx},$$

where $X(x, x + dx)$ is the number of flows in the interval $(x, x + dx)$. The expectation of this occupancy can be identified from the normalization constant by noting that the multiplier of dx in an n-fold integral gives the unnormalized probability density of having n flows in the system, one of them about point x, and that a given dx can be found at n places in an n-fold integral. Thus we get,

$$\gamma(x) = \frac{1+\int \Phi(y)\rho(y)dy + \frac{1}{2!}\iint \Phi(y,z)\rho(y)\rho(z)dydz + \ldots}{\Phi(x)+\int \Phi(x,y)\rho(y)dy + \frac{1}{2!}\iint \Phi(x,y,z)\rho(y)\rho(z)dydz + \ldots}. \tag{4.34}$$

In general, calculating the throughput exactly from (4.34) is difficult. However, it provides a means to expand the throughput in terms of the load parameter. As in the discrete case we define the scale-free balance function $\varphi(x_1, \ldots, x_n) = a(x_1)\cdots a(x_n)\Phi(x_1, \ldots, x_n)$, where $a(x)$ is the allocation for a sole flow with index x. Denoting $\varrho(x) = \rho(x)/a(x)$ we expand the expression to the second order, which is sufficient for, e.g., computation of the two first derivatives at $\rho(x) \to 0$,

$$\begin{aligned}\tfrac{\gamma(x)}{a(x)} &= 1 - \textstyle\int(\varphi(x,y)-1)\varrho(y)dy \\ &\quad -\tfrac{1}{2}\textstyle\iint(\varphi(x,y,z)-\varphi(y,z))\varrho(y)\varrho(z)dydz \\ &\qquad + \textstyle\int\varphi(x,y)\varrho(y)dy\int(\varphi(x,y)-1)\varrho(y)dy + \ldots .\end{aligned} \tag{4.35}$$

The throughput analysis of systems with continuous class index is generally more difficult than in the discrete class case. The light load behavior is determined by the above equations and the throughput can be evaluated from (4.34) to any desired order by applying, e.g., the Monte Carlo method to the multiple integrals.

This is itself an important achievement, since for instance a straightforward process simulation (in time) of the system does not easily work for determining $\gamma(x)$, because in the simulation no samples are obtained with the flow index exactly equaling x. In the heavy load regime, however, less is known. We can determine the stability limit where the throughput goes to zero but the directional derivative of the throughput at this limit is not known (this problem is left for future work).

Example: a two-cell network

As an example we study a two-cell network in linear configuration illustrated in Figure 4.7. Mobiles are located on the segment between the two base stations. The total load ρ is assumed to be distributed uniformly along the unit distance between the base stations.

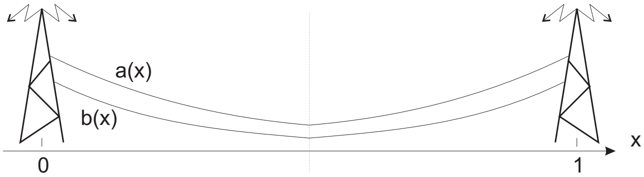

Figure 4.7: Linear two-cell system.

The system has three active transmission modes: either base station 1 is active, base station 2 is active, or both of them are active. Assuming a continuous link adaptation we have an infinity of flow classes indexed by the location $x \in (0,1)$. Each flow is served by the closest base station. Now $a(x)$, the maximum feasible rate of a mobile at point x, is achieved when only the closest base station is active. When both the base stations are active we denote the rate by $b(x)$ with $b(x) < a(x)$. For later use we give a special notation x^* for the point in $(0, 1/2)$ where $a(x^*) = 2\,b(x^*)$. If $a(1/2) < 2\,b(1/2)$, i.e. the equality is satisfied nowhere in the interval, we define $x^* = 1/2$.

We use the Shannon capacity formula with the standard signal attenuation behavior

$$\begin{aligned} a(x) &= \log_2\Big(1 + \tfrac{x^{-\alpha}}{\nu}\Big), \\ b(x) &= \log_2\Big(1 + \tfrac{x^{-\alpha}}{\nu + (1-x)^{-\alpha}}\Big), \end{aligned}$$

where α is the attenuation exponent and ν is the normalized noise power (normalized by the signal power at distance 1). The above formulae hold for $x \in [0, 1/2]$. Values for $x \in [1/2, 1]$ are obtained by symmetry, $a(x) = a(1-x)$ and $b(x) = b(1-x)$.

By the results of [BBP05] we know that the capacity of the system is:

$$C = \Big(\int_0^{x^*} \frac{1}{b(x)}\, dx + 2 \int_{x^*}^{1/2} \frac{1}{a(x)}\, dx \Big)^{-1},$$

in the sense that the network is stable if and only if $\rho < C$. To further analyze the system we consider the derivative at x when $\rho \to 0$. We denote it by $\gamma'(x)$, though the derivative is taken with respect to ρ (and not x). It follows from (4.35) that:

$$\gamma'(x) = -a(x) \int_0^1 \frac{\varphi(x,y) - 1}{a(y)}\, dy. \tag{4.36}$$

To compute $\varphi(x, y)$ we need to construct the capacity set $\mathcal{C}^\xi$ separately for all flow pairs (x, y). Without loss of generality we may assume that $x \in (0, 1/2)$. Then the numerator of the integrand can be split into three different cases (cf. Figure 4.8):

$$\varphi(x,y) - 1 =$$
$$= \begin{cases} 1, & y \le 1/2, \\ \min\Big(1, \frac{a(x)}{b(x)}(1 - \frac{b(y)}{a(y)})\Big), & 1/2 < y \le 1-x, \\ \min\Big(1, \frac{a(y)}{b(y)}(1 - \frac{b(x)}{a(x)})\Big), & 1-x < y \le 1, \end{cases}$$

which allows numerical evaluation of (4.36).

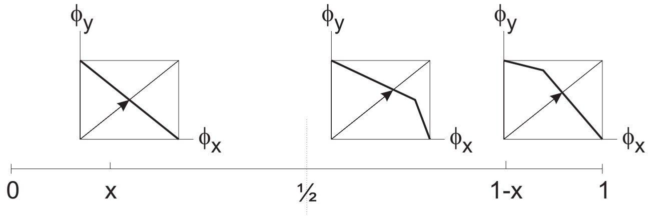

Figure 4.8: Balanced fair allocation for two flows at x (fixed) and y (three different regions).

As mentioned above, the derivative of the throughput at the capacity limit is a difficult problem, but we may still fix the two parameters of an interpolating function of the form (4.31) if we match also the second order derivative at load $\rho \to 0$. The second derivative can be found analogously to the first order case starting from (4.35). For brevity, the details are omitted. Figure 4.9 shows the resulting approximative throughput curves for the whole load range for flows located at points $x = 0.1$, 0.3, 0.5, with the assumed system parameters $\alpha = 3$ and $\nu = 1$. The figure presents also the numerical evaluation of (4.34) when the integrals both in the numerator and denominator were calculated up to 3rd, 5th and 8th order by Monte Carlo integration (numerical evaluation of the balance function in the LP form). This shows that a direct calculation of the throughput from the low load expansion is possible up to about one half of the maximum load, and it also suggests that the interpolation approximation is accurate.

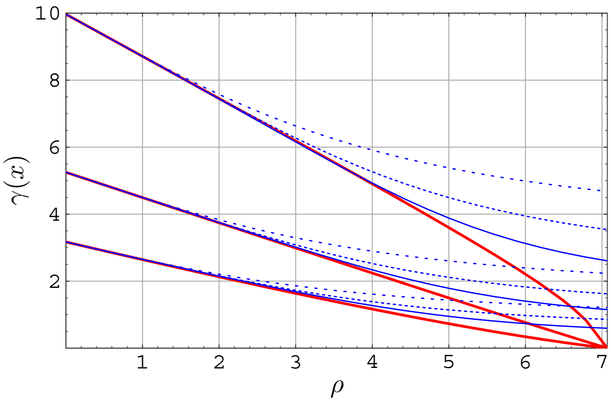

Figure 4.9: Flow throughput at points $x = 0.1, 0.3, 0.5$. Fitted rational functions are represented by thick lines. The results of numerical evaluation of (4.34) with integrals up to 3rd, 5th and 8th order are also shown.

4.11 Summary and conclusions

The present work has developed the concept of balanced fairness in two directions. First, we have presented ways in which flow-level performance analysis can be carried out in wireless networks. Second, we have introduced several computational methods, which allow larger systems to be analyzed. These include the following:

1. By suitably combining solving of the balance function recursion and the related LP problem, we efficiently minimize the per-state computations in numerical evaluation of the balance function. This method is feasible for all systems in which the link capacity set is given in terms of the instantaneous rate vectors.

2. By utilizing the values and derivatives of flow throughput at zero load and at the capacity limit, we can approximately evaluate almost any system along a given load line.

3. The introduction of continuous class indices makes possible the inclusion of, e.g., spatial characteristics of flows in the model.

4. If the link graph representing interference is a perfect graph, we may enumerate the cliques of the link graph to obtain the basic polytope form of the balance function recursion. This greatly facilitates the

computation of the recursion. Although the clique constraints are not sufficient for all systems, the constraints may be utilized as an approximation giving an upper bound for the performance.

Balanced fairness has emerged as an efficient performance analysis tool for communications networks. Whereas the existing work on balanced fairness has mainly considered fixed networks and solutions in simple examples, the focus of the work presented in this chapter has been especially on the large-scale application of BF. We have participated in the development of the concept by extending the analysis outside the wireline network domain and addressed the problems that arise when the system is no longer amenable to a closed-form solution.

We have provided a general computational framework for analysis of any wireless multihop network under balanced fairness resource sharing. Despite the unavoidable state space explosion, we have significantly expanded the reach of numerical performance analysis, both in problem size and achievable accuracy. Further development of computational schemes in the context, along with many problems related to routing and scheduling, depends on evolution of wireless network models. The capacity set of a given wireless system may be difficult to determine in the inequality form. On the other hand, the number of different instantaneous rate vectors is typically huge, which makes also the LP formulation difficult. The question is whether one can devise a model that adequately captures the effects of radio interference, but for which the capacity set can be determined without excessive computational effort.

The proposed approach of interpolating the throughput performance is an important step in analyzing the flow throughput in systems of arbitrary size. Along with the improved bounds for BF presented in [Bon06] the approximation scheme represents the current state-of-the-art flow-level analysis methods for general networks. In this context we have identified three tasks for further research. The capacity limit derivative remains to be derived for systems with a smooth capacity set (with $N > 2$) and for systems with continuous class indices. The third, and theoretically most important, task is to provide a formal proof of the conjecture of the asymptotic, heavy load behaviour being defined by the store-and-forward bound. This is closely related to the plausible conjecture on asymptotic equivalence of balanced fairness, proportional fairness and store-and-forward.

5 AUTHOR'S CONTRIBUTION

Publication 1 This paper is independent work of the author.

Publication 2 This paper is independent work of the author.

Publication 3 This paper is written by the present author. Algorithm implementation and the numerical studies are work of the present author. The idea of the algorithm is by prof. Jorma Virtamo.

Publication 4 The ideas and results in this paper are by the present author. The paper is written jointly by all the authors.

Publication 5 This paper is a joint work of the authors. The idea of extending the balanced fairness concept from fixed networks to a more general constraint model is by prof. Jorma Virtamo.

Publication 6 This paper is a joint work of the authors.

Publication 7 This paper is a joint work of the authors. The numerical studies in Section IV B were conducted by M.Sc. (Tech.) Juha Leino. The idea of the value extrapolation method is by prof. Jorma Virtamo.

Publication 8 This paper is a joint work of the authors. The present author contributed especially on the continuous class index model and participated actively in the writing process. The ideas related to the approximation scheme are partially results from the close co-operation of all the authors and partially by the other authors. Section 4.1 was written by Dr. Thomas Bonald and Appendix A by prof. Jorma Virtamo. The rest of the paper was jointly drafted by the present author and prof. Jorma Virtamo with Dr. Thomas Bonald significantly contributing to improvement of the presentation.

REFERENCES

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

[Ari84] Erdal Arikan. Some complexity results about packet radio networks. *IEEE Transactions on Information Theory*, IT-30(4):681–685, July 1984.

[ASSC02] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–104, August 2002.

[AW05] Ian F. Akyildiz and Xudong Wang. A survey on wireless mesh networks. *IEEE Communications Magazine*, 43(9):23–30, September 2005.

[AWF02] Khaled M. Alzoubi, Peng-Jun Wan, and Ophir Frieder. Distributed heuristics for connected dominating sets in wireless ad hoc networks. *Journal of Communications and Networks*, 4(1), March 2002.

[BBP05] Thomas Bonald, Sem Borst, and Alexandre Proutière. Intercell scheduling in wireless data networks. In *Proceedings of European Wireless*, April 2005.

[BG92] Dimitri Bertsekas and Robert Gallager. *Data Networks*. Prentice Hall, 2nd edition, 1992.

[BK73] Coen Bron and Joep Kerbosch. Algorithm 457: Finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575–578, September 1973.

[BM01] Thomas Bonald and Laurent Massoulié. Impact of fairness on internet performance. In *SIGMETRICS '01: Proceedings of the 2001 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 82–91, June 2001.

[BMPV06] Thomas Bonald, Laurent Massoulié, Alexandre Proutière, and Jorma Virtamo. A queueing analysis of max-min fairness, proportional fairness and balanced fairness. *Queuing Systems*, 53:65–84, 2006.

[Bon06] Thomas Bonald. Throughput performance in networks with linear capacity constraints. In *Proceedings of CISS 2006*, March 2006.

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

[BP03] Thomas Bonald and Alexandre Proutière. Insensitive bandwidth sharing in data networks. *Queuing Systems*, 44:69–100, 2003.

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

[BPRV03] Thomas Bonald, Alexandre Proutière, James Roberts, and Jorma Virtamo. Computational aspects of balanced fairness. In Charzinski and et al., editors, *Providing Quality of Service in Heterogeneous Environments (Proc. of ITC-18)*, pages 801–810. Elsevier Science, 2003.

[BSS93] Mokhtar Bazaraa, Hanif Sherali, and C.M. Shetty. *Nonlinear programming; theory and algorithms*. Wiley, 1993.

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

[BV05] Thomas Bonald and Jorma Virtamo. A recursive formula for multirate systems with elastic traffic. *IEEE Communications Letters*, 9:753–755, 2005.

[BVY03] Patrik Björklund, Peter Värbrand, and Di Yuan. Resource optimization of spatial TDMA in ad hoc radio networks: a column generation approach. In *Proceedings of INFOCOM 2003*, pages 818–824, April 2003.

[ČHE02] Mario Čagalj, Jean-Pierre Hubaux, and Christian Enz. Minimum-energy broadcast in all-wireless networks: NP-completeness and distribution issues. In *Proceedings of MOBICOM'02*, pages 172–182, 2002.

[CL87] Imrich Chlamtac and Anat Lerner. Fair algorithms for maximal link activation in multi-hop radio networks. *IEEE Transactions on Communications*, 35(7):739–749, July 1987.

[CN02] Kai Chen and Klara Nahrstedt. Efficient location-guided tree construction algorithms for small group multicast in MANET. In *Proc. of INFOCOM 2002*, pages 1180–1189, June 2002.

[CSS03] Julien Cartigny, David Simplot, and Ivan Stojmenovic. Localized minimum-energy broadcasting in ad-hoc networks. In *Proceedings of IEEE INFOCOM 2003*, pages 2210–2217, April 2003.

[CT00a] Jae-Hwan Chang and Leandros Tassiulas. Energy conserving routing in wireless ad-hoc networks. In *INFOCOM 2000*, volume 1, pages 22–31, 2000.

[CT00b] Jae-Hwan Chang and Leandros Tassiulas. Fast approximate algorithms for maximum lifetime routing in wireless ad-hoc networks. In *Lecture Notes in Computer Science; Networking 2000*, pages 702–713, May 2000.

[DH03] Doratha Drake and Stefan Hougardy. A simple approximation algorithm for the weighted matching problem. *Information Processing Letters*, 85(4):211–213, February 2003.

[Die00] Reinhard Diestel. *Graph Theory*. Springer-Verlag, 2000. Electronic Edition 2000, `http://www.math.uni-hamburg.de/home/diestel/books/graph.theory/`.

[Dzi97] Zbigniew Dziong. *ATM Network Resource Management*. McGraw-Hill, 1997.

[EG01] Ömer Egecioglu and Teofilo F. Gonzalez. Minimum-energy broadcast in simple graphs with limited node power. In *Proceedings of IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS 2001)*, pages 334–338, August 2001.

[FBP+01] Slim Ben Fredj, Thomas Bonald, Alexandre Proutière, Gwénaël Régnié, and James W. Roberts. Statistical bandwidth sharing: a study of congestion at flow level. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 111–122, August 2001.

[FdLFL+01] Guy Fayolle, Arnaud de La Fortelle, Jean-Marc Lasgouttes, Laurent Massoulié, and James Roberts. Best-effort networks: modeling and performance analysis via large networks asymptotics. In *Proceedings of IEEE INFOCOM 2001*, pages 709–716, April 2001.

[Gab90] Harold N. Gabow. Data structures for weighted matching and nearest common ancestors with linking. In *SODA '90: Proceedings of the first annual ACM-SIAM symposium on Discrete algorithms*, pages 434–443, January 1990.

[GK98] Sudipto Guha and Samir Khuller. Approximation algorithms for connected dominating sets. *Algorithmica*, 20(4):374–387, 1998.

[GK99] Sudipto Guha and Samir Khuller. Improved methods for approximating node weighted steiner trees and connected dominating sets. *Information and computation*, 150:57–74, 1999.

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

[GM03] Chao Gui and Prasant Mohapatra. Efficient overlay multicast for mobile ad hoc networks. In *Proc. of IEEE Wireless Communications and Networking 2003*, pages 1118–1123, March 2003.

[GM04] Chao Gui and Prasant Mohapatra. Scalable multicasting in mobile ad hoc networks. In *Proceedings of INFOCOM 2004*, pages 2119–2129, June 2004.

[GPM03] Rajiv Gandhi, Srinivasan Parthasarathy, and Arunesh Mishra. Minimizing broadcast latency and redundancy in ad hoc networks. In *Proc. of MobiHoc'03*, pages 222–232, June 2003. Corrected version: `http://crab.rutgers.edu/~rajivg/pubs.html`.

[HB01] Xiao Long Huang and Brahim Bensaou. On max-min fairness and scheduling in wireless ad hoc networks: analytical framework and implementation. In *Proc. ACM MobiHOC 2001*, pages 221–231, October 2001.

[HDL+02] Zygmunt J. Haas, Jing Deng, Ben Liang, Panagiotis Papadimitatos, and S. Sajama. Wireless ad hoc networks. In John Proakis, editor, *Encyclopedia of Telecommunications*. John Wiley, December 2002.

[HS88] Bruce Hajek and Galen Sasaki. Link scheduling in polynomial time. *IEEE Transactions on Information Theory*, 34(5):910–917, September 1988.

[IET] IETF. Mobile ad-hoc networks - charter. `http://www.ietf.org/html.charters/manet-charter.html`.

[JPPQ05] Kamal Jain, Jitendra Padhye, Venkata N. Padmanabhan, and Lili Qiu. Impact of interference on multi-hop wireless network performance. *Wireless Networks*, 11:471–487, 2005.

[JX04] Mikael Johansson and Lin Xiao. Scheduling, routing and power allocation for fairness in wireless networks. In *Proc. IEEE Vehicular Technology Conference (VTC 2004-Spring)*, pages 1355–1360, May 2004.

[Kel79] Frank P. Kelly. *Reversibility and stochastic networks*. Wiley, 1979.

[Kel97] Frank P. Kelly. Charging and rate control for elastic traffic. `http://www.statslab.cam.ac.uk/~frank/elastic.html`, 1997. Corrected version.

[KGV83] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220(4589):671–680, 1983.

[KR95] Philip Klein and R. Ravi. A nearly best-possible approximation algorithm for node-weighted steiner trees. *J. Algorithms*, 19(1):104–115, 1995.

[LA03] Rangsan Leelahakriengkrai and Rajeev Agrawal. Scheduling in multimedia CDMA wireless networks. *IEEE Transactions on Vehicular Technology*, 52(1):226–239, January 2003.

[LK00] Hyojun Lim and Chongkwon Kim. Multicast tree construction and flooding in wireless ad hoc networks. In *Proc. of MSWIM 2000*, pages 61–68, August 2000.

[LLB00] Haiyun Luo, Songwu Lu, and Vaduvur Bharghavan. A new model for packet scheduling in multihop wireless networks. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 76–86, August 2000.

[LV05] Juha Leino and Jorma Virtamo. Insensitive traffic splitting in data networks. In *Proceedings of 19th International Teletraffic Congress (ITC-19)*, pages 1355–1364, 2005.

[Mas05] Laurent Massoulié. Structural property of proportional fairness: stability and insensitivity. *Annals of applied probability*, 2005. Submitted.

[MAW96] Nick McKeown, Venkath Anantharam, and Jean Warland. Achieving 100% throughput in an input-queued switch. In *Proceedings of IEEE INFOCOM*, pages 296–302, March 1996.

[McK99] Nick McKeown. The iSLIP scheduling algorithm for input-queued switches. *IEEE/ACM Trans. Netw.*, 7(2):188–201, April 1999.

[MR00] Laurent Massoulié and James W. Roberts. Bandwidth sharing and admission control for elastic traffic. *Telecommunication Systems*, 15:185–201, March 2000.

[MR02] Laurent Massoulié and James Roberts. Bandwidth sharing: Objectives and algorithms. *IEEE/ACM Transactions on Networking*, 10(3):320–328, June 2002.

[MW00] Jeonghoon Mo and Jean Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking*, 8(5):556–567, October 2000.

[Nav05] Alfredo Navarra. Tighter bounds for the minimum energy broadcasting problem. In *Proceedings of WIOPT 2005*, pages 313–322, April 2005.

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

[NKGB00] Thyagarajan Nandagopal, Tae-Eun Kim, Xia Gao, and Vaduvur Bharghavan. Achieving MAC layer fairness in wireless packet networks. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 87–98, August 2000.

[Pre99] Robert Preis. Linear time $\frac{1}{2}$-approximation algorithm for maximum weighted matching in general graphs. In *Proceedings of Symposium on Theoretical Aspects of Computer Science (STACS 99)*, pages 259–269, 1999.

[Pri57] R.C. Prim. Shortest connection networks and some generalizations. *Bell Syst. Tech. Journ.*, 36:1389–1401, 1957.

[Ram96] S. Ramanathan. Multicast tree generation in networks with asymmetric links. *IEEE/ACM Transactions on Networking*, 4(4):558–568, August 1996.

[Sal04] Theodoros Salonidis. *Distributed topology organization and transmission scheduling in wireless ad-hoc networks*. PhD thesis, University of Maryland, College Park, August 2004.

[Sch79] P. Schweitzer. Approximate analysis of multiclass closed networks of queues. In *Proc. of International Conference on Stochastic Control and Optimization*, Amsterdam, 1979.

[Ser99] Richard Serfozo. *Introduction to Stochastic Networks*. Springer, 1999.

[Sha49] Claude E. Shannon. Communication in the presence of noise. *Proceedings of Institute of Radio Engineers*, 37(1):10–21, January 1949.

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

[Som02] Oscar Somarriba. On constraint power control for spatial TDMA in multi-hop ad hoc radio networks. In *Proceedings of the 2nd Swedish Workshop on Wireless Ad-hoc Networks*, Stockholm, Johannesbergs Slott, March 2002.

[SSZ02] Ivan Stojmenovic, Mahtab Seddigh, and Jovisa Zunic. Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks. *IEEE Transactions on parallel and distributed systems*, 13(1):14–25, January 2002.

[Sta05] William Stallings. *Wireless Communications and Networks*. Pearson Prentice Hall, 2nd edition, 2005.

[Tah97] Hamdy A. Taha. *Operations research, an introduction*. Prentice Hall, 6th edition, 1997.

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

[Tim03] Vesa Timonen. Simulation studies on performance of balanced fairness. Master's thesis, Helsinki University of Technology, 2003. http://www.netlab.hut.fi/tutkimus/fit/publ/thesis_Timonen.pdf.

[TS02] Leandros Tassiulas and Saswati Sarkar. Maxmin fair scheduling in wireless networks. In *Proc. IEEE INFOCOM 2002*, pages 764–772, June 2002.

[TT03] Vahid Tabatabaee and Leandros Tassiulas. MNCM a new class of efficient scheduling algorithms for input-buffered switches with no speedup. In *Proceedings of IEEE INFOCOM*, pages 1406–1413, March 2003.

[Vae01] Moises Navarro Vaello. Optimal routing of multicast traffic in ad hoc networks. M.Sc. thesis, Helsinki University of Technology, 2001.

[WaLF01] P.-J. Wan, G. Călinescu, X.-Y. Li, and O. Frieder. Minimum-energy broadcast routing in static ad hoc wireless networks. In *In Proceeding of INFOCOM 2001*, pages 1162–1171, April 2001.

[WC02] Brad Williams and Tracy Camp. Comparison of broadcasting techniques for mobile ad hoc networks. In *Proc. of MobiHoc'02*, pages 194–205, June 2002.

[WK04] Xin Wang and Koushik Kar. Distributed algorithms for maxmin fair rate allocation in ALOHA networks. In *Proceedings of the Allerton Conference*, October 2004.

[WNE99] Jeffrey E. Wieselthier, Gam D. Nguyen, and Anthony Ephremides. Algorithms for energy-efficient multicasting in ad hoc wireless networks. In *Proceedings of IEEE MILCOM 1999*, pages 1414–1418, 1999.

[WNE00] Jeffrey E. Wieselthier, Gam D. Nguyen, and Anthony Ephremides. On the construction of energy-efficient broadcast and multicast trees in wireless networks. In *Proceedings of IEEE INFOCOM 2000*, pages 585–594, 2000.

[WNE01] Jeffrey E. Wieselthier, Gam D. Nguyen, and Anthony Ephremides. Algorithms for energy-efficient multicasting in static ad hoc wireless networks. *ACM Mobile Networks and Applications*, 6:251–263, 2001.

[WP] Wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/Wireless_networks.

[WW03] Mirjam Wattenhofer and Roger Wattenhofer. Distributed weighted matching, 2003. Technical Report 420, ETH Zurich, Department of Computer Science, 2003.

[Yua05] Di Yuan. Computing optimal or near-optimal trees for minimum-energy in wireless networks. In *Proceedings of WIOPT 2005*, pages 323–331, April 2005.