Imperial College London

Non-convex Resource Allocation in Communication Networks

by

Georgios Tychogiorgos

Submitted in fulfilment of the requirements for the degree of
Doctor of Philosophy in Electrical and Electronic Engineering Department
of Imperial College London
and the Diploma of Imperial College London

Declaration

I herewith certify that all material in this dissertation which is not my own work has been properly acknowledged.

Georgios Tychogiorgos

Abstract

The continuously growing number of applications competing for resources in current communication networks highlights the necessity for efficient resource allocation mechanisms to maximize user satisfaction. Optimization Theory can provide the necessary tools to develop such mechanisms that will allocate network resources optimally and fairly among users. However, the resource allocation problem in current networks has characteristics that turn the respective optimization problem into a non-convex one. First, current networks very often consist of a number of wireless links, whose capacity is not constant but follows Shannon capacity formula, which is a non-convex function. Second, the majority of the traffic in current networks is generated by multimedia applications, which are non-concave functions of rate. Third, current resource allocation methods follow the (bandwidth) proportional fairness policy, which when applied to networks shared by both concave and non-concave utilities leads to unfair resource allocations. These characteristics make current convex optimization frameworks inefficient in several aspects. This work aims to develop a non-convex optimization framework that will be able to allocate resources efficiently for non-convex resource allocation formulations. Towards this goal, a necessary and sufficient condition for the convergence of any primal-dual optimization algorithm to the optimal solution is proven. The wide applicability of this condition makes this

a fundamental contribution for Optimization Theory in general. A number of optimization formulations are proposed, cases where this condition is not met are analysed and efficient alternative heuristics are provided to handle these cases. Furthermore, a novel multi-sigmoidal utility shape is proposed to model user satisfaction for multi-tiered multimedia applications more accurately. The advantages of such non-convex utilities and their effect in the optimization process are thoroughly examined. Alternative allocation policies are also investigated with respect to their ability to allocate resources fairly and deal with the non-convexity of the resource allocation problem. Specifically, the advantages of using Utility Proportional Fairness as an allocation policy are examined with respect to the development of distributed algorithms, their convergence to the optimal solution and their ability to adapt to the Quality of Service requirements of each application.

Contents

1. Introduction 15

1.1. Optimizing Communication Networks 15

1.2. Optimization Theory Overview 18

1.3. Literature Review 28

1.3.1. The Network Utility Maximization Framework 28

1.3.2. TCP as an Application of NUM 31

1.3.3. Shortcomings of NUM 37

1.3.4. Extending NUM for Current Communication Networks 40

1.3.5. The Notion of Fairness in NUM 53

1.4. Motivation and Contributions 58

1.5. Thesis Organization 61

2. A Non-convex Distributed Optimization Framework and its Application to Wireless Ad-hoc Networks 63

2.1. Introduction 63

2.2. TCP in Current Communication Networks 66

2.3. An Optimization Framework for Non-convex Problems 70

2.4. Resource Allocation in Wireless Ad-hoc Networks 76

2.4.1. Problem Formulation 76

2.4.2. Distributed Algorithm 78

2.4.3. Convergence and Oscillation Resolving Heuristic . . . 82
2.5. Numerical Results . . . 86
2.6. Concluding Remarks . . . 89

3. Non-convex Resource Allocation for Multi-tiered Multimedia Applications 92
3.1. Introduction . . . 92
3.2. Network Utility Maximization with Multi-sigmoidal Utilities . 97
3.2.1. Properties of a Multi-sigmoidal Utility Function . . . 97
3.2.2. Network Resource Allocation with Multi-sigmoidal Utilities . . . 98
3.3. The Price-based Rate Allocation Function . . . 102
3.3.1. Calculation . . . 102
3.3.2. Discontinuity . . . 103
3.3.3. Oscillations . . . 115
3.4. Resolving User Oscillations . . . 117
3.5. A Novel Multi-sigmoidal Function and its Application to NUM 120
3.5.1. A Hyperbolic Tangent Based Utility Function . . . 120
3.5.2. Approximation of the Optimal Rate . . . 123
3.6. Simulation Results . . . 130
3.6.1. Single bottleneck link . . . 131
3.6.2. Multiple bottleneck links . . . 134
3.7. Concluding Remarks . . . 136

4. Utility-Proportional Fairness for Multimedia Applications in Wireless Networks 138
4.1. Introduction . . . 138

4.2. Problem Formulation 143
4.2.1. Network Model 143
4.2.2. Optimization Problem 145
4.3. The Price-based Rate Allocation Function 147
4.3.1. Calculating a general rate equation 148
4.3.2. Application specific forms 149
4.3.3. Distributed Algorithm 154
4.4. Numerical Results 155
4.4.1. Convergence Comparison of Iterative and Analytical Solution Methods 156
4.4.2. Comparison of Bandwidth and Utility Proportional Fairness Methods 158
4.5. Concluding Remarks 168

5. Conclusion and Future Work 169
5.1. Summary of the Results 169
5.2. Future Work 172

A. Appendix 175
A.1. Proof of Optimal Rate Allocation Function for HTTP Application 175
A.2. Proof of Optimal Rate Allocation Function for FTP Application 176
A.3. Proof of Optimal Rate Allocation Function for Single-tiered Video Application 176

List of Figures

1.1. Examples of functions with respect to convexity 19
1.2. Utility Functions of Widely Used Application Types 42

2.1. Example of a Single-bottleneck Network 67
2.2. Example of utility functions 68
2.3. Improvement of the Optimization Algorithm over TCP 70
2.4. Example of the rate allocation function $r_i\left(\lambda^i\right)$ for sigmoidal utilities 83
2.5. Example Network Topology 87
2.6. Convergence of Rate Allocation 88
2.7. Convergence of Objective Function 89
2.8. Convergence of Transmission Power Allocation 90

3.1. Example of Multi-tiered Utility Functions 94
3.2. Example of a multi-sigmoidal utility with four discontinuity points 105
3.3. Example of a multi-sigmoidal utility derivative and its four hyperbolic secant components 127
3.4. Example of a network topology with a single bottleneck link . 131
3.5. Convergence of rate without oscillation 132
3.6. Convergence of rate allocation when oscillation occurs 133

3.7. Convergence of the objective function with oscillation 134
3.8. Example of a network topology with multiple bottleneck links 135
3.9. Convergence of rate allocation with oscillation 136
3.10. Convergence of the objective function with oscillation 137

4.1. The feasible rate region of a sigmoidal utility function 139
4.2. Simple Network Topology Example 156
4.3. Convergence Speed Comparison of Iterative and Analytical Solution Rate Allocation Methods 159
4.4. Network Topology Example 160
4.5. Convergence of Utility and Objective Functions 161
4.6. Convergence of Rate Allocation 162
4.7. Convergence of Transmission Power Allocation 163
4.8. Wireless Network Topology Example 164
4.9. Convergence of User Utility Functions 165
4.10. Convergence of Rate Allocation 166
4.11. Convergence of Transmission Power Allocation 167

List of Tables

1.1. TCP version and User Utility Function 37
1.2. Application Types and the Respective User Utility Functions 41

4.1. The Optimal Resource Allocation Function for Widely Used Types of Applications 150
4.2. Tangent Components and the Respective Utility and Aggregate Price Value Regions for a Utility with 4 sigmoidal components 152
4.3. The Utility functions of Some Widely Used Types of Applications 157

Acknowledgements

It is true that pursuing a PhD is something like a marathon. No matter how hard you train, you are never going to finish unless you have the most appropriate people around you to guide you and support you. During my marathon, I had been lucky enough to have interacted, in one or another way, with several wonderful people that I need to thank explicitly.

At first, I would like to express my gratitude to my supervisor, Professor Kin K. Leung. He strongly encouraged me to pursue a PhD and did everything within his powers to overcome any difficulties and make this happen. I feel grateful also for all his support and guidance during my PhD. He was not only a good supervisor for my PhD but also an inspiration for my future career.

During my PhD, I had the opportunity to collaborate with a number of people and excel both my research and personal skills. Therefore, I would like to thank Dr. Athanasios Gkelias for working closely with me during the last two years of my PhD. I admire the breadth of his technical knowledge and really enjoyed our non-technical conversations on careers and entrepreneurship. I would also like to express my gratitude to Dr. Chatschik Bisdikian (IBM Research, USA). I feel honoured to have worked with him at IBM Research. Even though our research is not part of this thesis, he has been a true mentor for me and affected implicitly the quality of the work

presented in this thesis.

Of course, I owe everything to my family, my parents Christos and Alexandra and my sister Daphne. They are responsible not only for my studies but also for the person I have become. They have taught me that with hard work, integrity and persistence I can achieve any goal. Moreover, I need to thank all my friends in the UK and Greece, who have provided me with their support during my PhD. Things would be even tougher without them.

Last but not least, I would like to thank Maria Dourou for all she has done for me. She not only supported and encouraged me in the difficult moments of this marathon but she made this her marathon as well by running beside me.

List of Publications

PhD Related Publications

IEEE Conferences

1. G. Tychogiorgos, A. Gkelias and K.K. Leung, "Utility-Proportional Fairness in Wireless Networks", Proc. IEEE PIMRC 2012, Sydney, Australia, September 2012 **(Best Student Paper Award)**

2. G. Tychogiorgos, A. Gkelias and K.K. Leung, "A New Distributed Optimization Framework for Hybrid Ad-hoc Networks", Proc. IEEE GLOBECOM - Workshop on Heterogeneous, Multi-Hop, Wireless and Mobile Networks, Houston, TX, USA, December 2011

3. G. Tychogiorgos, A. Gkelias and K.K. Leung, "Towards a Fair Non-convex Resource Allocation in Wireless Networks", Proc. IEEE PIMRC, Toronto, Canada, September 2011

Other Publications

1. G. Tychogiorgos and C. Bisdikian, "A Framework for Managing the Selection of Spatiotemporally Relevant Information Providers", IFIP/IEEE IM, Ghent, Belgium, May 2013

2. G. Tychogiorgos and C. Bisdikian, "A Framework for Managing the Selection of Spatiotemporally Relevant Information Providers", Annual International Technology Alliance (ITA) Conference, Southampton, UK, September 2012

3. G. Tychogiorgos and C. Bisdikian, "Seeking Providers of Spatially Relevant Sensory Information", Annual International Technology Alliance (ITA) Conference, Maryland, USA, September 2011

4. G. Tychogiorgos and C. Bisdikian, "Selecting Relevant Sensor Providers for Meeting 'Your' Quality Information Needs", Proc. IEEE Conference on Mobile Data Management (MDM), Lulea, Sweden, June 2011

5. G. Tychogiorgos and C. Bisdikian, "Selecting Relevant Sensor Providers for Meeting Your Quality of Information Needs", IBM Res. Rep. RC25116, December, 2010

6. G. Tichogiorgos, K.K. Leung, A. Misra and T. LaPorta, "Distributed Network Utility Optimization in Wireless Sensor Networks Using Power Control", Proc. IEEE PIMRC, Cannes, France, September 2008

7. G. Tichogiorgos, K.K. Leung, A. Misra and T. LaPorta, "Distributed Network Utility Optimization in Wireless Sensor Networks Using Power Control", Annual International Technology Alliance (ITA) Conference, London, UK, September 2008

Patents

1. G. Tychogiorgos and C. Bisdikian, "Characterizing and Selecting Providers of Relevant Information Based on Quality of Information Metrics", Submitted to the US Patent Office, 2011

1. Introduction

1.1. Optimizing Communication Networks

Since the creation of ARPANET [1], the first packet switching communication network in 1969 that connected university laboratories, industrial and government research centers in the US, there has been a tremendous change in the extend and characteristics of communication networks, and especially the *internet*, the amount of data that is shared through them and the variety of applications that generate this traffic.

The initial ARPANET implementation involved the connection of four computers using wired links which gradually were increased to a few hundreds, while a satellite link was also utilized. Current communication networks however, are consisted of many interconnected sub-networks that consist of both wired and wireless links and must be able to communicate with each other despite any incompatibilities. The *OSI Reference Model* [2] and particularity the Transport and Network layers assisted to overcome these incompatibilities and allow the communication between heterogeneous networks. Moreover, the development of cellular networks since 1990 and their continuous growth to support more users and provide more applications to their users have led cellular companies to create unified high-capacity IP-based networks that consist of both wireless and wired (backbone) links.

Recent Cisco IP traffic studies [3][4] provide useful information and insights regarding the traffic characteristics in current communication networks. The total internet traffic currently exceeds 31 Exa-bytes per month and this amount is forecasted to increase to more than 40 Exa-bytes per month by 2013. On the other hand, mobile traffic has seen an explosive increase in the past years. While total mobile traffic in 2008 was no more than 33 Peta-bytes per month, mobile traffic is forecasted to reach 2.1 Exa-bytes per month by the end of 2013. The reason causing this abrupt increase in the traffic in both internet and mobile networks can be justified if one looks carefully at these statistics from another perspective; that of the applications that generate the traffic. While in 2008 the majority of the traffic was generated by "traditional" types of applications, such as web browsing, email and file sharing applications, that accounted for 77% of the total traffic in the internet, multimedia applications, such as VoIP, Video Streaming etc., dominate the traffic nowadays exceeding 57% of the total traffic in the internet. The statistics are similar in the mobile internet as well, where the video traffic alone will account for two-thirds of the total mobile traffic by 2013.

This abrupt increase of the total traffic highlights the necessity for more efficient methods of sharing the available bandwidth so that users are receiving the maximum possible satisfaction and the best possible experience when using a communication network. In addition, taking into account that users are being charged by the network providers for access, the more efficient the resource allocation is, the more satisfied the users will be and consequently the more willing to continue paying the provider for the service. The heterogeneity of the provided applications also shows that all traffic does not have the same resource requirements. This strengthens the

need for more sophisticated allocation methods that will be able to distinguish between different types of applications and try to allocate resources in a way that maximizes user satisfaction for each application type.

Optimization Theory can provide a powerful tool towards the development of such methods for various reasons. Optimization Theory has been used successfully in many areas related to communication networks, such as optimal routing, power control etc., but also in other applications, such as chemical engineering [5], fleet management and inventory organization, since it leads to the best possible solutions for a given problem. In addition, there are techniques, such as the *Langrangian Method* that can lead to the development of distributed algorithms. Distributed calculation of the optimal solution is of significant importance in communication networks, which consist of numerous network nodes and traffic sources that behave independently and selfishly to achieve the maximum possible level of satisfaction using the resources of the network. Moreover, optimization theory can also be used to assure that the allocation of resources to each application will follow its *Quality of Service* requirements and satisfy some notion of fairness. This can be achieved by the appropriate formulation of the optimization problem and the use of specific allocation policies according to the desired type of fairness.

Optimization Theory has been also used in the past to optimize the resource allocation in communication networks but there are very important research challenges that are yet to be answered. The work described in this thesis attempts to answer some of these open research questions that relate to both the fundamentals of Optimization Theory itself and the practical considerations that one must make in order to design efficient resource allocation protocols for current communication networks. To this purpose,

the focus of the remaining of this Chapter will be threefold. First, a brief overview of *Optimization Theory* will be provided in order to help the readers familiarize themselves with the necessary optimization tools and methods that will be used in the remaining of this thesis to support our work. Then, a literature review of the resource allocation research area will be provided. This will allow us to describe the foundations on which our research has been based upon, highlight the motivation behind our work and the contribution of our research, which will be presented in the remainder of this Thesis.

1.2. Optimization Theory Overview

This section provides a brief description of the basic notions in *Optimization Theory*, based on textbooks [6] and [7]. The main focus of this overview is on the areas of function and problem *convexity*, optimization problem formulation, as well as on the advantages that distributed optimization techniques can offer to solve such problems. The interested reader is referred to the aforementioned textbooks for a complete presentation and analysis of *Optimization Theory*.

A set C is a *convex* set if it is a subset of $\Re^n$ and if $\alpha x + (1-\alpha)\, y \in C$, $\forall\ x, y \in C$ and $\forall\ \alpha \in [0,1]$. In accordance, a function $f : C \to \Re$, where C is a convex subset of $\Re^n$, is *convex* if

$$f\left(\alpha x + (1-\alpha)\, y\right) \leq \alpha f\left(x\right) + (1-\alpha)\, f\left(y\right), \quad \forall x, y \in C, \forall \alpha \in [0,1]\,. \tag{1.1}$$

An intuitive interpretation of (1.1) is that the line segment between $(x, f\left(x\right))$ and $(y, f\left(y\right))$, which is the chord from x to y, lies always above the graph

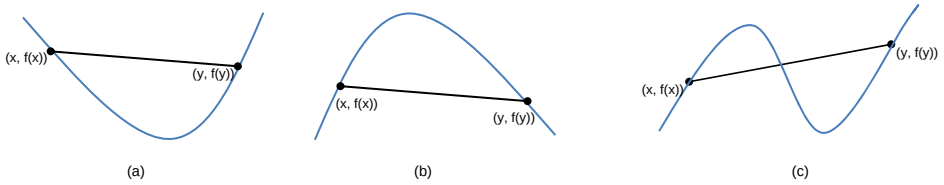

Figure 1.1.: Examples of functions with respect to convexity

of f. On the other hand, a function f is called *concave* if

$$f\left(\alpha x + (1-\alpha)\, y\right) \geq \alpha f\left(x\right) + (1-\alpha)\, f\left(y\right), \quad \forall x, y \in C, \forall \alpha \in [0,1]\,. \tag{1.2}$$

Relating *convex* and *concave* functions one can comment that if f is a *convex* function, then $-f$ is *concave* and vice versa. In addition, function f is called *strictly convex* if inequality (1.1) is strict for all $x, y \in C$ with $x \neq y$, and for all $\alpha \in (0,1)$ and equivalently there is a *strictly concave* function if inequality (1.2) is strict. A function can be neither *convex* nor *concave*. Figure 1.1 shows examples of a convex function, a concave function and a function that is neither convex nor concave. In general, *convex* functions are convenient for minimization problems since their local minima are also global and equivalently *concave* functions are convenient for maximization problems. However, functions whose convexity properties change, such as the example in Figure 1.1c, is generally difficult to optimize since it can have many local optima.

To represent an optimization problem, we use the notation

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{subject to} && h_i(x) \leq 0, \quad i = 1, \ldots, m \\ &&& f_i(x) = 0, \quad i = 1, \ldots, p \end{aligned} \tag{1.3}$$

in order to describe the problem of finding the value of the *optimization variable* x that will minimize the *objective function* $f(x)$ among all possible values of the variable so that the conditions $h_i(x) \leq 0,\ i = 1, \ldots, m$ and $f_i(x) = 0,\ i = 1, \ldots, p$ are all satisfied. A general problem formulation such as the one presented in (1.3) can have a number of locally and globally optimal solutions. *Global Optimization* [8][9] is the area of Optimization Theory interested in calculating the globally optimal solutions of an optimization problem that will minimize the value of the objective function $f(x)$ within the feasible region. In addition, research in Global Optimization is also interested in the feasibility characterization of optimization problems and in determining upper and lower bounds of their objective functions [10]. *Convex Optimization* is a specific area of Global Optimization where locally optimal solutions are also globally optimal. An optimization problem with such property is called a Convex Optimization problem. More specifically, an optimization problem such as (1.3) is called *convex* if the following conditions hold:

- the objective function $f(x)$ is a convex function of the optimization variable x,
- the inequality constraint functions $h_i(x)$ are convex, and
- the equality constraint functions $f_i(x)$ are affine.

In order to solve an optimization problem, such as the one described in (1.3), we can use *Duality Theory*. According to this, problem (1.3) is called the *primal* problem and we need to create the so-called *dual* problem and consequently solve both, *primal* and *dual*, problems at the same time in a distributed way. In order to create the *dual problem*, we first need to define

the *Lagrangian* function. This is a function $L : R \times R^m \times R^p \to R$ given by:

$$L(x, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(x) + \sum_{i=0}^{m} \lambda_i h_i(x) + \sum_{i=0}^{p} \mu_i f_i(x). \tag{1.4}$$

The idea behind the *Lagrangian* function is to take the constraints into account by augmenting the objective function with a weighted sum of the constraint functions. These weights are called *Lagrange multipliers* and we refer to λ_i as the *Lagrange multiplier* associated with the i^{th} inequality constraint $h_i(x) \leq 0$ and to μ_i as the *Lagrange multiplier* associated with the i^{th} equality constraint $f_i(x) = 0$. In addition, the vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are called the *dual variables* or *Lagrange multiplier vectors* of problem (1.3)[1]. Then, we define the *Lagrange dual function* $D : R^m \times R^p \to R$ as the minimum value of the Lagrangian over x or otherwise

$$D(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \inf_x L(x, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \inf_x \left(f(x) + \sum_{i=0}^{m} \lambda_i h_i(x) + \sum_{i=0}^{p} \mu_i f_i(x) \right), \tag{1.5}$$

where inf is *the greatest lower bound*[6] and is used to handle the case where the Lagrangian is unbounded below in x and hence the minimum cannot be calculated. Note that the *dual* optimization problem is a maximization problem. If the primal problem had been a maximization problem, then the dual would have been a minimization problem. Moreover, the dual problem is always convex even when the primal is not. This is because the dual function is the point-wise infimum of a family of affine functions of $\boldsymbol{\lambda}$ and

[1] In network optimization problems, the dual variables often represent link prices and therefore the vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are also called price vectors

$\boldsymbol{\mu}$. The *Lagrange dual problem*[2] associated with problem (1.3) is defined as

$$\begin{aligned} &\text{maximize} && D\left(\boldsymbol{\lambda}, \boldsymbol{\mu}\right) \\ &\text{subject to} && \boldsymbol{\lambda} \geq 0. \end{aligned} \tag{1.6}$$

Due to the sign of the inequality constraints, the positiveness of the Lagrange multipliers in (1.6) is necessary in order to avoid the dual variables from going to $-\infty$ while attempting to maximize the dual objective function. A formal proof of the sign of the dual variables can be found in [7]. The *Lagrange dual problem* gives a lower bound on the optimal value of the *primal problem* and therefore we can write that the following inequality shows the relationship between the optimal values of the primal, p^*, and the dual problem, d^*,

$$d^* \leq p^*. \tag{1.7}$$

This property is called *Weak Duality* and holds for every *primal-dual* pair of problems if both d^* and p^* are finite. In case equality holds in (1.7), we say that *strong duality* holds between these two problems and the *Duality Gap*, $p^* - d^*$, is zero. In general, if problem (1.3) is convex then strong duality holds. However, as it will be shown in Chapter 2, strong duality is possible to hold even for some non-convex optimization problems.

In case strong duality holds for a problem, then the *complementary slackness* condition also holds. According to that, if the duality gap is zero, the following holds at the optimal solutions

$$\sum_{i=0}^{m} \lambda_i^* h_i\left(x^*\right) + \sum_{i=0}^{p} \mu_i^* f_i\left(x^*\right) = 0 \tag{1.8}$$

[2] or, simply, the dual problem

and since each of these terms has the same sign it follows that

$$\begin{aligned} \lambda_i^* h_i(x^*) &= 0, \quad i = 1, \ldots, m \\ \mu_i^* f_i(x^*) &= 0, \quad i = 1, \ldots, p. \end{aligned} \tag{1.9}$$

The *complementary slackness* condition for the first constraint can be also written as:

$$\lambda_i^* > 0 \Rightarrow h_i(x^*) = 0 \tag{1.10}$$

which means that the i^{th} optimal *Lagrange multiplier* is zero unless the i^{th} constraint is active at the optimum.

For any optimization problem where the objective function and the constraint functions are differentiable and strong duality holds, any pair of optimal primal and dual variables, x^* and $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$, must satisfy the *Karush-Kuhn-Tucker* (KKT) conditions. These conditions stem from the fact that the gradient of the Lagrangian function given by

$$\nabla L(x^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = \nabla f(x^*) + \sum_{i=0}^{m} \lambda_i^* \nabla h_i(x^*) + \sum_{i=0}^{p} \mu_i^* \nabla f_i(x^*) \tag{1.11}$$

must be equal to zero at the optimal points and thus

$$\begin{aligned} h_i(x^*) &\leq 0, \quad i = 1, \ldots, m \\ f_i(x^*) &= 0, \quad i = 1, \ldots, p \\ \lambda_i^* &\geq 0, \quad i = 1, \ldots, m \\ \lambda_i^* h_i(x^*) &= 0, \quad i = 1, \ldots, m \\ \nabla f(x^*) + \sum_{i=0}^{m} \lambda_i^* \nabla h_i(x^*) + \sum_{i=0}^{p} \mu_i^* \nabla f_i(x^*) &= 0, \end{aligned} \tag{1.12}$$

which are called the KKT conditions. Note that for non-convex optimization problems, these conditions are the necessary conditions for optimality,

while in the case of convex problems, they are the necessary and sufficient conditions for optimality.

Network optimization problems are in most cases consisted of independent nodes and links and therefore centralized Global Optimization algorithms [9][11] are often hard to be implemented. For this reason, distributed solutions are always preferable. In order to solve an optimization problem in a distributed way, we often use an iterative method called *Gradient Method*. This method is applicable only for differentiable objective function f and constraints. [3] At each iteration k of the algorithm the new value of the optimization variable is determined based on the current value and the gradient of Lagrangian function. Specifically, the iterative calculation is carried out using the formula

$$x^{k+1} = x^k - \alpha^k D^k \nabla L\left(x^k\right) \tag{1.13}$$

where α is the *step size* and D^k is a positive definite symmetric matrix. The term $-D^k \nabla L\left(x^k\right)$ is often referred as the direction d^k of the gradient method.

There are many variations of the gradient method that mainly differ in the choice of the step size and the direction d^k. Among others, there is the *Steepest Descent Method* and *Newton's Method*. At the former method, matrix D^k is a $n \times n$ identity matrix. This is the simplest choice and the choice of least complexity but often leads to slow convergence. In the latter method, we select $D^k = \left(\nabla^2 L\left(x^k\right)\right)^{-1}$, provided that $\nabla^2 L\left(x^k\right)$ is a positive definite matrix. The idea behind *Newton's Method* is that at each iteration the quadratic approximation of the Lagrangian function L should be optimized. *Newton's Method* is one of the fastest gradient methods but

[3] In case of non-differentiable objective functions other iterative methods can be used, such as the *Subgradient Method*, for which a complete description and analysis is presented in [7].

also of relatively high complexity since at each iteration the Hessian matrix of the objective function, and its inverse, must be calculated in order to determine the direction of movement at the next iteration.

Regarding the step size α^k of the gradient method, there are a number of options for its value. The most common of them are:

- *Constant step size*, where $\alpha^k = \alpha$ is a positive constant,
- *Constant step length*, where $\alpha^k = \frac{\alpha}{\|\nabla L(x^k)\|}$, and
- *Square summable but not summable*, which actually implies that $\alpha^k \geq 0$, $\sum_{k=1}^{\infty} \alpha_k^2 \leq 0$ and $\sum_{k=1}^{\infty} \alpha_k = \infty$.

Concerning the convergence of the gradient method for each of the choices of step size, it has been proven that the first two force the gradient method to converge to a solution very close to the actual optimal solution, as long as the step size has sufficiently small value, while for *square summable but not summable* step sizes the gradient algorithm will converge to the theoretical optimal value.

Since the gradient method is an iterative method, it is necessary to determine the stopping criteria of the algorithm. A usual stopping criterion is when the norm of the gradient becomes sufficiently close to zero, which can be written as:

$$\|\nabla L\left(x^k\right)\| \leq \epsilon. \tag{1.14}$$

Even though the exact value of ϵ is not known a priori for a solution sufficiently close to the optimal, however the distance from the optimal given the positive scalar ϵ is given by:

$$f(x) - f(x^*) \leq \frac{\epsilon^2}{m} \tag{1.15}$$

where m is positive scalar.

It is very common in network optimization applications that the optimization variables take values within a closed interval rather than $\Re$, i.e. if x is the optimization variable $x \in [x_{min}, x_{max}]$. An example of such variable that takes values within an interval would be the transmission rate of a node, which is a positive quantity and is restricted by the maximum data generation rate of the source. The *Gradient Descent Method* as described above can not force the optimization variables to stay within this range. Therefore, a variation of the gradient method, called *Gradient Projection Method*, is used instead. The idea behind this method is that as soon as the values of the optimization variable leave the feasible range, the algorithm maps its value to the closest feasible value. Formally, equation (1.13) becomes:

$$x^{k+1} = \left[x^k - \alpha^k D^k \nabla L\left(x^k\right)\right]_{x_{min}}^{x_{max}}, \tag{1.16}$$

where D^k is a diagonal matrix.

The strict convexity of $f(x)$ and the continuity of the constraint functions in problem (1.3) also implies the differentiability of the objective function of the dual problem (1.5). However, for the cases that the objective function is not strictly convex, there are methods to transform the primal objective function into a *strictly convex* function and hence to convexify the optimization problem. The *Proximal Minimization Algorithm* [12] is such an algorithm to assure that the dual objective function is differentiable. According to it, a new variable $y \in \Re^n$ is introduced and the optimization

problem (1.3) takes now the form:

$$\begin{aligned} \text{minimize} \quad & f(x) + \tfrac{1}{2c}\|x-y\|_2^2 & & \\ \text{subject to} \quad & h_i(x) \leq 0, & & i = 1, \ldots, m \\ & f_i(x) = 0, & & i = 1, \ldots, p \end{aligned} \tag{1.17}$$

where c is positive scalar parameter and $\|.\|_2$ is the Euclidean norm. Problem (1.17) is now *strictly convex* and it can be proven that its solution is the same as that of problem (1.3), i.e $x = x^*$ and $y = x^*$. The *Proximal Minimization Algorithm* is applied in various network optimization scenarios where the objective function is convex but not *strictly* convex.

Other convexification methods for specific families of optimization problems that have found significant applications to network optimization are the *Semidefinite Relaxation (SDR)* technique [13][14], which can be used to provide a convex approximation of non-convex *quadratically constrained quadratic problems (QCQPs)*, the *Sum-of-squares (SOS)* method [15] that can be used to calculate a tight bound of the optimal solution in polynomial time, and the method described in [16] that can be used to convexify optimization problems with certain monotone properties and is used mostly in reliability optimization applications.

Optimization Theory has found extensive use in network optimization applications. Moreover, it consists the corner stone of the *Network Utility Maximization* framework which has been used extensively to optimize the *resource allocation* in current communication networks and evaluate the performance of various transport layer protocols. The *resource allocation problem*, the *Network Utility Maximization (NUM)* framework and other pieces of work that deal with the optimal allocation of network resources are presented in the next section.

1.3. Literature Review

Modern communication networks must encompass and simultaneously support multiple users, services and applications with diverse demands and requirements that push networks performance closer to their limit. Therefore, optimum resource allocation between users and/or applications is of paramount importance in order to assure efficient utilization of the network. The resource allocation problem is one of the numerous research areas in which Optimization Theory has found extensive use, since it can lead to the development of distributed algorithms to assure optimal allocation of resources in a network. This section provides an overview of prior research in the area.

1.3.1. The Network Utility Maximization Framework

In 1998, Kelly et al. formulated the *Resource Allocation Problem* for wired networks in an innovative way that led to many research activities ever since. In this seminal paper [17], they introduce the notion of *Network Utility Maximization* (NUM) and formulate the resource allocation as an optimization problem for the first time. The authors assume a system consisting of fixed capacity links and a set of users that want to transmit data to a set of destination nodes. The path that the traffic follows to reach the destination nodes is known a priori and does not change during the optimization process. The resource allocation problem for the system is formulated as:

$$\begin{aligned} \max \quad & \sum_{r \in R} U_r\left(x_r\right) \\ \text{subject to} \quad & Ax \leq C \\ & x \geq 0, \end{aligned} \tag{1.18}$$

where a route r is associated with a user, the rate x_r is the allocated rate to user r and $U_r(x_r)$ is the utility that user r receives by the allocated rate x_r. In essence, the utility represents the degree of satisfaction of a user as a function of the transmission rate. Moreover, the optimization variable vector $x = (x_r, r \in R)$ is the vector with the allocated rates of all users and $C = (C_l, l \in J)$ is the vector containing the capacities of all links l. Finally, $A = (A_{lr, l \in J, r \in R})$ is a $0-1$ routing matrix of the network with $A_{lr} = 1$ denoting that route r contains link l and $A_{lr} = 0$ otherwise[4]. The physical interpretation of this formulation is the maximization of the total utility of the system (objective function) while taking into account that the total rate flowing through each link can be at most equal to the capacity of that link (problem constraint).

The authors decompose problem (1.18) into two simpler problems that can be solved distributedly by each user and the network with minimum information exchange. The proposed problems can be solved optimally if problem (1.18) is convex, i.e only under the assumption that the utility functions $U_r(x_r)$ are increasing, strictly concave and continuously differentiable functions of x_r. Under these assumptions, the authors propose a set of differential equations based on Lyapunov functions that solve the problem optimally in a distributed way.

In 1999, Low et al. [18] proposed an alternative methodology to solve the same resource allocation problem. Instead of using differential equations they develop a methodology based on *Duality Theory*. Initially, they form

[4] J and R represent the sets of all links and all users in the system respectively.

the Lagrangian function and consequently the *dual* optimization problem:

$$\begin{aligned} \min_{p} \quad & D(p) \\ \text{subject to} \quad & p \geq 0 \end{aligned} \tag{1.19}$$

where p is the vector containing the dual variables and the dual objective function $D(p)$ is calculated as follows:

$$D(p) = \max_{x_r} \left[\sum_{r} \left(U_r(x_r) - x_r \sum_{l \in S(r)} p_l \right) + \sum_{l} p_l C_l \right] \tag{1.20}$$

with p_l being the dual variables and $S(r)$ the set of links that user r is using along its path and C_l the capacity of link l. Note that the summation in the parenthesis includes only the dual variables that correspond to the links along the path that user r is sending traffic. The authors propose two distributed algorithms based on *gradient projection method* that can solve the problem optimally under the assumption of concave utility functions. The algorithms are based on the iterative gradient based equations:

$$x_r(p^r(t)) = \left[U_r^{'-1}(p^r(t)) \right]_{m_r}^{M_r} \tag{1.21}$$

$$p_l(t+1) = \left[p_l(t) + \gamma \left(x^l(t) - C_l \right) \right]^{+} \tag{1.22}$$

where $p^r(t) = \sum_{l \in S(r)} p_l(t)$ is the aggregate price along the route that user r is sending traffic at time t and $x^l(t) = \sum_{r \in F(l)} x_r(t)$ is the aggregate traffic passing through link l at time t. Moreover, m_r and M_r are the minimum and maximum feasible values of rate x_r and $[a]^+$ is the projection of a into the positive plane. $F(l)$ is the set of users that send their traffic through link l.

An alternative approach for solving various formulations of the resource allocation problem using duality theory was proposed by Palomar et al. in [19] and [20]. The authors describe a detailed problem decomposition theory that allows to develop the most appropriate distributed algorithm for each convex problem formulation. The idea behind their work is to decompose the original problem into smaller subproblems that can be solved distributedly while the optimization process is coordinated by a *master problem* with minimum signaling exchange.

More specifically, the authors identify two main types of decomposition; *primal* and *dual*. The former is suitable for problems with coupling variables, while the latter for problems with coupling constraints. Additional decomposition methods include *indirect* decomposition, where the problem is transformed using *auxiliary variables* before applying a primal or dual decomposition method, and *hierarchical* decomposition, where primal/dual decomposition methods are used recursively in order to decompose the initial problem. Using these decompositions as building blocks, the authors attempt to decouple some example optimization formulations and provide *distributed* optimization algorithms, with a different trade-off among convergence speed, message overhead and distributed computation architecture, for some common optimization problems, such as problem (1.18) and the *Quality of Service* (QoS) rate allocation problem. Analytical description of the mathematical theory of decomposition as a tool to solve optimization problems can also be found in [21] and [22].

1.3.2. TCP as an Application of NUM

The most popular *resource allocation* mechanism currently in the internet is the *Transmission Control Protocol* (TCP) [23]. TCP was designed based on

heuristic techniques and best practices that recently proved to be implicitly solving a resource allocation problem. This section will present a brief overview of TCP and the pieces of work that connect it with optimization theory.

TCP is an end-to-end connection-oriented protocol. The former means that it uses an end-to-end Acknowledgement - ACK scheme in order to guarantee reliability, while the latter implies that there is a three-way handshake interactive process before any data transmission. Only when this connection has been established will the sender start transmitting packets to the destination. TCP is designed to rely only on implicit information that it can learn from the network, or in other words the protocol makes estimates of the state of the network at every time instance in order to adjust the transmission rate of a connection. The *congestion control* mechanism in TCP relies on four algorithms; *Slow Start*, *Congestion Avoidance*, *Fast Retransmit* and *Fast Recovery*.

Slow Start is used by the sender in order to adjust its transmission rate according to the rate of receipt of acknowledgements for the packets it sends. When a new TCP connection starts, the algorithm specifies a *congestion window*, which at the beginning of the TCP execution is equal to one segment[5]. Each time an acknowledgement is received, the congestion window is doubled up to the maximum window size that the receiver has already advertised. In the case where the transmission window becomes too large for the network to handle and therefore there are packets dropped due to congestion, the sender initiates the *Congestion Avoidance Algorithm*.

The *Congestion Avoidance* algorithm is used if one or more packets are dropped due to congestion. The sender realizes that when the retransmis-

[5]A typical size for the maximum TCP segment is 536 bytes.

sion timer expires without the receipt of an acknowledgement for a packet or when a number of duplicate acknowledgement packets are received. Note, that a duplicate ACK is an acknowledgement for a packet, while at least one of the previous packets has not been acknowledged yet. In that case, the sender sets the transmission window to half of the current window size with a minimum of two segments. If congestion avoidance algorithm was invoked because of a timeout, the congestion window is set to one segment and if it was invoked because of duplicate ACKs, the *Fast Retransmit* and *Fast Recovery* algorithms are evoked. For all the packets that are acknowledged during this phase, the congestion window is increased using *Slow Start* but up to half the congestion window that caused the lost packets initially. After that point, the congestion window will start to be increased by one for every acknowledged packet. This will force the transmission rate to increase slowly towards the value that caused the congestion earlier.

The *Fast Retransmit* algorithm is invoked when duplicate ACKs are received. This could have happened for two reasons. First, a TCP segment was lost but the next one was transmitted and acknowledged successfully or, second, the segment was delayed in the network and was received out of order and therefore other packets were acknowledged before that one. Normally, if three or more duplicate ACKs have been received, the sender will assume that the packet was dropped somewhere in the network and will immediately retransmit the dropped packet.

The *Fast Recovery* algorithm, which is actually a variation of the *Slow Start* algorithm, is used so that the transmission rate recovers to relatively high level faster when duplicate ACKs have been received. The idea behind this algorithm is that since duplicate ACK packets have been received, the packet was lost most probably not due to serious congestion in the network

but due to an instantaneous network problem and should be treated as an isolated event. The algorithm consists of a *Fast Retransmit* period followed by a *Congestion Avoidance* period where the new congestion window is larger than the default *Slow Start* value.

TCP was initially designed to operate efficiently in wired networks and the most popular example of that is its use at the Internet. However, as technology evolved and networks have migrated from cables to the wireless medium, a major disadvantage of the protocol was revealed that prevents it from operating efficiently on wireless networks. A wired link is generally considered a reliable medium that packets almost never get lost for reasons other than congestion. However this is not the case for wireless links, where the interference among links can cause errors that are not related to congestion and therefore must be treated differently. TCP can not distinguish between these two causes of error. Therefore, if a packet is transmitted with errors, while there is no congestion in the network, the receiver will not send an ACK for that and the protocol will assume that there is congestion and consequently will use the *congestion avoidance* algorithm leading to a reduction of the transmission rate. However, the optimal choice in that case would be to retain the current rate and transmit the packet again.

There have been various attempts to improve TCP performance in such lossy systems. Balakrishnan et al. [24] make a comparison between the most important of these mechanisms, which make use of two different approaches. The first tries to hide any non-congestion related losses from the congestion protocol and, therefore, requires no changes to existing transmitter implementations. The intuition behind this approach is that TCP does not need to be aware of the characteristics of individual links and any losses that might occur due to the wireless medium and, therefore, tries to

make the lossy links to appear as links of higher quality but of lower bandwidth. The second approach attempts to make the transmitter aware of the existence of wireless links in the network and realize when a packet loss is not due to congestion with the use of *Explicit Loss Notification* (ELN) in the acknowledgement packets.

The congestion control schemes described in [24] are classified in three groups based on their fundamental philosophy: *End-to-end* methods, *Split-connection* methods and *Link-layer* methods. *End-to-end* methods try to improve TCP based on two techniques: *Selective Acknowledgements - SACKs* [25] and *Explicit Loss Notification - ELN* [26]. The *Selective Acknowledgements* allow the sender to recover from multiple packet losses within a single window by receiving acknowledgement only for the packets that have been successfully received. The *Explicit Loss Notification*[6] mechanism is used when a received packet is in damage. The receiver, then, sets the ELN bit in the corresponding ACK header to inform the sender that the packet was received damaged due to the bit-error rate of the channel and not due to congestion in the network. This requires that the header of the packet must have been received without any errors so that the receiver can read its sequence number in order to send ACK for it.

The *Split-connection* protocols split every TCP connection into two separate connections, one between the base station and the receiver, and one between the base station and the transmitter. Then, a wireless transmission protocol that lacks the disadvantages of TCP can be used over wireless links. An example of such a *Split-connection* protocol is the *Snoop Protocol* [28] that introduces the *snoop agent* at the base station. The snoop agent monitors every packet that passes through a TCP connection and main-

[6] mentioned also in literature as *Explicit Congestion Notification - ECN* [27].

tains a cache of the segments sent across the link and have not yet been acknowledged. If duplicate ACKs are received for a segment the agent retransmits it and suppresses the duplicate acknowledgements. Most of the *Link-layer* protocols use techniques such as *Forward Error Correction* and retransmission of lost segments in response to *Automatic Repeat Request - ARQ* messages. A link-layer protocol has the advantage that it can operate independently of protocols in higher layers of the protocol stack. Typical examples of *Link-layer* protocols are CDMA [29], TDMA [30] and AIRMAIL [31].

Over the years, different variations of the TCP protocol have been suggested, such as TCP Reno [32], TCP Vegas [33] and TCP New-Reno [34] that try to address disadvantages of the initial version of TCP, also known as TCP OldTahoe, and propose improvements to it [35]. When, all these variations of TCP were designed, there was no interest in looking at the congestion control issue as an optimization problem but rather a number of heuristic approaches were followed that proved to be working efficiently. However, Low et al. [36][37][38] proved that TCP is achieving congestion control by implicitly solving an optimization formulation of the resource allocation problem.

Specifically, they show that the optimization problem in (1.18) is solved implicitly using a primal/dual optimization algorithm. Similarly as in (1.18), the source rates are the primal variables and the congestion measures are the dual variables. Moreover, the primal iteration, which determines the source rate, is carried out by TCP while the dual iteration is carried out by an *active queue management* (AQM) algorithm, such as DropTail, RED [39] or REM [40]. Moreover, it has been shown that the different TCP versions responsible for determining the source rates leads to a different utility

TCP Version	Utility Function
TCP Reno-1	$U(x_s) = \frac{\sqrt{\frac{3}{2}}}{D_s} \tan^{-1}\left(\sqrt{\frac{2}{3}} x_s D_s\right)$
TCP Reno-2	$U(x_s) = \frac{1}{D_s} \log \frac{x_s D_s}{2 x_s D_s + 3}$
TCP Vegas	$U(x_s) = \alpha_s d_s \log x_s$

Table 1.1.: TCP version and User Utility Function

function for the optimization problem. Table 1.1 shows three variations of TCP along with the respective user utility functions. The difference between Reno-1 and Reno-2 is that the former halves the window every time a mark by the AQM protocol is found on a packet, while the latter halves it only once. Then, D_s is the equilibrium round trip time[7], α_s is a parameter of TCP Vegas and d_s is the round trip propagation delay.

Formulating TCP as a primal/dual algorithm that solves an optimization problem allows researchers to compare TCP with other optimization-based approaches with respect to how accurate the optimization problem they solve is based on the characteristics of current communication networks. Such comparison allows us to identify some of the shortcomings of TCP and the original NUM framework and motivate us for further research in the area.

1.3.3. Shortcomings of NUM

The NUM framework as proposed in [17] and [18] makes two restricting assumptions. The first assumption is that all links in the network have fixed capacity that does not change during the optimization process, and the second is that all user utilities are concave functions of the transmission rate. These assumptions are necessary to assure that the optimization formulation is convex. *Convexity* of an optimization problem is considered the *watershed*

[7] i.e. propagation delay plus equilibrium queuing delay

that differentiates an easy from a hard optimization problem [41]. The reason is the fact that convex optimization problems have a number of convenient mathematical properties. These include:

- Convex problems can be solved with gradient-based algorithms since a local optimum is also a global optimum.
- Distributed algorithms can be developed to calculate the optimal solution. This property is particularly important for network applications that consist of independent nodes whose behavior can not easily be controlled centrally.
- Strong duality holds for convex problems, which allows the development of algorithms solving the dual problem since both problems, primal and dual, have the same solution.

However, despite the aforementioned advantages, these two assumptions are responsible for a number of shortcomings of NUM when applied to current networks.

Networks with links of fixed capacity can be only assumed when all links are wired. However, current communication networks consist of a number of wireless links, whose capacity is not constant but is affected by other wireless transmissions in the neighborhood that interfere at the receivers. In other words, modelling current communication networks should take into account the interference among links and, therefore, the capacity of the wireless links can not be assumed to be fixed for the duration of the optimization process.

Concave utilities are ideal to model applications that generate *elastic* traffic [42]. Elasticity describes an application's ability to adapt easily to changes in the network conditions, such as delay, throughput etc., while

still meeting the user's *Quality of Service* (QoS) needs. Examples, of elastic applications include FTP and HTTP, which used to be the majority of the internet traffic until recently [43]. However, the majority of the traffic in current communication networks is generated by real-time applications which are considered *inelastic*. Such inelastic applications include VoIP, video streaming etc. that can not be modelled using concave utilities. As mentioned earlier, according to Cisco [3][4] the percentage of traffic generated by inelastic applications is expected to reach 57% of the Internet traffic and 66% of the mobile traffic by 2015. Modelling those applications using concave utilities can lead to significantly inefficient resource allocations.

As explained above, in practice, the resource allocation is carried out nowadays mostly using TCP, which is implicitly solving an optimization problem. However, this optimization formulation is not appropriate to model current communication networks for a number of reasons. First, similarly to the NUM framework, the formulation assumes that all links have fixed capacity. Moreover, user satisfaction is not only modelled using concave utilities independently of the type of application, elastic or inelastic, but also the utility function is the same for all users, as shown in Table 1.1, not taking into account if the application is FTP, HTTP or VoIP. These reasons make TCP operate suboptimally in current networks and highlights the necessity for a new optimization-based resource allocation protocol[8].

Designing new optimization-based resource allocation protocols is a research direction that has become particularly popular lately. Authors in [44] argue that instead of designing protocols based on heuristics that can be tuned for particular applications, network designers should move towards the direction of designing optimization-based protocols that operate opti-

[8]Quantitative results on the improvement that an optimization-based algorithm can provide against current TCP implementations are presented in Section 2.2.

mally for each particular application. Moreover, the authors provide a set of guidelines on how optimization theory can be applied in traffic management in current networks. Specifically, they suggest methods to convexify the problem constraints, decouple them using auxiliary variables and combine different objectives in order to derive optimization-based protocols.

1.3.4. Extending NUM for Current Communication Networks

As explained in the previous paragraphs, the main assumptions in the initial NUM framework were, first, that the utilities must be concave functions of the transmission rate, and, second, that all links have fixed capacity. Regarding the former, Shenker [43] highlighted the differences between *elastic* and *inelastic* traffic and the fact that concave functions can not model inelastic applications efficiently.

The utility function of a user represents the degree of satisfaction that a user enjoys when sending at a specific rate. In other words, the user utility function reflects the *Quality of Experience* (QoE) of a user when some data content is delivered at a specific data rate. This QoE cannot be determined precisely for each user, but prior work in the literature has identified approximate forms/shapes of QoE for various applications. The author in [43] was the first to suggest various non-concave single-sigmoidal utility shapes to model user satisfaction for applications that generate inelastic traffic, such as multimedia applications. Within the context of resource allocation, a single-sigmoidal utility is a shape that has one convex region followed by a concave one. The intuition behind this utility shape is that low values of rate offer very low degree of satisfaction to the user, and as the allocated data rate increases, user satisfaction increases rapidly until a point

Application	Utility Function
HTTP	$U(x_s) = U_{max} \frac{\log\left(\frac{x_s}{x_s^{min}}\right)}{\log\left(\frac{x_s^{max}}{x_s^{min}}\right)} \frac{sgn\left(x_s - x_s^{min}\right)+1}{2}$, $0 \leq x_s \leq x_s^{max}$
VoIP	$U(x_s) = U_{max} \frac{sgn(x_s - x_s^{min})+1}{2}$, $0 \leq x_s$
IPTV	$U(x_s) = \frac{U_{max}}{1+\frac{1}{\epsilon-1}e^{-x_s \cdot \alpha}}$, $\alpha = \frac{2\ln\left(\frac{1}{\epsilon-1}\right)}{x_s^{max}}$ and $0 \leq x_s \leq x_s^{max}$

Table 1.2.: Application Types and the Respective User Utility Functions

where saturation starts appearing and user satisfaction reaches its maximum value. These approximate shapes were later defined more accurately based on several traffic investigations and measurements in [45].

Table 1.2 summarizes the proposed utility function for each application type. Note that U_{max} is the maximum value of the utility function, which is typically set to 1 and function $sgn(x)$ is the sign function which takes value -1 if $x < 0$, 0 if $x = 0$ and 1 otherwise. Finally, x_s^{min} and x_s^{max} are the minimum and maximum data rates supported by the application. For example, for HTTP applications these variables take the values $x_s^{min} = 24Kbps$ and $x_s^{max} = 10Mbps$. For IPTV and generally video streaming applications, the authors in [46] propose a slightly different utility function that follows the shape:

$$U(x_s) = c\left(\frac{1}{1+e^{-\alpha(x_s-b)}} + d\right), \tag{1.23}$$

where α, c and d are calibration parameters and b is the inflection point of the sigmoidal shape, i.e. the point where the second derivative of the utility function diminishes. Figure 1.2 shows a graphic representation of these utilities. In the case of the *Voice over IP* (VoIP) utility function, the threshold rate is $x_s^{min} = 64Kb/s$, whereas $x_s^{min} = 24Kb/s$ for the utility function of HTTP applications.

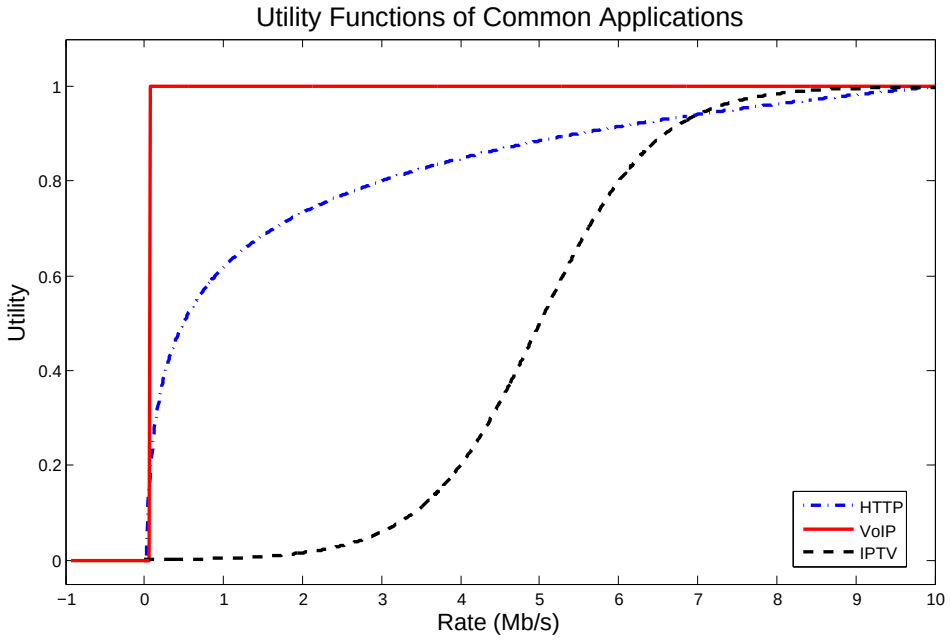

Figure 1.2.: Utility Functions of Widely Used Application Types

Since these seminal pieces of work that introduced the NUM framework, researchers have proposed extensions that address one or both of the initial framework's restricting assumptions. To the best of our knowledge, the first significant attempt to remove the assumption of concave utility functions was published by Fazel et al. in 2005 [15]. The authors are trying to solve the resource allocation problem (1.18) when the utility functions can be non-concave. Despite the fact that the problem they are trying to solve is an NP hard problem, they make use of a family of convex semi-definite programming techniques based on the *Sum of Squares* relaxation and the *Positivstellensatz Theorem* in order to solve an approximation of the initial problem. They develop a centralized algorithm that can offer a bound of the maximum network utility in polynomial time along with a sufficiency test that can reveal whether the bound is exact or not. This method was

a significant step towards solving some non-convex optimization problems, but it cannot be decomposed into a distributed algorithm and therefore does not have much practical interest in the networking area.

Chiang et al. examined the case of resource allocation for applications generating *inelastic traffic*, i.e applications such as video streaming, VoIP etc. For the special case of sigmoidal utilities, they propose in [47] and [48] a set of necessary and sufficient conditions so that the initial *NUM* formulation in problem (1.18) can be solved distributedly using an iterative gradient based algorithm despite the fact that the formulation is not convex. To achieve that they express the optimal rate allocation x_s as a function of the *dual* variables and they prove that continuity of the rate allocation function around at least one of the optimal prices is a necessary condition so that the distributed algorithm proposed in [17] can converge to the globally optimal solution. Note that the primal problem in this case is solved using the gradient algorithm by restricting the possible rate values within the concave region of the utility function. Moreover, the authors argue that by appropriate capacity provisioning in modern networks, it is possible to restrict the distributed algorithm to regions where the necessary and sufficient conditions hold.

Regarding the continuity properties of the optimal rate allocation with respect to the dual variables for the case of single-sigmoidal utilities, the authors in [46][49] extend the *NUM* framework even further. Their work examines the continuity properties of x_s as a function of the dual variables λ. Specifically, the optimal rate allocation depends on the aggregate price for all the links that the traffic is using to reach the destination. The authors prove that x_s has the following properties when the utilities are single-sigmoidal:

- $x_s(\lambda^s)$ has two values (zero and positive) and is discontinuous at λ^s_{max},
- $x_s(\lambda^s)$ is positive and decreasing function of λ, for $\lambda^s_{min} \leq \lambda^s < \lambda^s_{max}$,
- $x_s(\lambda^s)$ is zero for $\lambda^s > \lambda^s_{max}$,
- $x_s(\lambda^s)$ is M_i for $\lambda^s \leq \lambda^s_{min}$ and
- $U_s(x_s(\lambda^s))$ is achieved at the concave region of U_s.

Note that for single-sigmoidal utility functions M_s is the maximum transmission rate, λ^s_{min} is the maximum non zero aggregate price for which user s transmits at the maximum rate, i.e. M_s, and λ^s_{max} is the maximum aggregate price the user s is willing to pay. For any aggregate price higher than λ^s_{max}, user s will always select a zero optimal transmission rate.

According to the first property, function $x_s(\lambda^s)$ is discontinuous at only one point. This discontinuity point can cause oscillations when trying to solve the problem using a gradient iterative method. The authors also propose a set of conditions with at least one of them to hold when a user oscillates. These conditions, group the users that send traffic through a link l in three subsets:

$$\begin{aligned} S^H(l, \boldsymbol{\lambda}) &= \{i | \lambda^s_{max} > \lambda^s, s \in S(l)\} \\ S^S(l, \boldsymbol{\lambda}) &= \{i | \lambda^s_{max} = \lambda^s, s \in S(l)\} \\ S^L(l, \boldsymbol{\lambda}) &= \{i | \lambda^s_{max} < \lambda^s, s \in S(l)\} \end{aligned} \tag{1.24}$$

where $\boldsymbol{\lambda}$ is the vector containing the prices of all links in the network and $S(l)$ is the set containing the users that send traffic through link l. In other words, subset S^H includes the sources whose maximum willingness to pay is *higher* than the aggregate price, S^S those that the aggregate price is *equal*

to their maximum willingness to pay, and, finally, subset S^L includes the sources whose maximum willingness to pay is *lower* than the aggregate price along the path they are using. Based on these subsets, the authors in [46] prove that for a link l^* and when the Lagrangian is not differentiable for price vector $\boldsymbol{\lambda}^o$ at least one of the the following conditions holds, and in this case one or more users oscillate:

1. $\sum_{s \in S^H(l^*, \boldsymbol{\lambda}^o)} x_s(\lambda^s) < C_{l^*} - \epsilon_1$ and $\sum_{s \in S^H(l^*, \boldsymbol{\lambda}^o) \cup S^L(l^*, \boldsymbol{\lambda}^o)} x_s(\lambda^s) > C_{l^*} + \epsilon_2$
2. $\sum_{s \in S^H(l^*, \boldsymbol{\lambda}^o)} x_s(\lambda^s) \leq C_{l^*}$ and $\sum_{s \in S^H(l^*, \boldsymbol{\lambda}^o) \cup S^L(l^*, \boldsymbol{\lambda}^o)} x_s(\lambda^s) > C_{l^*} + \epsilon_3$
3. $\sum_{s \in S^H(l^*, \boldsymbol{\lambda}^o)} x_s(\lambda^s) < C_{l^*} - \epsilon_4$ and $\sum_{s \in S^H(l^*, \boldsymbol{\lambda}^o) \cup S^L(l^*, \boldsymbol{\lambda}^o)} x_s(\lambda^s) \geq C_{l^*}$

where ϵ_1, ϵ_2, ϵ_3 and ϵ_4 are positive constants.

To resolve these oscillations, the authors propose a heuristic, called *Self-Regulating Property* of a user, that drives the user causing the oscillation in the network to stop transmitting and allows the rest of the users to stabilize to a finite solution. This approach has been shown to approach the optimal solution as the number of users in the network tend to ∞. This approach, however, is a form of admission control in the network, since it is excluding some users from being allocated resources and, therefore, can be questioned for its fairness.

Research in the area of resource allocation in wireless networks has not been as extensive as in the case of wired networks. While the resource allocation problem in wired networks requires a Transport layer mechanism to adapt the transmission rates of the users, the problem in wireless networks turns into a joint Transport and MAC layer optimization problem that optimizes both the transmission powers of the wireless links and the allocated rates of the users. Therefore, any proposed approaches must rely on cross-

layer algorithms. Nonetheless, various approaches have been proposed that deal differently with this problem in wireless networks.

In [50] the authors examine the case of *NUM* in three wireless network scenarios, a *Single-Cell* downlink scenario, a *Multi-Cell* downlink scenario and a *Hybrid* network scenario with both wired and wireless links. The main difference of the formulation proposed is the fact that the capacity of a wireless link is no longer fixed and known a priori but it depends on the transmission power of the transmitter. The authors examine both cases of no interference between base stations and cases with interference while making the necessary assumptions to preserve convexity of the problem and therefore the distributed nature of the algorithm. More specifically, in the single-cell downlink case, the authors assume that the resource allocation problem is formulated as:

$$\begin{aligned} \max \quad & \sum_i U_i\left(R_i\right) \\ \text{subject to} \quad & R_i \leq \log\left(g_i P_i\right) \quad ,\forall i, \\ & \textstyle\sum_i P_i \leq P^{max}, \\ & \boldsymbol{P} \geq \mathbf{0} \end{aligned} \tag{1.25}$$

where R_i, the rate of user i, and P_i, the transmission power of user i, are the optimization variables, P^{max} is the maximum transmission power of the based station, g_i is the channel gain and $\boldsymbol{P}$ is the vector containing all transmission powers. Problem (1.25) is convex and therefore can be easily solved distributedly. However, its applicability is limited and therefore the authors proposed the multi-cell downlink problem formulation that takes into account the interference among different cells and introduces the *Signal-*

to-Interference Ratio (SIR) defined as:

$$SIR_i(\boldsymbol{P}) = \frac{G_{ii}P_i}{\sum_{j\neq i}^{N} P_j G_{ij} + N_i} \tag{1.26}$$

where G_{ij} is the path loss coefficient from the transmitter of link i to the receiver of link j. Consequently, the authors define the transmission rate as a function of the SINR at the receiver according to:

$$R_i = \frac{1}{T}\log\left(1 + K \cdot SINR_i\right) \tag{1.27}$$

where T is the symbol time and K is a constant that depends on the modulation type and the desired bit error probability. However, in order to preserve convexity of the problem formulation they assume that the SINR at the receiver is always much larger than 1 and approximate (1.27) using:

$$R_i = \frac{1}{T}\log\left(K \cdot SINR_i\right). \tag{1.28}$$

Based on this capacity function, the authors propose a distributed algorithm to allow each node in the cellular network to determine the optimal transmission power and rate.

A similar approach was followed in [51], where the author examines the resource allocation problem in wireless multi-hop networks. The formulation is similar with problem (1.18) with the main difference being the fact that the capacity C_l of a link is not constant but is, instead, a function of the SINR at the receiver, i.e $C_l(\boldsymbol{P}) = \frac{1}{T}\log\left(1 + K \cdot SINR_l(\boldsymbol{P})\right)$. However, the channel capacity function makes the formulation non-convex and therefore any distributed algorithm is possible to converge to a local but not global optimum. To resolve this issue, the author assumes that the system

operates in high SINR environment, which allows the channel capacity to be approximated efficiently by

$$C_l(\boldsymbol{P}) = \frac{1}{T}\log\left(K \cdot SINR_l(\boldsymbol{P})\right). \tag{1.29}$$

Consequently, it is proved that (1.29) is a strictly concave function of the transmission powers. In order to prove this, a log-transformation of the capacity function is required. With the use of (1.29) to compute the channel capacity at each iteration, the author proposes a joint rate and power allocation distributed algorithm based on the TCP congestion control mechanism that can optimize the performance of the network under the presence, however, only of elastic applications. This algorithm consists of the following iterative equations:

$$\begin{aligned} x_s(t+1) &= \frac{w_s(t+1)}{D_s(t)} \\ \lambda_l(t+1) &= \left[\lambda_l(t) + \frac{\gamma}{c_l(t)}\left(\sum_{s:l\in L(s)} x_s(t) - c_l(t)\right)\right]^+ \qquad (1.30) \\ P_l(t+1) &= P_l(t) + \frac{\kappa\lambda_l(t)}{P_l(t)} - \kappa\sum_{j\neq l} G_{lj} m_j(t) \qquad (1.31) \end{aligned}$$

where γ and κ are positive step sizes, $w_s(t+1)$ is the TCP window size and $m_j(t)$ is a *message* that is calculated using:

$$m_j(t) = \frac{\lambda_j(t)\, SINR_j(t)}{P_j(t)\, G_{jj}}. \tag{1.32}$$

To assure convergence of the power control iterative algorithm, the author assumes the existence of minimum and maximum power values, $P_{l,min}$ and $P_{l,max}$ respectively.

In [52] the authors examine the case of downlink power allocation in CDMA cellular networks and remove the assumption for concave utility functions. They show that the optimal power allocation occurs when the base station is transmitting at full power and prove that the properties of the transmission power, as a function of the dual prices λ, follow the properties of $x_i(\lambda^s)$ as described in [46]. Their proposed algorithm allows cooperation between the base station and the mobile nodes. At the first stage of their proposed algorithm, called *mobile selection* stage, the base station selects the mobiles that will be allocated some power and during the second stage, called *power allocation* stage, the base station allocates power to the selected mobile nodes.

Hou et al. in [53] extended the *NUM* framework for wireless multi-hop sensor networks with explicit consideration in the sensors' energy constraint. This emphasizes the fact that a typical sensor is powered by a battery and thus has limited lifetime. The authors are trying to maximize the amount of traffic that will be transmitted in the life time of the network given that each sensor has a battery of specific energy capacity. The problem formulation proposed is shown to be convex under some assumptions and a distributed gradient based algorithm is proposed.

In our previous work [54], we have extended the *NUM* framework in order to take into account the interference among links while retaining the convexity of the problem formulation. To achieve this, the formulation in [17] was extended with an additional constraint to assure that there will be a minimum *SINR* level for every wireless link in the network. In other words, the traditional resource allocation formulation for the optimization variables r_i, representing the rates, and p_j, representing the transmission

powers, was extended as follows:

$$\begin{aligned} \max \quad & \sum_{i=1}^{M} U_i\left(r_i\right) - \sum_{j=1}^{L} V_j\left(p_j\right) & \\ \text{subject to} \quad & \sum_{i \in Z(j)} r_i \leq C_j & \forall \text{ links j} \\ & \frac{G_{jj} p_j}{\sum_{k=1, k \neq j} G_{jk} p_k + n_j} \geq \gamma_j & \forall \text{ links j} \end{aligned} \tag{1.33}$$

where γ_j is the target SINR ratio for link j, $Z\left(j\right)$ represents the set of traffic flows that pass through link j, G_{ij} is the path loss gain coefficient from the transmitter or link i to the receiver of link j, $V_j\left(p_j\right)$ is a cost function which represents the cost of using the limited power resources of a wireless network and n_j is the noise at the receiver of link j. In other words, the first constraint of the formulation proposed is responsible for the rate allocation and the second constraint is actually a power control problem. Under the assumptions of concave utility functions $U_i\left(r_i\right)$ and convex cost functions $V_j\left(p_j\right)$ the problem is convex and a distributed algorithm is proposed that will converge to the optimal solution as long as the power control problem is feasible. [55] provides a necessary and sufficient condition for the existence of a feasible power vector. The intuition behind the use of the SINR threshold in the second constraint is that the capacity C_j of link j will be guaranteed if the threshold γ_j is satisfied and the following expression relates the two quantities:

$$C_j = B \cdot \log_2\left(1 + \gamma_j\right), \tag{1.34}$$

where B is the channel bandwidth.

Recently, researchers have extended the wireless channel model in resource allocation problems by incorporating phenomena such as channel fading and bit error probability. For example, the authors of [56] prove

that the resource allocation problem in wireless networks, where nodes distributedly optimize transmission powers and rates, has zero duality gap if the channel *Cumulative Distribution Function* (CDF) is continuous. Moreover, this continuity requirement is satisfied by several practical channel models such as Rayleigh, Rice and Nakagami. The proof of zero duality gap for some of the most widely used channel fading models is a significant theoretical contribution. From a practical perspective, however, this result will be applicable to communication networks only under the development of distributed methods to solve non-convex optimization problems.

In the same context, Papandriopoulos et al. [57] propose a resource allocation formulation that takes into account the *rate-outage probability* in slow fading channels. Based on a target rate-outage probability, the authors propose a channel capacity formula, they scale the SINR of the channel so that the resulting capacity satisfies the target rate-outage requirements. More specifically, the authors suggest the following channel capacity formula:

$$C_l\left(\boldsymbol{P}\right) = B \cdot \log_2\left(1 + M_l \cdot \overline{SINR}\left(\boldsymbol{P}\right)\right), \tag{1.35}$$

where $\overline{SINR}\left(\boldsymbol{P}\right)$ is the average $SINR$, and M_l is a positive weighting scalar, which is a function of the rate-outage probability, according to:

$$M_l = -\log\left(1 - \Omega_l^{rate}\right) \tag{1.36}$$

where Ω_l^{rate} is the maximum tolerable rate-outage probability. Moreover, the authors prove that the capacity function (1.35) is *quasiconcave* and the resulting resource allocation formulation is convex if the utility functions are at least (log,x)-concave. This relaxes the requirement for concave utilities and covers a number of common applications, such as the utility functions

of TCP (TCP Vegas, TCP Reno etc).

Another research work that takes into account the time varying characteristics of the wireless medium is described in [58]. The authors enhanced the NUM framework so that the average performance of the wireless network is optimized over time and the optimal control policy is selected to anticipate with the existence of time varying interference conditions. The authors use the so-called *Full Resource Optimization with Expected Constraints* (FROEC) method to solve the optimization problem, which takes the sequence of channel states, as seen by the network, as its input and produces estimates of the optimal Lagrange multipliers and optimal policy values. The method samples the condition of the network periodically and calculates stochastic gradients in order to calculate an estimate of the optimal resource allocation. Therefore, this approach could be classified in the area of *Stochastic Optimization* [59].

Despite the fact that this PhD thesis will remain focused on deterministic resource allocation problem formulations and deterministic optimization techniques, stochastic optimization techniques have been used, in order to address the issues of the traditional NUM framework that relate to the stochastic dynamics of modern networks, and therefore should be briefly mentioned in this literature review. The authors in [60] group the challenges when dealing with the dynamics in networks in three categories: session level, packet level and constraint level. The first category refers to the issues that arise from the random arrival rates of sessions in the network and the finite queue lengths at intermediate nodes. Research in this area includes determining the stochastic stability region of a network based on specific arrival and service models. Prior work in the session level research area is presented in references [61], [62], [63] and the references therein.

Stochastic network utility maximization at the packet level deals with the burstiness of the incoming packets and the short-term dynamics that include probabilistic marking and dropping of packets. Prior work in this research area can be found in [64], [65], [66] and the references therein. Finally, constraint level stochastic problems deal with the dynamic of the wireless channel and try to maximize the network utility and assure stability of the network under such phenomena. Recent work in this area includes [67], [68], [69] and [70].

More specifically in [70], the author tries to solve an optimization formulation that can be applied to the stochastic resource allocation problem in communication networks. It is anticipated that the user utility function can be single-sigmoidal to model real-time multimedia applications, which turns the formulation into a non-convex problem. The solution proposed calculates a local optimum of the problem based, on the *drift-plus-penalty* approach [71] and Lyapunov optimization, while assuring the stability of the queues in the network.

1.3.5. The Notion of Fairness in NUM

The resource allocation problem in communication networks describes the problem of sharing the network resources to competing users so that we maximize the satisfaction in the network. The notion of *fairness* plays a very important role in the process of resource sharing and the attempt to maximize satisfaction in the network.

There are a number of different fairness policies in research work that relate to the Network Utility Maximization (NUM) framework. Kelly et al. in their seminal work [17], which introduced the NUM framework, define the notion of *(bandwidth) proportional fairness* and prove that the distributed

algorithm that they propose to solve the resource allocation problem shares bandwidth according to this. More specifically, they define a vector of rates $\boldsymbol{x}$ to be proportionally fair if it is feasible and for any other feasible vector $\boldsymbol{x}^*$ we have that:

$$\sum_{r \in R} w_r \frac{x_r^* - x_r}{x_r} \leq 0, \tag{1.37}$$

where w_r is a positive weighting term that refers to user r and x_r is the proportionally fair rate of user r. In other words, a (bandwidth) proportionally fair vector is a vector that maximizes the sum of a number of logarithmic utility functions.

Another common fairness policy is the *(bandwidth) max-min* criterion [72], which is also called the *bottleneck optimality* criterion [73]. According to the definition, a feasible rate vector is *(bandwidth) max-min* fair if any rate x_i can not be increased without decreasing another rate x_j which is smaller or equal to x_i. In essence, *max-min* fairness is the optimization of the worst case. Most of the algorithms in literature that achieve *max-min* fairness require significant amount of information exchange and therefore it is difficult to implement a truly distributed algorithm for the resource allocation problem. Moreover, the authors in [74] argue that max-min fairness is not appropriate for wireless multi-hop networks as it leads to equal rates and powers regardless of the network topology and routing, despite the fact that modern networks are often asymmetric and such allocations might not be feasible. The authors in [75] propose a simple *water-filling* procedure to achieve (bandwidth) max-min fairness:

1. Start from a bit rate equal to zero for all flows in the network;

2. Increase the bit rate of all flows uniformly until the bit rate of some flows is constrained by the capacity set; freeze the bit rate of these

flows;

3. Repeat Step 2 for any non-frozen flows until all flows in the network are constrained by the capacity set.

Tassiulas et al. in [76] propose a scheduling policy in wireless ad hoc networks that achieves max-min fairness. The authors use the example of bluetooth to create the wireless channel model for their analysis and prove a necessary and sufficient condition regarding max-min fairness of a bandwidth allocation. According to this, *a bandwidth allocation is max-min fair if and only if every flow satisfies at least one of the following conditions: (a) the flow has at least one bottleneck node, (b) the bandwidth allocated to the flow equals its long term arrival rate.* The authors also argue that the fact that max-min fairness gives by default priority to the flows that receive the worst quality of service might not be desirable in modern communication networks, which are shared by flows with different quality needs that also follow different pricing schemes and, therefore, their proposed max-min policy associates a priority weight to each flow.

Lately, Wang et al. [77] show that while *(bandwidth) proportional fairness* is efficient when all users follow the same logarithmic utility functions, it has some contradictory behavior in heterogeneous networks, i.e. in networks where users follow different utility functions and have different QoS needs. This happens due to the fact that the (bandwidth) proportional fairness policy allocates rates based on the value of the utility derivative. Users with the largest derivative are allocated the most rate. However, this seems unfair when dealing with heterogeneous networks because large value of derivative means that this particular user is easily satisfied. For instance, single-sigmoidal utility functions, such as the one presented in Fig-

ure 1.2, are more difficult to satisfy compared to concave utilities because their derivative is small for low rate regions. This causes a (bandwidth) proportional fair algorithm to allocate rate first to concave applications, such as browsing and file transfer, and allocate the remaining rate to the more demanding multimedia ones. Therefore (bandwidth) proportional fairness has the counter intuitive behavior that allocates less rate to users that need it the most.

To resolve this contradictory behavior, the authors in [77] propose a novel type of fairness, the so-called *utility proportional fairness*. Following the same intuition as the initial (bandwidth) proportional fairness, a bandwidth allocation vector $\boldsymbol{x}^*$ is utility proportionally fair if it is feasible and for any other feasible vector $\boldsymbol{x}$ we have that:

$$\sum_{s\in\mathcal{S}} \frac{x_s - x_s^*}{U_s\left(x_s^*\right)} \leq 0. \tag{1.38}$$

Of course, when $U_s\left(x_s\right) = x_s$, the utility proportional fairness policy reduces to the initial bandwidth proportional fairness one. Consequently, the authors propose the following resource allocation formulation to allow the development of a utility proportional fair distributed algorithm:

$$\begin{aligned} \max \quad & \sum_{s=1}^{S} \mathcal{U}_s\left(x_s\right) \\ \text{subject to} \quad & \sum_{s\in\mathcal{S}_l} x_s \leq c_l, \quad \text{l=1, } \ldots\text{, L} \end{aligned} \tag{1.39}$$

where $\mathcal{S}_l$ is the set of flows that use link l to send their traffic to the destination nodes and the *transformed* utility function $\mathcal{U}_s\left(x_s\right)$ is given by:

$$\mathcal{U}_s\left(x_s\right) = \int_{m_s}^{x_s} \frac{1}{U_s\left(y\right)} dy, \quad m_s \leq x_s \leq M_s, \tag{1.40}$$

where m_s and M_s are the minimum and maximum transmission rates for user s respectively. This transformation of the utility function offers a number of advantages to problem (1.39) compared to problem (1.18). First, problem (1.39) is convex even for non-concave utility functions. This means that a distributed gradient-based algorithm will be able to calculate the optimal solution without any oscillations or local optimality problems. Then, as the authors in [77] propose, a utility proportional fair policy can be implemented using the following iterative equations:

$$x_s^*\left(\lambda^s\right) = U_s^{-1}\left(\frac{1}{\lambda^s}\right) \tag{1.41}$$

$$\lambda_l(t) = \lambda_l(t-1) - \gamma(t)\left(\sum_{s\in\mathcal{S}_l} x_s - c_l\right) \tag{1.42}$$

where $\gamma(t)$ is a small positive step size and $U_s^{-1}(\cdot)$ is the inverse of the user's utility function. Moreover, the authors prove that the aforementioned problem formulation can lead to *utility max-min fair* resource allocations if the path price is defined as:

$$\lambda^s = \max_{l\in\mathcal{L}_s} \lambda_s\left(t\right), \tag{1.43}$$

where set $\mathcal{L}_s$ includes all the links along the path that user s is using. In other words, by changing λ^s from the aggregate price of the path to be the largest link price of the path the distributed algorithm achieves utility max-min fairness. Note that *utility max-min fairness* was defined in [78] based on the initial bandwidth max-min fairness following the same intuition as definition of utility proportional fairness.

1.4. Motivation and Contributions

Section 1.3 presented in detail the most significant prior work in literature in the area of Network Utility Maximization. Despite the significant advancements and extensions since the initial *NUM* framework, there are a number of remaining research challenges that motivated our work and on which this thesis attempts to make some contributions.

As explained earlier, the initial NUM framework [17] made two restricting assumptions; all utilities must be concave and the capacity of all links in the network is constant. There has been significant work in extending the framework with respect to these two assumptions on either one or the other direction. However, these extensions are restricted only to specific resource allocation formulations and do not provide a general optimization framework. Moreover, recent pieces of work that show that TCP is implicitly solving an optimization problem, where all applications are modelled using the same concave utility, highlight the need for the development of a novel optimization-based transport layer protocol that will be able to optimize the allocation of resources in networks utilized by heterogeneous applications and consisted of both wired and wireless links.

With this motivation, Chapter 2 makes the following contributions. First, we show that current resource allocation protocols, such as TCP, that were designed intuitively using heuristics, fail to allocate resource optimally in current communication networks. Then, we propose a general non-convex optimization framework that can be applied to any optimization problem and prove a sufficient condition to identify the non-convex formulations that can be solved optimally by the framework. This condition is also proven to be necessary as well under mild conditions. This general framework can be

the basis of a novel optimization-based resource allocation protocol. Consequently, we discuss the phenomenon of user oscillations and propose an efficient heuristic to resolve such oscillations and allow the distributed algorithm to approximate the optimal solution. To illustrate the applicability of the non-convex framework, we propose a non-convex resource allocation problem formulation in wireless TDMA/CDMA ad-hoc networks and propose a specific distributed algorithm to jointly optimize transmission powers and data rates, which will be extensively simulated in various network topologies.

All pieces of work in literature that extend the NUM framework by considering non-concave utilities use single-sigmoidal functions of the form described in (1.23) to model inelastic applications. However, single sigmoidal utilities may not be suitable to model current multimedia applications for the following reason. Most video streaming applications used nowadays support service at different quality levels. Each one of these quality profiles has different requirements and lead to different level of user satisfaction, which can not be modeled satisfactorily by single sigmoidal utilities.

The work presented in Chapter 3 is motivated by the inability of single-sigmoidal utilities to model multi-tiered inelastic applications. This chapter introduces the concept of multi-sigmoidal utilities and explains the reasons that make the incorporation of such utilities in NUM suitable. In addition, we provide a detailed analysis on the implications of such a choice in the continuity properties of the rate allocation function, a significant aspect of the non-convex framework presented in Chapter 2. Consequently, we propose a mathematical representation of such a multi-sigmoidal function and discuss how the parameters of this function can be calibrated. The non-convex problem formulation that results from the incorporation of such

utilities is examined and a distributed algorithm is proposed to solve it when possible. When a solution can not be obtained due to oscillations, an extension of the oscillation resolving heuristic of the previous chapter is proposed to approximate it.

As explained in the previous sections, the initial NUM framework and most of the work proposed in literature allocates resources according to the bandwidth proportional fairness policy. This policy however has some significant drawbacks when dealing with heterogeneous networks, i.e. networks that are used by both elastic and inelastic applications. These drawbacks relate to the fact that applications that need more data rate, such as multimedia applications, tend to receive less bitrate compared to applications that as easily satisfied, such as FTP, HTTP etc. In addition, the absence of convexity of the resulting optimization problem creates unwanted phenomena such as users oscillating and preventing the distributed algorithm from converging to the optimal solution. *Utility Proportional Fairness* seems to be a promising alternative in allocating resources and assuring the convexity of the optimization problem even with non-concave utilities. However, prior work in utility proportional fairness is limited only to networks with wired links and for a short range of utility functions.

In order to exploit the potential of the incorporation of Utility Proportional Fairness (UPF) in NUM, the work presented in Chapter 4 makes the following contributions. First, we discuss the advantages of UPF regarding the convexity of the optimization problem, its ability to prevent user rate oscillations and lead to closed form solutions for the optimal rate allocation function. Then, we propose a resource allocation problem formulation for high-SINR wireless networks and propose a distributed utility proportional fairness algorithm to solve it. Furthermore, we describe an analytical

methodology to derive analytical solution of the most widely used types of applications and show that UPF allows the utilization of the full range of the feasible rates contrary to bandwidth proportional fairness where rate oscillations restrict users to only a small portion of it. The proposed methodology is also simulated extensively in various wireless network topologies.

1.5. Thesis Organization

This PhD thesis is organized as follows. Chapter 2 describes a non-convex optimization framework and proves the condition that allows a distributed gradient-based algorithm to calculate the optimal solution of a non-convex problem formulation. The advantage of this framework is its generality and its applicability to a wide range of applications, such as the non-convex resource allocation formulation that will be proposed and solved. then, Chapter 3 introduces the notion of multi-sigmoidal utilities, proposes a novel mathematical representation of such functions based on hyperbolic tangent functions, provides arguments about the need of using such functions in NUM and proves significant theoretical results regarding the continuity properties of the optimal rate allocation function under the presence of multi-sigmoidal utilities. These functions are incorporated in a resource allocation formulation in wired networks and the performance of the proposed distributed algorithm is evaluated in a number of network topologies. Chapter 4 proposes a utility proportionally fair distributed algorithm for wireless networks, provides an analytical methodology to calculate closed form solutions for the optimal rate allocation function for a number of widely used utilities, including the multi-sigmoidal utility proposed in Chapter 3. The algorithm is also evaluated for various wireless network topologies. Finally,

Chapter 5 concludes our work and outlines our future research plans.

2. A Non-convex Distributed Optimization Framework and its Application to Wireless Ad-hoc Networks

2.1. Introduction

Modern communication networks must encompass and simultaneously support multiple users, services and applications with diverse demands and requirements that push networks' performance closer to their limit. Therefore, optimum resource allocation between users and/or applications is of paramount importance in order to assure efficient utilization of the network. The *resource allocation* problem is one of the numerous research areas in which *Optimization Theory* has found extensive use, since it can lead to the development of distributed algorithms to assure optimal allocation of resources in a network.

As described earlier in Chapter 1, Kelly et al. in their seminal paper [17], and Low et al. [18] later using a different mathematical approach, introduced the *Network Utility Maximization (NUM)* framework, where the *resource allocation* problem is expressed as an optimization problem. This

convex optimization framework has found numerous applications in network resource allocation in wired and wireless networks. The main focus of these pieces of work, however, are on modeling applications that generate *elastic* traffic[42]. TCP is an example of a protocol designed to perform optimally for this traffic in wired networks. However, modern internet traffic is dominated by real-time applications, such as video and audio streaming, that are considered *inelastic*[42].

The main challenge when attempting to optimize networks shared by inelastic applications is that they cannot be modeled using concave utility functions and therefore the resulting problem turns into a non-convex one, which is difficult to solve. This is because, contrary to what happens in *convex optimization*, the gap between the primal and dual optimal solutions in *non-convex* problems can be positive and then more sophisticated techniques must be employed to solve them [7]. The lack of convexity due to the existence of inelastic traffic in current communication networks, makes TCP operate suboptimally. Recent work tries to relax the assumption for concave utilities in the context of *NUM* by proposing the use of sigmoidal or step functions to model such traffic.

Most of the aforementioned work is restricted only to specific non-concave formulations and do not provide a general optimization framework. The absence of alternative transport protocols to allow network optimization for inelastic applications is the main motivation behind this work. This chapter makes the following contributions:

- Demonstrates the inability of current resource allocation protocols, such as TCP, to behave optimally in current communication networks.

- Proposes a non-convex optimization framework that removes the crit-

ical assumptions for convexity of the problem formulation and proves a sufficient (and in some cases necessary) condition so that the framework can solve a non-convex optimization formulation. The significance of this framework is its generality and, therefore, its suitability to a wide range of applications.

- Proposes an efficient resource allocation heuristic to resolve user oscillations that occur when the condition does not hold.

- Presents an application of the aforementioned framework in wireless TDMA/CDMA ad-hoc networks. The proposed resource allocation formulation, firstly, incorporates the interference among links, and secondly, introduces a power penalty term in the objective function to ensure convergence and energy efficiency of the power control subproblem.

- Develops a distributed joint rate allocation and power control algorithm, which enables network nodes to optimize their performance, even for the case of inelastic traffic.

The rest of this chapter is organized as follows. Section 2.2 highlights the shortcomings of the widely used TCP protocol in allocating bandwidth to networks shared by various types of applications. Section 2.3 presents the general optimization framework and proves a sufficient condition to assure optimality of the solution. In Section 2.4, the framework is applied to the *resource allocation problem* in *wireless* ad-hoc networks and a distributed gradient-based algorithm is proposed. The case of source rate oscillation is discussed and an efficient heuristic is proposed to resolve it efficiently. Then, the performance of the method is evaluated by simulations in Section 2.5, and, finally, Section 2.6 summarises the work presented in this chapter.

2.2. TCP in Current Communication Networks

The *Transmission Control Protocol - TCP* [23] is currently the most popular *resource allocation* mechanism. As mentioned analytically in Chapter 1, TCP is an end-to-end connection-oriented protocol which relies only on implicit information that is used to estimate the state of the network and adjust the transmission rate of a connection. The *congestion control* in TCP is implemented using a "window", whose size varies based on an implicit measurement of the congestion in the network; the more unacknowledged packets, the more congestion in the network. The size of the window essentially determines the transmission rate of the source with larger window leading to higher bitrate. Over the years, a number of TCP variations have been proposed in order to overcome some of the shortcomings of the initial protocol with the most popular being TCP Reno [32] and TCP Vegas [33].

TCP was designed based on a set of practical algorithms to adjust the size of the transmission window without any optimization theory considerations. However, Low et. al [37][38] proved that TCP implicitly solves a resource allocation optimization problem and that the various TCP variations differ in the utilities comprising the objective function of the problem. More specifically, TCP Reno solves Problem (1.18) with utility function $U_r(x_r) = \frac{1}{D_r} \log \frac{x_r D_r}{2x_r D_r + 3}$, where D_r is the round trip delay, and TCP Vegas solves the same problem but with utility function $U_r(x_r) = \alpha_r d_r \log x_r$, where α_r is a positive calibration parameter and d_r is the round trip propagation delay of source r.

It is evident from the above that the resource allocation mechanism of TCP assigns the same concave utility function to all flows in the network independently of the nature of the application generating the traffic. When

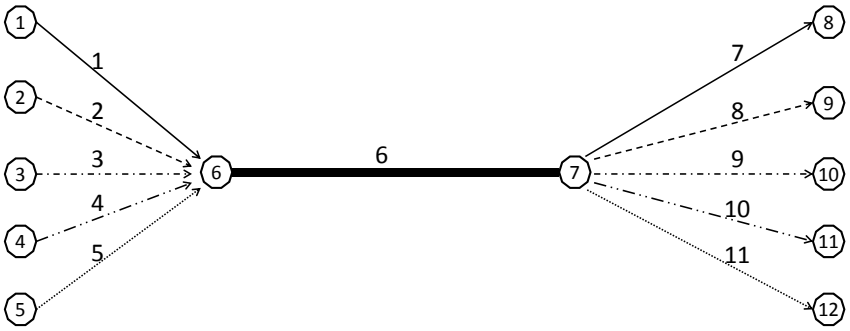

Figure 2.1.: Example of a Single-bottleneck Network

TCP was designed, the majority of the traffic over the Internet was elastic but the capacity of current communication networks is mainly used for real-time applications [3][4]. With such significant amount of traffic generated by inelastic applications, the use of TCP can lead to significantly suboptimal resource allocations.

An optimization-based algorithm, such as Algorithm 1 presented later in this chapter, can allocate the resources of current networks more efficiently. The use of such an algorithm to allocate network resources would have two advantages over TCP. First, each application in the network will be modeled using a different utility function based on the user quality perception for this application. This implies that elastic applications will be modeled using concave utilities and inelastic using non-concave ones.

To illustrate the performance improvement that can be achieved using an optimization-based resource allocation algorithm, consider the single bottleneck wired network topology of Figure 2.1, which consists of five traffic flows that share the capacity of link 6. The capacity of links $1-5$ and $7-11$ is assumed to be sufficiently large to serve any transmission rate of source nodes $1-5$ while the capacity of link 6 is assumed to be insufficient to accommodate all flows at their maximum transmission rate, thus creating

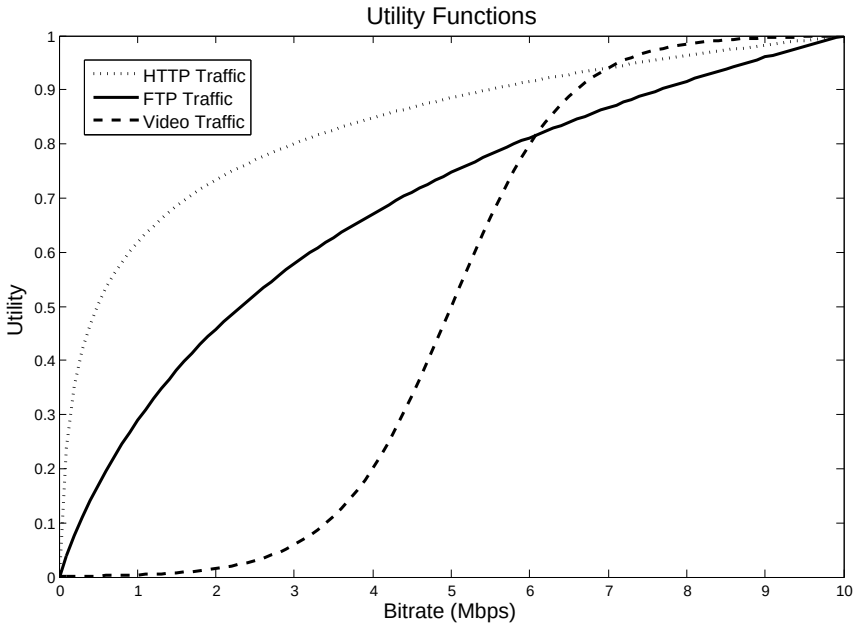

Figure 2.2.: Example of utility functions

a bottleneck in the network. For the comparison shown in this section the bottleneck link was set to $28Mb/s$. The applications sharing the network included HTTP, FTP and video streaming.

The utilities that were used are shown in Figure 2.2. FTP and HTTP applications have been proven [36]-[38] to follow concave utility shapes and such application were assumed to dominate the traffic according to the NUM framework. Video applications however for a single-sigmoidal utility such as that shown with dashed line in Figure 2.2. The intuition behind such as utility shape is the following: When the bitrate is very low (e.g. 0-4 Mb/s), the video quality is particularly low and hence the user is very dissatisfied with the resulting video. As the bitrate increases (e.g. 4-7 Mb/s), however, quality is improved vastly and therefore user satisfaction increases rapidly. This rapid increase in user satisfaction continues until a point where the quality is already exceptional (e.g. for bitrates above

7 Mb/s) and any further increase in the bitrate will not cause significant increase in the perceived by the user quality of the video.

Figure 2.3 shows the improvement that can be achieved if the resource allocation is carried out by an optimization-based algorithm as opposed to the congestion control mechanism in TCP. The two methods were compared while the number of real-time applications varied. The x-axis in both subfigures shows the number of real-time applications out of five applications that compete for resources in the network. The rest were either HTTP or FTP applications. For example, the performance comparison for two real-time applications corresponds to a scenario with two sigmoidal utilities, one FTP concave utility and two HTTP concave utilities. The red and black lines at the top figure show the total network utility that each method achieved, while the blue line at the bottom shows the percentage of improvement that the optimization-based algorithm achieved.

It is evident that the more real-time applications share the network, the worse TCP performs by modelling all applications with the same concave utility. On the other hand, an optimization-based algorithm can allocate network bandwidth efficiently since it uses a different utility for each application. Moreover, the improvement in performance can be even larger in networks with a number of wireless links since Algorithm 1 takes into account the interference in order to calculate the link capacities while TCP does not.

Motivated by these results, the next section will focus on the development of an optimization framework that can offer the foundations of future optimization-based resource allocation protocols.

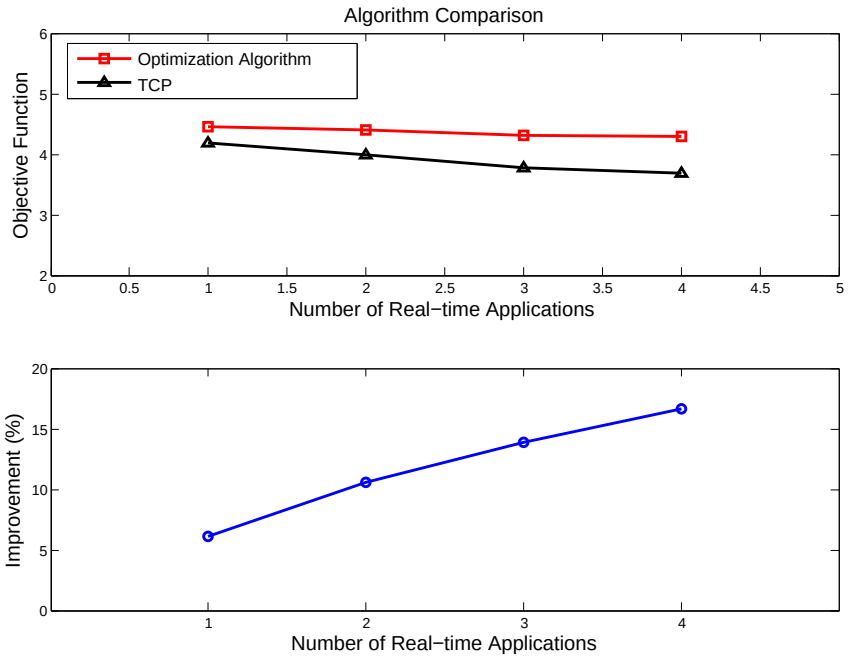

Figure 2.3.: Improvement of the Optimization Algorithm over TCP

2.3. An Optimization Framework for Non-convex Problems

The *NUM* framework as presented in [17] and [18] is restricted by the need for concave utilities and the fact that the capacity of all links is fixed. However, as explained above, such assumptions are not valid for the majority of current communication networks. Any prior work that attempts to remove any of them refers to very specific applications, thus lacking generality. This highlights the need for a general non-convex optimization framework that will be able to solve optimization problems resulting from any non-convex network application.

Not all non-convex optimization problems are difficult to solve. In fact, there are cases that can be solved as easy as a convex optimization problem. *Therefore, our main consideration is to develop an optimization framework*

that can first identify such non-convex problems and then solve them in a distributed way, while being generic enough in order to cover as many applications as possible. Towards the development of such a framework, first, consider the following maximization problem over the vector of variables $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]$:

$$\begin{aligned} \max_{\boldsymbol{x}} \quad & f(\boldsymbol{x}) \\ \text{s. t.} \quad & h_i(\boldsymbol{x}) \leq 0,\, \boldsymbol{x} \geq \mathbf{0}. \quad \forall\, i \in [0, M] \end{aligned} \tag{2.1}$$

To form the *dual* problem, we first define the *Langrangian* function $L\left(\boldsymbol{x}, \boldsymbol{\lambda}\right) = f\left(\boldsymbol{x}\right) + \sum_{i=0}^{M} \lambda_i h_i\left(\boldsymbol{x}\right)$, where M is the number of constraints of the optimization problem, λ_i is the dual variable associated with the i^{th} constraint and $\boldsymbol{\lambda}$ is the vector containing all dual variables. According to *Duality Theory*, the *dual* objective function is defined as $d\left(\boldsymbol{\lambda}\right) = \sup_x L\left(\boldsymbol{x}, \boldsymbol{\lambda}\right) = \sup_{\boldsymbol{x}} \left\{ f\left(\boldsymbol{x}\right) + \sum_{i=0}^{M} \lambda_i h_i\left(\boldsymbol{x}\right) \right\}$ and the dual optimization problem is:

$$\begin{aligned} \min_{\boldsymbol{\lambda}} \quad & d(\boldsymbol{\lambda}) = L(\boldsymbol{x}^*(\boldsymbol{\lambda}), \boldsymbol{\lambda}) \\ \text{s. t.} \quad & \boldsymbol{\lambda} \geq 0, \end{aligned} \tag{2.2}$$

where $\boldsymbol{x}^*\left(\boldsymbol{\lambda}\right)$ is a function that maximizes the Lagrangian for a given vector $\boldsymbol{\lambda}$, i.e.

$$\boldsymbol{x}^*(\boldsymbol{\lambda}) = \arg\max_{\boldsymbol{x}} L(\boldsymbol{x}, \boldsymbol{\lambda}). \tag{2.3}$$

Each of the dual variables λ_i corresponds to a specific inequality constraint that are often referred as *shadow prices*. In addition, $\boldsymbol{x}^*\left(\boldsymbol{\lambda}\right)$ is the optimal solution of problem (2.1) for the particular price vector $\boldsymbol{\lambda}$. The dual function $d\left(\boldsymbol{\lambda}\right)$ is always convex as a point-wise supremum of a family of affine functions of $\boldsymbol{\lambda}$ and problem (2.2) is always convex even if the primal problem (2.1) is not concave [6]. Therefore, it is possible to solve the dual problem

using the iterative equation:

$$\lambda_i(t+1) = \lambda_i(t) - \delta_\lambda \frac{\partial L(\boldsymbol{x}, \boldsymbol{\lambda})}{\partial \lambda_i} \tag{2.4}$$

where δ_λ is the step size and $\frac{\partial L(\boldsymbol{x},\boldsymbol{\lambda})}{\partial \lambda_i}$ is the partial derivative of Lagrangian function with respect to λ_i. The uniqueness of the optimal vector $\boldsymbol{\lambda}$ is not guaranteed in all cases but prior work in literature can provide necessary and sufficient condition for its uniqueness [79].

Equations (2.3) and (2.4) constitute an iterative primal-dual optimization algorithm which would converge to the optimal solution if problem (2.1) had been concave. However, convergence to the optimal is not guaranteed otherwise. Nonetheless, there are non-concave problems where the duality gap is zero and (2.3) and (2.4) can converge to the optimal solution. To identify these cases, one can use the condition of Theorem 1.

Theorem 1 (Sufficient Condition)**.** *If the price based function* $\boldsymbol{x}^*(\boldsymbol{\lambda})$ *is continuous around at least one of the optimal Lagrange multiplier vectors* $\boldsymbol{\lambda}^*$ *then the iterative algorithm consisting of equations* (2.3) *and* (2.4) *converges to the globally optimal solution.*

Proof. We start by showing that continuity of $\boldsymbol{x}^*(\boldsymbol{\lambda})$ around the optimal dual variables λ_i^* implies that *complementary slackness* is satisfied for problem (2.1). Recall that the *complementary slackness* condition states that $\lambda_i^* h_i(\boldsymbol{x}^*(\boldsymbol{\lambda}^*)) = 0$, $\forall i$ at the optimal solution $x^*(\boldsymbol{\lambda}^*)$.

First, the case where $\lambda_i^* > 0$ for an arbitrary chosen i is examined. A very small positive constant $\epsilon > 0$ and a new vector $\boldsymbol{\lambda}^-$ are defined where

$$\lambda_j^- = \begin{cases} \lambda_j^* - \epsilon & \text{, if } j = i \\ \lambda_j^* & \text{, if } j \neq i \end{cases} \tag{2.5}$$

In other words, vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}^-$ differ only at one element, which has been reduced by the constant ϵ. Then, by definition of the sub-gradient, we have that

$$d(\boldsymbol{\lambda}^*) \geq d(\boldsymbol{\lambda}^-) + (\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^-)^T \boldsymbol{\Lambda}(\boldsymbol{\lambda}^-) \qquad \Leftrightarrow \qquad d(\boldsymbol{\lambda}^*) \geq d(\boldsymbol{\lambda}^-) + \epsilon \frac{\partial L(\boldsymbol{x}, \boldsymbol{\lambda}^-)}{\partial \lambda_i} \Leftrightarrow$$

$$d(\boldsymbol{\lambda}^*) - d(\boldsymbol{\lambda}^-) \geq \epsilon h_i(\boldsymbol{x}^*(\boldsymbol{\lambda}^-)). \tag{2.6}$$

where $\boldsymbol{\Lambda}$ is a vector containing the partial derivatives of the Lagrangian with respect to the dual variables, i.e. $\boldsymbol{\Lambda} = [\frac{\partial L(\boldsymbol{x}, \boldsymbol{\lambda})}{\partial \lambda_i}, i \in [1, M]]$. But since the dual problem is a minimization problem and $\boldsymbol{\lambda}^*$ is its optimal solution, it follows that $d(\boldsymbol{\lambda}^*) \leq d(\boldsymbol{\lambda}^-)$ and hence by (2.6)

$$h_i(\boldsymbol{x}^*(\boldsymbol{\lambda}^-)) \leq 0. \tag{2.7}$$

Working at the same way, a second vector $\boldsymbol{\lambda}^+$ is defined as

$$\lambda_j^+ = \begin{cases} \lambda_j^* + \epsilon & \text{, if } j = i \\ \lambda_j^* & \text{, if } j \neq i \end{cases} \tag{2.8}$$

Again, by definition of the sub-gradient, it follows that

$$d(\boldsymbol{\lambda}^*) \geq d(\boldsymbol{\lambda}^+) + (\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^+)^T \boldsymbol{\Lambda}(\boldsymbol{\lambda}^+) \Leftrightarrow d(\boldsymbol{\lambda}^*) \geq d(\boldsymbol{\lambda}^+) - \epsilon \frac{\partial L(\boldsymbol{x}, \boldsymbol{\lambda}^+)}{\partial \lambda_i} \Leftrightarrow$$

$$d(\boldsymbol{\lambda}^*) - d(\boldsymbol{\lambda}^+) \geq -\epsilon h_i(\boldsymbol{x}^*(\boldsymbol{\lambda}^+)). \tag{2.9}$$

But for the same reason as before, $d(\boldsymbol{\lambda}^*) \leq d(\boldsymbol{\lambda}^+)$ and hence by (2.9), we conclude that

$$h_i(\boldsymbol{x}^*(\boldsymbol{\lambda}^+)) \geq 0. \tag{2.10}$$

From (2.7) and (2.10) we get to the conclusion that as long as $\boldsymbol{x}^*(\boldsymbol{\lambda})$ is

continuous around $\boldsymbol{\lambda}^*$, then

$$h_i(\boldsymbol{x}^*(\boldsymbol{\lambda}^-)) = h_i(\boldsymbol{x}^*(\boldsymbol{\lambda}^+)) = h_i(\boldsymbol{x}^*(\boldsymbol{\lambda}^*)) = 0 \tag{2.11}$$

and hence *complementary slackness* is satisfied and the solution $\boldsymbol{x}^*(\boldsymbol{\lambda}^*)$ is primal feasible.

Then, the case where $\lambda_i^* = 0$ is examined. In this case, it is obvious that *complementary slackness* is satisfied. Primal feasibility of the solution can be shown using the positive constant ϵ and the price vector $\boldsymbol{\lambda}^+$ are defined as before. Equation (2.10) is reached again and under the continuity condition it follows that $h_i(\boldsymbol{x}^*(\boldsymbol{\lambda}^+)) \geq 0$. Hence, the *complementary slackness* condition is satisfied under the condition that the price-based function $\boldsymbol{x}^*(\boldsymbol{\lambda})$ is continuous at the optimal price vector $\boldsymbol{\lambda}^*$.

By definition of the dual problem, its optimal solution is given by $d^* = f(\boldsymbol{x}^*(\boldsymbol{\lambda}^*)) + \sum_{i=0}^{m} \lambda_i^* h_i(\boldsymbol{x}^*(\boldsymbol{\lambda}^*))$, and since *complementary slackness* holds, it reduces to $d^* = f\left(\boldsymbol{x}^*\left(\boldsymbol{\lambda}^*\right)\right)$, which by definition of the primal problem is $f\left(\boldsymbol{x}^*\left(\boldsymbol{\lambda}^*\right)\right) \leq f^*\left(\boldsymbol{x}\right)$ and hence $d^* \leq p^*$. But by *weak duality* it is known that $d^* \geq p^*$ and therefore it follows that $d^* = p^*$, where p^* and d^* are the optimal values of the primal and the dual problem respectively.

Therefore, it has been proven that continuity of the price based function (2.3) around at least one of the optimal price vectors implies that the *duality gap* is zero and that by solving the dual optimization problem the optimal solution $\boldsymbol{x}^*$ is also obtained. □

The aforementioned condition is also a necessary condition for convergence of the distributed gradient-based algorithm for some non-convex optimization problems as the following theorem suggests.

Theorem 2. *If at least one constraint of problem* (2.1) *is active at the optimal solution, the condition in Theorem 1 is also a necessary condition.*

Proof. According to Complementary Slackness, which is a necessary condition for optimality, the fact that at least one constraint is active at the optimal solution implies that at least one of the optimal Lagrange multipliers is non-zero and therefore the algorithm cannot converge unless (2.11) holds. Hence, continuity of $\boldsymbol{x}^*(\boldsymbol{\lambda})$ around at least one of the optimal Lagrange multiplier vectors is a necessary condition. □

Theorems 1 and 2 provide a condition for convergence to the globally optimal solution by the gradient-based algorithm consisted of equations (2.3) and (2.4). Note that (2.3) represents the optimal rate for a given price vector. In the case of non-concave utilities, the optimization problem described in (2.3) is also non-convex. However, as shown in the remainder of this thesis, this is a simpler problem that is in some cases easier to be solved, especially for resource allocation formulations, such as the ones described later, by taking advantage of the exact shape of the user utility function and its continuity properties.

The condition in Theorem 1 constitutes a significant contribution to optimization theory in general. Compared to other pieces of work, such as [47], that refer to specific non-convex *NUM* formulations in wired networks, this work provides a far more general optimization formulation and therefore can be widely applicable. The applicability of the framework to a specific problem relies on the continuity properties of the price-based function $\boldsymbol{x}^*(\boldsymbol{\lambda})$. Even though the development of a general procedure to determine continuity of $\boldsymbol{x}^*(\boldsymbol{\lambda})$ for any optimization problem is an open research issue, there are cases that either the calculation of a closed form solution is possible or

the continuity properties of $\boldsymbol{x}^*(\boldsymbol{\lambda})$ are known. Nonetheless, this is a significant result that shows that a family of non-convex problems can be solved distributedly using a gradient based method.

2.4. Resource Allocation in Wireless Ad-hoc Networks

The non-convex optimization framework presented in the previous section can be applied to the resource allocation problem in wireless networks in order to identify and solve non-convex problem formulations that stem from the incorporation of inelastic traffic and the existence of wireless links in the network. The analysis of such a non-convex formulation is the focus of this section.

2.4.1. Problem Formulation

Consider a multi-hop wireless network where each node can operate either as traffic source, destination or relay that just forwards traffic to its neighbors. We define the transmission rate vector $\boldsymbol{r} = [r_1, r_2, \ldots, r_M]^T$ which includes the transmission rates of all M source nodes in the wireless network. Moreover, we define the link l as the tuple (T_l, R_l), where T_l is the transmitting and R_l the receiving node, respectively. We also define $\boldsymbol{p} = [p_1, p_2, \ldots, p_L]^T$ as the vector which includes the transmission powers of the L links. The wireless channel is modelled as follows. Let $\boldsymbol{G}$ be a matrix of size $L \times L$, where G_{km}, with $k, m \in 1, 2, \ldots, L$, represents the path loss coefficient for the path between the transmitter of link k and the receiver of link m. The elements of the path loss matrix $\boldsymbol{G}$ depend on the physical characteristics of the wireless links.

The network performance optimization can be formulated as a maximization problem of the form:

$$\begin{aligned} \max_{\boldsymbol{r},\boldsymbol{p}} \quad & \sum_{i=1}^{M} U_i(r_i) - \gamma \sum_{l=1}^{L} V_l(p_l) \\ \text{s. t.} \quad & \sum_{i=1}^{M} \alpha_{il} r_i \leq C_l\left(\boldsymbol{p}\right), \qquad \forall \text{ links } l \end{aligned} \tag{2.12}$$

where parameter α_{il} is one if the traffic of user i is passing through link l, and zero otherwise. The parameters α_{il}, with $i \in \{1, 2, \ldots, M\}$ and $l \in \{1, 2, \ldots, L\}$, form the routing matrix $\boldsymbol{A}$ of the network, which is considered to be fixed and known a priori for the duration of the optimization process. The rates r_i and powers p_l are positive quantities and γ is a positive weighting parameter.

In order to account for the half duplex limitations of wireless transceivers and avoid excessive interference, a hybrid TDMA/CDMA scheme is assumed to operate in the network. More specifically, we consider Orthogonal-CDMA (OCDMA) for transmissions towards the same receiver, and pseudo-noise-CDMA (PN-CDMA) between different receivers. This means that the transmitted signal is first spread through multiplication by a Welsh-Hadamard (WH) sequence with N chips per symbol. Then a PN sequence is overlayed either without further spreading (i.e., with the same chip rate) or with further spreading by a factor K (i.e., number of chips per WH chip). All users transmitting towards the same receiver employ the same PN sequence, and N orthogonal sequences are reused at each receiver. Moreover, TDMA is employed throughout the multihop routes. This implies that time is divided into frames, each of them comprises of two equally sized slots, where transceivers alter from transmitting to receiving mode.

Based on this channel model, the capacity of a link follows Shannon's

capacity formula, $C_l(\boldsymbol{p}) = B \cdot \log_2(1 + SINR_l)$ and is a function of the Signal to Noise plus Interference Ratio (SINR) at the receiver of the link. This formula is a non-concave function of powers and this might prevent any gradient based algorithm from converging to the optimal power vector. However, under the assumption that $SINR_l \gg 1$, the concave formula $C_l(\boldsymbol{p}) = B \log_2(SINR_l)$ can provide a sufficiently accurate approximation of link capacity [21]. Such a *high SINR* environment can be easily achievable for the aforementioned TDMA/CDMA channel model. For the remainder of this paper, the link capacity $C_l(\boldsymbol{p})$ will be calculated using this approximation.

The choices for utility $U_i(r_i)$ in problem (2.12) are not restricted to concave functions, as in the traditional NUM framework, so that the problem formulation can be applied to networks with various types of traffic. This makes problem (2.12) non-convex and therefore can be solved distributedly only if Theorem 1 holds. Comparing Problem (2.12) with other pieces of work in literature, this formulation extends NUM for *wireless* networks by allowing non-concave utility functions while considering mutual interference among links and by using a power penalty term to ensure energy efficiency and convergence of the distributed power control algorithm.

2.4.2. Distributed Algorithm

Problem (2.12) optimizes the allocation of resources in an ad-hoc network and therefore the applicability of any solution relies on the ability to develop a distributed algorithm with minimum message overhead among nodes. *Duality Theory* provides the means to develop such a distributed algorithm,

and to this purpose, we first define the *Lagrangian* function as:

$$L(\boldsymbol{r},\boldsymbol{p},\boldsymbol{\lambda}) = \sum_{i=1}^{M}\left\{U_i(r_i) - r_i\cdot\left(\sum_{l=1}^{L}\alpha_{il}\lambda_l\right)\right\} + \sum_{l=1}^{L}\lambda_l B\log\left(\frac{NKp_lG_{ll}}{\sum_{k\neq l}p_kG_{kl}+n_l}\right) - \gamma\sum_{l=1}^{L}V_l(p_l). \tag{2.13}$$

Regarding the physical meaning of the major terms on the Lagrangian function, $U_i(r_i)$ is the "profit" that source i will make for sending its traffic at rate r_i and quantity $r_i\cdot\left(\sum_{l=1}^{L}\alpha_{il}\lambda_l\right)$ represents the total cost for source i in order to send r_i b/s of traffic through the network. Then, term $\sum_{l=1}^{L}\lambda_l B\log\left(\frac{NKp_lG_{ll}}{\sum_{k\neq l}p_kG_{kl}+n_l}\right)$ represents the total "profit" that the links will make by charging each unit of their capacity with λ_l and term $\gamma\sum_{l=1}^{L}V_l(p_l)$ represents the *weighted* cost for the links to achieve a capacity of $B\cdot\log\left(\frac{NKp_lG_{ll}}{\sum_{k\neq l}p_kG_{kl}+n_l}\right)$ for $l=1,\ldots,L$. After a careful observation of the Lagrangian function, one can see that the optimization process consists of two subproblems of the primal variables $\boldsymbol{r}$ and $\boldsymbol{p}$ coupled by the dual optimization variable vector $\boldsymbol{\lambda}$. The first subproblem is the *rate allocation*, maximizing the net revenue of each *source*, and the second is a *power control* problem, determining the optimal transmission power of the *links*.

Based on the Lagrangian function, every source i can calculate its optimal rate $r_i^*(\boldsymbol{\lambda})$ using:

$$r_i^*(\boldsymbol{\lambda}) = \arg\max\left[U_i(r_i) - r_i\cdot\lambda^i\right], \tag{2.14}$$

where $\lambda^i = \sum_{l=1}^{L}\alpha_{il}\lambda_l$ is the aggregate price for user i and it represents the cost of sending a unit of traffic through the network. There are several methods to solve the optimization problem of (2.14). First, it is known that

the optimal solution will be at the point where the first derivative of the objective function diminishes and therefore

$$r_i^* (\boldsymbol{\lambda}) = {U'}_i^{-1} \left(\lambda^i\right), \tag{2.15}$$

where ${U'}_i^{-1} (\cdot)$ is the inverse function of the first derivative of the utility function. It is evident that (2.15) can be used if ${U'}_i (\cdot)$ is an one-to-one function and its inverse can be calculated or if ${U'}_i (\cdot)$ can be broken to invertible one-to-one parts. In cases, that this is not possible, one should use alternative methods, such as the gradient based iterative equation:

$$r_i(t+1) = r_i(t) + \delta_r(t) \frac{\partial L(\boldsymbol{r}, \boldsymbol{p}, \boldsymbol{\lambda})}{\partial r_i} \tag{2.16}$$

where $\delta_r(t)$ is a positive step size and the gradient of the Lagrangian function is given by:

$$\frac{\partial L(\boldsymbol{r}, \boldsymbol{p}, \boldsymbol{\lambda})}{\partial r_i} = U_i^{'}(r_i) - \sum_{l=1}^{L} \alpha_{il} \lambda_l. \tag{2.17}$$

In general, iterative gradient-based equations such as (2.16) should be used with care as they can converge to local optima instead of global. However, knowledge of the shape of the optimal rate allocation function can be used in some cases, such as in the case of Problem (2.12) to assure that (2.16) will converge to the globally optimal solution. Nonetheless, the distributed Algorithm 1 uses the general equation (2.14) to allow the implementation of the most appropriate method for r_i^t.

A similar approach can be used to calculate the power and price variables, p_l and λ_l respectively:

$$\lambda_l(t) = \lambda_l(t-1) - \delta_\lambda(t) \frac{\partial L(\boldsymbol{r}, \boldsymbol{p}, \boldsymbol{\lambda})}{\partial \lambda_l} \tag{2.18}$$

$$p_l(t) = p_l(t-1) + \delta_p(t)\frac{\partial L(\boldsymbol{r}, \boldsymbol{p}, \boldsymbol{\lambda})}{\partial p_l}, \tag{2.19}$$

where $\delta_\lambda(t)$ and $\delta_p(t)$ are small positive step sizes and the gradients are given by:

$$\frac{\partial L(\boldsymbol{r}, \boldsymbol{p}, \boldsymbol{\lambda})}{\partial \lambda_l} = B \cdot \log_2 \left(\frac{NKp_l G_{ll}}{\sum_{k \neq l} p_k G_{kl} + n_l} \right) - \sum_{i=1}^{M} \alpha_{il} r_i \tag{2.20}$$

$$\frac{\partial L(\boldsymbol{r}, \boldsymbol{p}, \boldsymbol{\lambda})}{\partial p_l} = -\gamma V_l^{'}(p_l) + \frac{1}{p_l \ln(2)} \left[\lambda_l - \sum_{m \neq l} \lambda_m \frac{G_{lm} P_l}{\sum_{k \neq m} G_{km} P_k + n_m} \right]. \tag{2.21}$$

Equations (2.14)[1], (2.18) and (2.19) constitute an iterative distributed algorithm, which is summarized in Algorithm 1. At every iteration, each link and each source are updating their power, price and rate according to the feedback they get from the network. Regarding the stopping criterion of the algorithm, one could stop the optimization process when all derivatives have diminished or when the value of the objective function has not changed significantly for a number of consecutive iterations [7]. In any case, the values of the step sizes δ_λ and δ_p constitute an important trade-off between convergence speed and accuracy. The initial vectors of $\boldsymbol{r}$, $\boldsymbol{\lambda}$ and $\boldsymbol{p}$ for $t = 0$ can be set to any feasible value.

Regarding the information exchange of the algorithm, users need to know the aggregate link price λ^i. This can be either stored in the ACK packets sent by the destination to the source node, or, if the link price is viewed as the link delay, it can be implicitly measured by the packet queuing delay in the network. In addition, the power calculation process requires that a

[1] or (2.15) if the inverse utility derivative can be calculated

Algorithm 1 – Iterative Distributed Algorithm

1: $t = 1$;
2: **repeat**
3: Links calculate p_j^t using (2.19) based on $\boldsymbol{\lambda}^{t-1}$ and channel state information;
4: Links calculate λ_j^t using (2.18), based on λ_j^{t-1}, p_j^t, channel state information and aggregate rate traversing link j;
5: New prices $\boldsymbol{\lambda}$ are sent to sources;
6: Sources calculate r_i^t using (2.14) based on received λ^i and r_i^{t-1};
7: $t = t + 1$;
8: **until** *algorithm_converges*

link knows the channel conditions of neighboring nodes. This information can be easily obtained from the lower layers of the protocol stack with no additional signaling overhead.

2.4.3. Convergence and Oscillation Resolving Heuristic

The cost function $V_l(p_l)$ assures that the optimization problem will have a finite optimal power vector. In the absence of this cost function, i.e. when $\gamma = 0$, it is possible to fall in a situation where equation (2.21) is always positive, leading to infinite powers. On the other hand, when $\gamma > 0$, there will be a finite power vector $\boldsymbol{p'}$ at which any further increase would lead to a decrease in the network utility and thus the algorithm will converge to a finite power value. In literature, this case of infinitely increasing power is often prevented by assuming a maximum transmission power value p_l^{max}. Such an assumption, even though is reasonable in a practical system, causes distortion in the theoretical analysis since it creates artificial convergence points. Specifically, according to the *Brouwer Fixed Point Theorem* [80], a continuous mapping of the power vector within a closed range $\left[p_l^{min}, p_l^{max}\right]$ creates fixed points of an algorithm that might otherwise never converge.

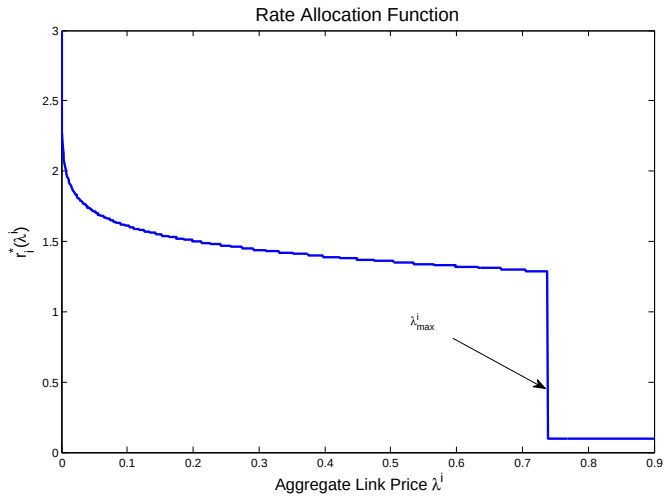

Figure 2.4.: Example of the rate allocation function $r_i\left(\lambda^i\right)$ for sigmoidal utilities

Therefore, the use of the penalty function $V_l\left(p_l\right)$ is a more natural way of assuring both energy efficiency and convergence of the distributed power control algorithm.

Algorithm 1 is an extension of the standard gradient algorithm to solve any convex optimization problem and whose convergence properties have been extensively studied in prior work [7]. According to theorems 1 and 2 the sufficient (and in some cases necessary as well) condition for optimality is continuity of (2.14) around at least one of the optimal price vectors $\boldsymbol{\lambda}^*$. The continuity properties of (2.14) rely on the shape of the utility function $U_i\left(r_i\right)$. More specifically, if $U_i\left(r_i\right)$ has a concave shape, i.e. it is modelling an elastic application, (2.14) is a continuous function of the aggregate price λ^i. If, however, $U_i\left(r_i\right)$ models an *inelastic* application, (2.14) can be discontinuous at one or more points and user oscillations can occur when the optimal price vector λ^i leads to an aggregate price (for that specific user) equal to a discontinuity point. While a generic procedure to determine the

Algorithm 2 – Source Rate Calculation (Oscillation Resolving Heuristic)

```
1: if user_takes_part_in_optimization then
2:    if user_is_oscillating (r, θ_i) then
3:       r_i^t = inflection_point_rate;
4:       user_takes_part_in_optimization = false;
5:    else
6:       Calculate r_i^t using (2.14);
7:    end if
8: else
9:    r_i^t = r_i^{t-1};
10: end if
```

continuity properties of $r_i\left(\lambda^i\right)$ for any utility function is an open research problem, these properties have been extensively studied for single-sigmoidal utilities [46]. For such utility function, $r_i^*\left(\boldsymbol{\lambda}\right)$ is discontinuous at only one point, which represents the user's maximum willingness to pay, λ_{max}^i, and there is an analytical methodology to be calculated. Figure 2.4 shows an example of the rate allocation function $r_i\left(\lambda^i\right)$ for a single-sigmoidal utility which is discontinuous for $\lambda^i = \lambda_{max}^i = 0.7385$. In the remainder of this chapter, we will assume that inelastic applications will be modelled by single-sigmoidal utility shapes, such as the one in Figure 2.2, which is the most widely used shape to model real-time multimedia applications.

The phenomenon of oscillation occurs when the optimal rate function $r_i^*\left(\boldsymbol{\lambda}\right)$ of a specific user i is a discontinuous function of the aggregate price and the optimal price vector λ^i leads to an aggregate price (for that specific user) equal to the discontinuity point. As explained earlier, the existence of discontinuity points in $r_i^*\left(\boldsymbol{\lambda}\right)$ depends only on the shape of the utility function. Specifically, for sigmoidal utilities, $r_i^*\left(\boldsymbol{\lambda}\right)$ is discontinuous only for aggregate price $\lambda^i = \lambda_{max}^i$ and when the optimal price vector leads to that aggregate price, the rate of user i oscillates and the distributed algorithm can not converge.

Practically, a user oscillation occurs when the user transmits at an excessive data rate (compared to the available capacity) in an iteration of the algorithm, and in the next iteration, the user transmits at an exceedingly low rate. An oscillation is formed as the repetition of these two events continues indefinitely, which prevents the user from converging to the optimal transmission rate. In this case, user i needs to resolve this oscillation and approximate the optimal solution. To this purpose, Algorithm 2 describes an efficient heuristic that ensures convergence to the optimal solution, when users do not oscillate, and stability when one or more users oscillate. Note that Algorithm 2 is carried out distributedly by each source node to determine the most appropriate rate at time t after an updated aggregate price is received and, in essence, replaces the initial rate update mechanism in line 6 of Algorithm 1.

Algorithm 2 is based on the idea that an oscillating user will be allocated some rate, and will be removed from the rest of the optimization process to allow stability of the network. User oscillations indicate that the optimal rate allocation is non-zero, but due to the discontinuity at λ_s^{max}, the optimal rate can not be calculated. More specifically, user i is associated with a parameter θ_i, the maximum number of consecutive oscillations before the oscillation resolving mechanism is evoked (line 2). As long as an oscillation is not detected, user i calculates its rate based on the aggregate price (line 6). Once oscillations are detected, user i starts transmitting at rate equal to the inflection point of its sigmoidal utility and leaves the optimization process (lines 2-5 and 9).

Removing oscillating users from the optimization process is an obvious decision to ensure stability of the network but the question lies in the allocated rate to these users. Authors in [46] attempt to solve the oscillation problem

by removing them without allocating any rate. However, such approaches lack fairness because they unnecessarily prevent some users from accessing the network resources. Algorithm 2 has the following advantages against this approach. The self-regulating heuristic has been proven to be optimal for wired networks with infinite number of users/data sources. If the number of users is finite though, by completely removing an oscillating user from the optimization problem, there is a non-zero probability that the remaining users will not be able to exploit the remaining available resources and therefore the resulting resource allocation is significantly suboptimal. The oscillation resolving heuristic presented in this chapter can accommodate more users since it allocates some rate even to oscillating ones. In addition, allowing more users to transmit in a high SINR environment makes a better use of the capacity of the wireless medium and ultimately leads to higher aggregate utility in the network for practical applications. This will be shown by an example in Section 2.5.

2.5. Numerical Results

Algorithms 1 and 2 were simulated in MATLAB for various network scenarios. For illustration purposes, in this section let us consider the network topology shown in Figure 2.5. The wireless network consists of four source nodes, four intermediate nodes and one destination node. Source nodes 1 and 4 serve real-time applications with single-sigmoidal utilities while source nodes 2 and 3 serve HTTP applications with concave utilities.

In the topology example of Figure 2.5, the two time slots are designated with blue and red color. In other words, nodes $1-4$, 7 and 8 transmit only at the first time slot while nodes 5 and 6 only during time slot 2. The

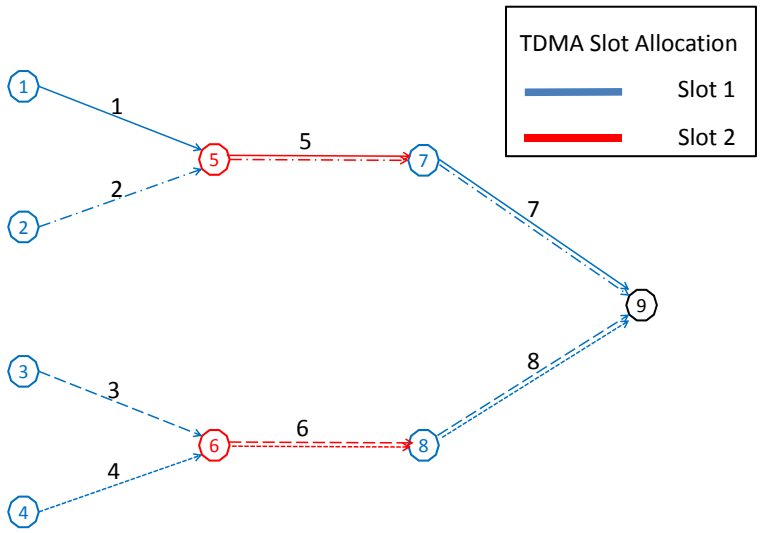

Figure 2.5.: Example Network Topology

hybrid TDMA/CDMA scheme described in Section 2.4 was deployed with $N = 2$ chips per symbol, a spreading gain $K = 4$ and channel bandwidth of $B = 2$MHz. Finally, the utility functions of the four sources where defined as $U_i(r_i) = \frac{1}{1+e^{-\alpha(r_i-\beta)}}$ [46], with $\alpha = 1.38$ and $\beta = 5$ for $i \in \{1, 4\}$, $U_2(r_2) = \frac{\log(r_2+1)}{\log(\alpha+1)}$ [45], with $\alpha = 6$, and $U_3(r_3) = \alpha \cdot log(\beta \cdot r_2 + \gamma)$ [46], with $\alpha = 0.417$, $\beta = 0.417$ and $\gamma = 1$. Regarding the feasible power vectors, it is assumed that there is a feasible power vector to achieve capacity adequate to accommodate the non-concave utilities when transmitting at rate equal to their inflection point.

The performance of Algorithms 1 and 2 is compared against that of the standard gradient algorithm when the self-regulating heuristic [46] is applied to resolve oscillations. Figures 2.6, 2.7 and 2.8 illustrate their performance. Soon after the optimization process starts, the aggregate price for users 1 and 4 exceeds their maximum "willingness to pay" and they start oscillating. As shown clearly in Figure 2.6, the rate oscillation of users 1 and 4 cause

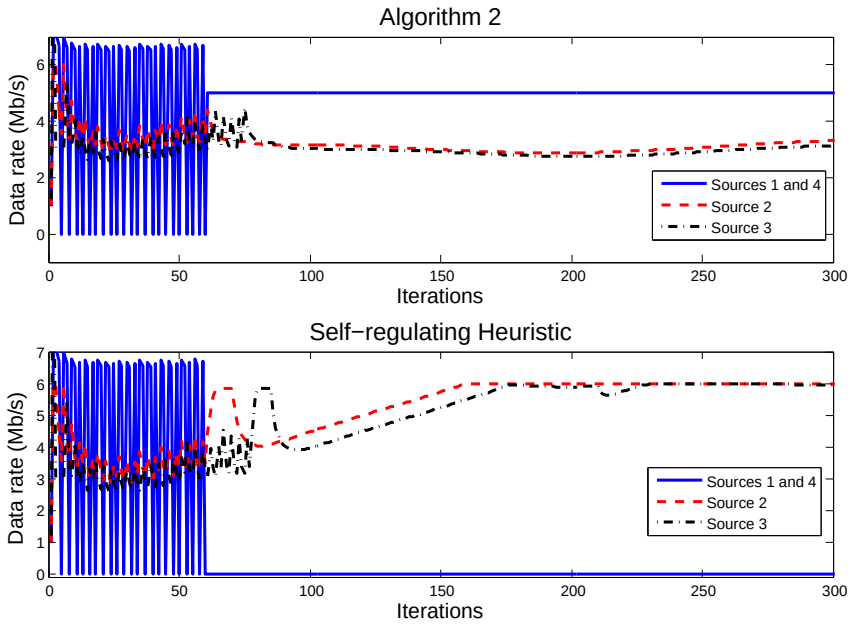

Figure 2.6.: Convergence of Rate Allocation

oscillations of smaller degree to other users as well. This happens since oscillations cause abrupt changes in the competition for resources in the network. When such an oscillation is observed, a heuristic is evoked to resolve it. Algorithm 2 sets the rate to the non-zero value of the inflection point (in this case to 5 Mb/s) and continues the optimization process. In the self-regulating heuristic case the rate is set to zero.

As illustrated in Figure 2.7, the decision for non zero rate for the oscillating users yields higher value of the objective function compared to the self-regulating heuristic, for the reasons explained earlier. Note that, as Theorem 1 states, when the optimal vector $\boldsymbol{\lambda}$ does not lead to oscillations, the optimization process comprised of Algorithms 1 and 2 converges to the globally optimal solution. Figure 2.8 shows the convergence of the transmission powers allocation of the first 4 links in the network, the ones initiated from the 4 source nodes. The difference in dealing with oscilla-

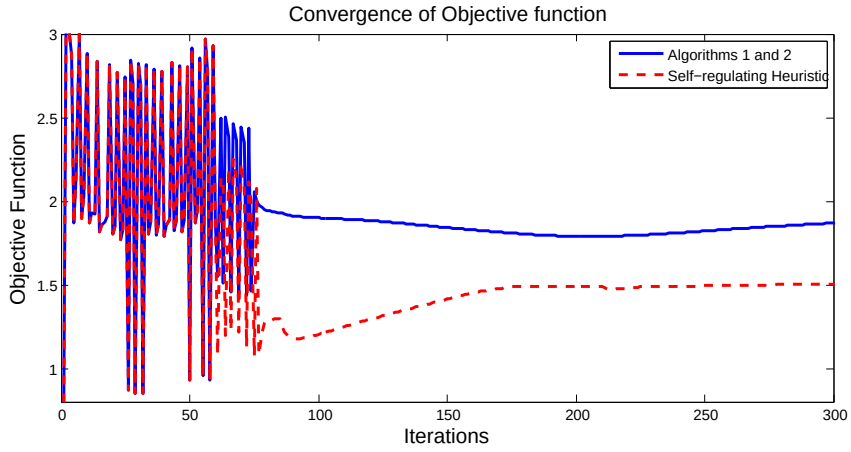

Figure 2.7.: Convergence of Objective Function

tions between the two heuristics is illustrated in the power vectors as well. The self-regulating heuristic leads to zero powers for the oscillating users while Algorithm 2 gives non zero powers to achieve the necessary channel capacity. Finally, since we have assumed the operation at a high SINR environment, we should mention that the SINR ranges from $7dB$ to $18dB$, and therefore the error introduced by our capacity approximation in the worst case is less than 10% (note that for $SINR > 10dB$ the error is less than 4%). It is important to mention here that the approximation provides an underestimation of the link capacity, and therefore the upper bound of the Shannon capacity formula is not violated. This justifies the valid use of the approximated capacity formula.

2.6. Concluding Remarks

Motivated by the non-convex resource allocation problems in *Network Utility Maximization* and the necessity for a novel optimization-based resource allocation protocol, this chapter presents a general optimization framework

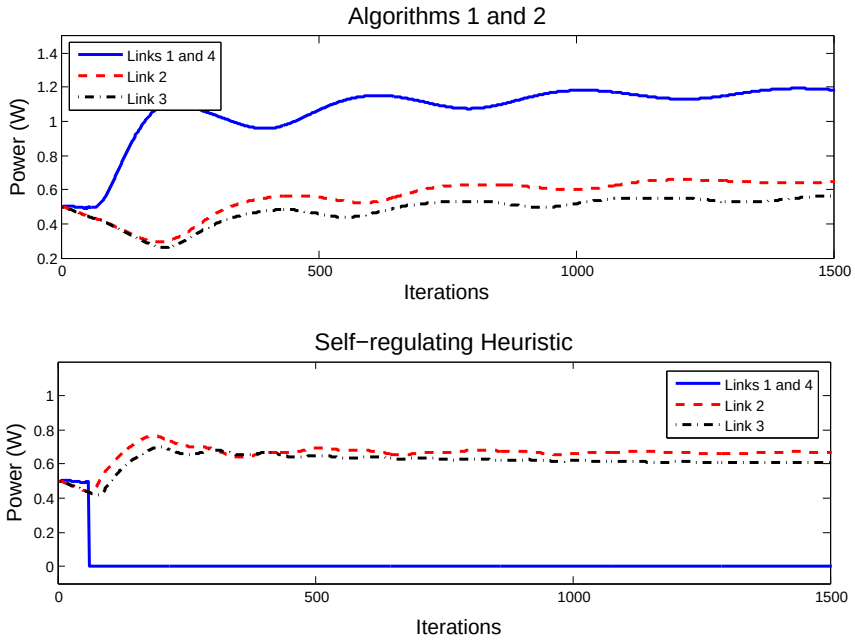

Figure 2.8.: Convergence of Transmission Power Allocation

for non-convex problems and provides a condition to assure that a distributed gradient-based algorithm converges to the optimal solution. The optimization framework is applied on an optimization problem formulation in wireless ad-hoc networks. This formulation includes a power penalty function to assure convergence and energy efficiency of the power allocation. Consequently, a distributed algorithm to solve this problem is developed and an oscillation resolving heuristic is presented to assure network stability in non-convex problems whose optimal solution can not be calculated distributedly.

The focus of the next chapters will be twofold. First, in Chapter 3, we will examine the applicability of the optimization framework to a wider range of utility functions and problem formulations. To that purpose, we will describe the motivation behind the use of a novel family of multi-sigmoidal

utility functions to model multi-tiered multimedia applications and, then, will propose means to overcome the research challenges that such utilities impose to the application of the non-convex framework presented in this chapter.

Secondly, in Chapter 4 we will work towards an alternative policy of allocating bandwidth towards a more fair resource allocation. We will show that *Utility proportional fairness* can be an efficient way to convexify the problem formulations that result from the incorporation of non-concave utilities and, moreover, offer a more fair method to allocate resources with a priority of applications that need them the most.

3. Non-convex Resource Allocation for Multi-tiered Multimedia Applications

3.1. Introduction

The end-to-end communication and resource allocation services in current communication networks are provided by Transport Layer protocols such as TCP. As shown earlier, the various TCP Algorithms proposed during the last decades have been shown to implicitly solve a resource allocation optimization problem [38] where all applications have been modeled using concave utility functions. Although this was a valid assumption in the past, the network traffic generated by modern applications has such *Quality of Service (QoS)* requirements that need to be modeled by non-concave functions. Therefore, as we showed in Chapter 2, existing resource allocation schemes provide suboptimal solutions that may significantly affect both network performance and user experience.

Network Utility Maximization (NUM) [17], contrary to the resource allocation algorithm in TCP, can distinguish between elastic and inelastic applications by choosing different utility functions for each one. This clearly highlights the important role that NUM can play towards the development

of new transport layer protocols that would optimize the allocation of resources in heterogeneous networks, where elastic and inelastic applications compete for resources. Recall the distinction between elastic and inelastic traffic described in the previous chapters. Elastic applications include file transfer (FTP), email, network management (SNMP) and Web access (HTTP), while inelasticity usually characterizes *real-time* applications such as Video Streaming, Teleconferencing, Voice over IP (VoIP), Stock Trading etc., that have some minimum requirements in throughput and/or delay. Since the seminal work of Kelly et al. [17], there have been several pieces of work that cultivated a deep understanding in the ways that optimization theory can be utilized in solving various convex resource allocation formulations in a distributed way.

The fast growing number of multimedia applications in current networks led the research community to work towards the incorporation of a more accurate modeling of the "inelasticity" of such applications in the NUM framework. The authors in [49] and [52] first discuss the properties of a single sigmoidal utility function and the implications of such a utility shape in the convergence of a gradient based algorithm to the optimal solution.

Despite the aforementioned extensions of the NUM framework, single sigmoidal utility functions may not be suitable to model many state of the art multimedia applications. Several video streaming applications used nowadays offer services at different quality levels with each level having different bit-rate requirements and offering different *Quality of Experience (QoE)* for the user. For example, assume that an online video content provider offers four distinct levels of video quality (e.g. low, medium, high, ultra high) based on the video resolution and bitrate. Each quality option represents a different level of user satisfaction. Moreover, for a specific video

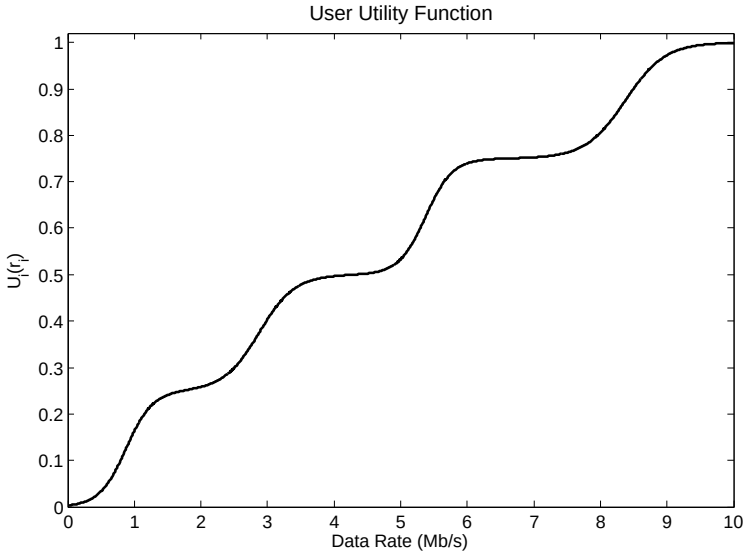

Figure 3.1.: Example of Multi-tiered Utility Functions

resolution the allocated bitrate affects user satisfaction. For example, if low resolution is chosen, the increase of bitrate above a certain level will not result in significantly better visual results since the resolution is too low for a visible improvement. Therefore, user satisfaction at this quality level is saturated and further increase can only be a result of the transition to a higher resolution profile. Such multi-tiered multimedia applications can not be modeled satisfactorily well by single sigmoidal utilities.

The most intuitive, yet very challenging, solution to this problem is the use of multi-sigmoidal utilities. Multi-sigmoidal utility functions, such as the one shown in Figure 3.1, are capable of capturing the step-like behavior of user satisfaction with respect to the various quality levels of modern video applications. *The development of appropriate multi-sigmoidal utility functions that can capture the QoS/QoE characteristics of the underlying applications and the extension of NUM framework to incorporate such utilities are the main motivation behind the work presented in this chapter.*

More specifically, this chapter attempts to provide answers to the following questions:

- Which are the properties that a multi-sigmoidal utility function must possess in order to be appropriate to model multi-tiered multimedia applications?
- Can the existing NUM framework be used to solve the resource allocation problem under the existence of such utility functions?
- What would be the implications of such a utility shape in the continuity of the optimal rate allocation function?
- What would be an appropriate mathematical formulation of a multi-sigmoidal utility function?

To the best of our knowledge this is the first work in literature that tries to provide answers to these questions. In an attempt to answer them, this chapter makes the following contributions:

- Introduces the concept of multi-sigmoidal utility function to express user experience/satisfaction in multi-tiered multimedia applications.
- Examines the incorporation of multi-sigmoidal utility functions to the existing NUM framework, gives an insight into the impact of such a choice on the continuity properties of the optimal rate allocation function and describes a detailed procedure to determine all these discontinuity points.
- Proposes an efficient heuristic algorithm in order to resolve network oscillations, caused by these discontinuities, while preserving fairness.

- Proposes a novel mathematical representation of a multi-sigmoidal utility function and provides a thorough discussion on how the function's parameters can be calibrated.

- Proposes a distributed gradient based algorithm for this specific family of multi-sigmoidal functions to solve the resulting NUM problem optimally, when possible.

The rest of the chapter is organized as follows. First, Section 3.2 discusses the properties that a multi-sigmoidal function must possess in order to be in accordance with the physical interpretation of a utility function and highlights the research challenges that arise when such utilities are applied to NUM. Section 3.3 reveals the direct connection between the utility function and the discontinuities in the rate calculation mechanism, presents a detailed procedure to determine these discontinuities and discusses the network oscillations that these discontinuities might cause during the optimization process. Consequently, Section 3.4 proposes an efficient, low complexity, distributed heuristic that allows users to resolve these oscillations when they occur. Then, Section 3.5 presents a novel mathematical representation of a multi-sigmoidal function, discusses the reasons that make it appropriate for NUM, and presents in detail an efficient approximation method to the optimal resource allocation that leads to the development of a joint primal-dual distributed algorithm. Section 3.6 presents extended simulation results of the proposed algorithms in various network topologies that illustrate their efficiency, and, finally, Section 3.7 concludes the work presented in this chapter.

3.2. Network Utility Maximization with Multi-sigmoidal Utilities

3.2.1. Properties of a Multi-sigmoidal Utility Function

The heterogeneity of the applications competing for resources in current communication networks dictates the use of application-specific utility functions to capture user satisfaction efficiently. The introduction of single-sigmoidal utility functions was the first step towards the development of a generic non-convex resource allocation framework to optimize multimedia applications but they are not always the most suitable choice for modelling real-time applications.

The need for a new family of utilities originates from the fact that most multimedia content providers (either video or audio) offer content at a number of discrete quality levels (low, medium, high etc.), each of them having different bandwidth requirements. Therefore, the gradations of user QoE according to the selected quality level must be depicted in the utility functions that model such applications. It is evident from the above that the most appropriate utility function is a multi-sigmoidal function with multiple inflection points, or else, multiple sigmoidal components. In addition, this family of functions possesses some additional properties necessary to support the physical meaning of a utility function as a user satisfaction indicator with respect to the allocated transmission rate. Therefore we are interested in functions that:

P1) take positive values in the range $[0, 1]$;

P2) are increasing functions of the transmission rate,

P3) are zero when no rate is allocated to a particular user;

P4) have a maximum rate, r_{max}, above which its value is always 1;

P5) are continuous in the range $(0, r_{max})$.

One could argue that a potential sixth property could be added as well. This describes the need that all quality levels, i.e. all concave parts, of the utility to be reachable by a NUM algorithm. In other words, a multi-sigmoidal utility can indeed model multi-tiered applications only if all distinct utility levels can be optimal selections under some conditions. While this will be explained in more detail later, in Section 3.3, it is not considered a requirement for a multi-sigmoidal utility since the exact shape of a utility function is determined by each user depending on the user's appreciation of the allocated bitrate without having in mind the operational characteristics of NUM. Moreover, this chapter provides a detailed methodology to determine which of the levels of a multi-sigmoidal utility are reachable by a NUM algorithm and which not.

The incorporation of multi-sigmoidal utilities in the existing NUM framework is not as straightforward as someone may think due to the convexity properties of such utilities. The next section describes the NUM framework in detail and discusses the research challenges that multi-sigmoidal utilities raise.

3.2.2. Network Resource Allocation with Multi-sigmoidal Utilities

The *Network Utility Maximization (NUM)* framework [17][18] expresses the bandwidth allocation in communication networks as an optimization problem under the assumption that all utilities are concave (and most commonly logarithmic) functions of rate in order to assure convergence of the dis-

tributed algorithm. This section will present a generalized NUM framework where the utilities can be multi-sigmoidal, i.e. non-concave, and discuss the research challenges that this imposes to the distributed algorithm. These challenges will be answered later in this chapter.

Consider a multi-hop network where M nodes act as sources sending streams of traffic to a set of destination nodes using a set of J links. A single node can operate as a source, destination or even as a relay node that just forwards traffic to its neighbors. It is assumed that all links in the network are wired, vector $\boldsymbol{C} = [C_1, C_2, \cdots, C_J]^T$ contains the capacity of each link[1] and vector $\boldsymbol{r} = [r_1, r_2, \cdots, r_M]^T$ includes the transmission rates of all sources. The optimization problem describing the *Network Resource Allocation (NRA)* problem is:

$$\begin{aligned} \max_{\boldsymbol{r}} \quad & \sum_{i=1}^{M} U_i(r_i) \\ \text{s. t.} \quad & \sum_{i=1}^{M} \alpha_{i,j} r_i \leq C_j, \quad \forall \text{ links } j \end{aligned} \tag{3.1}$$

where routing coefficient $\alpha_{i,j}$ is 1 if user i sends traffic through link j and 0 otherwise. We assume that the routing matrix $\boldsymbol{A}$, containing all routing coefficients $\alpha_{i,j}$, is known a priori and considered fixed throughout the optimization process. The rates r_i, $i \in [1, M]$, in $\boldsymbol{r}$ are positive variables since they represent the transmission rates of the respective source nodes.

Problem (3.1) can be solved distributedly using Duality Theory. For this purpose, we first construct its dual. The langrangian function can be written as:

$$L(\boldsymbol{r}, \boldsymbol{\lambda}) = \sum_{i=1}^{M} \left\{ U_i(r_i) - r_i \lambda^i \right\} + \sum_{j=1}^{J} \lambda_j C_j \tag{3.2}$$

[1]Links are assumed to have fixed capacity, i.e. they model wired links. The case of multi-sigmoidal utilities in wireless networks will be examined in Chapter 4.

where λ_j are the "*Lagrange* multipliers", which represent the "price" that user i has to pay in order to send each of the r_i units of traffic through link j and $\lambda^i = \sum_{j=1}^{J} \alpha_{i,j}\lambda_j$ is the aggregate price to send its traffic to the destination node. Vector $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_J]^T$ contains the *dual* optimization variables with each one of them corresponding to a constraint of the primal problem and, therefore, to a link in the network. Based on (3.2), the objective function of the *dual* problem will be $d(\boldsymbol{\lambda}) = \sup_{\boldsymbol{r}} L(\boldsymbol{r}, \boldsymbol{\lambda})$ and the resulting *dual* problem:

$$\min_{\boldsymbol{\lambda}} d(\boldsymbol{\lambda}) \qquad \text{s. t.} \qquad \boldsymbol{\lambda} \geq 0. \tag{3.3}$$

As explained in Chapter 1, problem (3.3) is a convex problem as a point-wise supremum of a family of affine functions [6]. It is clear from (3.2) that each user is trying to maximize their *Net Utility*, i.e. $NU_i(r_i) = U_i(r_i) - r_i \cdot \lambda^i$ and thus the optimal resource allocation for user i will be:

$$r_i^*(\boldsymbol{\lambda}) = \text{argmax } \{NU_i(r_i)\}. \tag{3.4}$$

Equation (3.4) can be used to calculate the optimal rate of user i for a given price vector $\boldsymbol{\lambda}$. The optimal value of the dual variables λ_j, $j \in [1, J]$, can be calculated iteratively using a gradient method, such as the *Gradient Projection*[2] [7],

$$\lambda_j(t+1) = \lambda_j(t) - s_\lambda(t)\frac{\partial L(\boldsymbol{r}, \boldsymbol{\lambda})}{\partial \lambda_j}, \tag{3.5}$$

to assure that any λ_j will take non-negative values. $s_\lambda(t)$ is the step size of the method at time t and affects the convergence speed and distance from the true optimum [7]. The partial derivative of $L(\boldsymbol{r}, \boldsymbol{\lambda})$ with respect to λ_j

[2] A projection method to the feasible set of values is necessary in this case to assure that all λ_j, $j \in [1, J]$, take non-negative values.

is:

$$\frac{\partial L(\boldsymbol{r}, \boldsymbol{\lambda})}{\partial \lambda_j} = C_j - \sum_{i=1}^{M} \alpha_{i,j} r_i. \tag{3.6}$$

Equations (3.4) and (3.5) constitute a joint primal-dual distributed algorithm of NUM, which can converge to an optimal solution, even in the case of non-concave utilities (such as single-sigmoidal), as long as (3.4) is continuous around the optimal price vector $\boldsymbol{\lambda}^*$ [52]-[48]. In fact, Theorem 1 in Chapter 2 shows that any gradient based algorithm will converge to the optimal solution for any non-convex optimization problem as long as the primal variables are continuous functions of the dual variables around their optimal values. Regarding equations (3.4) and (3.5), the following research questions emerge regarding the use of multi-sigmoidal utilities in NUM:

- Is (3.4) a continuous function of the dual variables?
- If not, is it possible to develop an analytical methodology to identify the points of discontinuity?
- Is it possible to calculate or approximate a closed form solution for (3.4) in the case of multi-sigmoidal utilities?
- Is there a fair and efficient method to resolve possible network oscillations due to the discontinuity of (3.4)?

Taking into account that (3.4) is discontinuous at one point for single-sigmoidal utilities [49], we expect to have at least one point of discontinuity at the multi-sigmoidal case as well. Answers to these questions will be provided in the rest of this chapter.

3.3. The Price-based Rate Allocation Function

The price-based rate allocation function (3.4) is the solution of an optimization problem that calculates the rate to maximize the *net utility* of user i for a specific price vector $\boldsymbol{\lambda}$. This section will, initially, discuss the difficulties in calculating a closed form solution for non-concave utilities, and then, will examine the important role that its continuity plays in the convergence of the distributed algorithm.

3.3.1. Calculation

According to the NUM framework, all users in the network are acting selfishly and try to optimize their individual *net utility*. In other words, user i tries to solve the following optimization problem at each time instant and for the current price vector $\boldsymbol{\lambda}$:

$$\begin{aligned} \max_{r_i} \quad & U_i(r_i) - r_i \cdot \lambda^i \\ \text{s. t.} \quad & r_i \geq 0, \end{aligned} \tag{3.7}$$

where $\lambda^i = \sum_{j=1}^{J} \alpha_{i,j}\lambda_j$. In essence, this is the optimization problem that must be solved in (3.4). The optimal solution of Problem (3.7) is also the optimal rate for user i for Problem (3.1). This rate is at a point where the derivative of the objective function diminishes [7], which leads to:

$$r_i^* \left(\boldsymbol{\lambda}\right) = U_i' \left(\lambda^i\right)^{-1}, \tag{3.8}$$

where $U_i' \left(\cdot\right)^{-1}$ is the inverse first derivative function. The calculation of the inverse of the first derivative is possible for concave functions, such as the widely used $U_i \left(r_i\right) = \log r_i$ function, used to model applications such as FTP, HTTP, etc. In addition, for concave utilities it is also possible to use

the following gradient based iterative equation

$$r_i(t+1) = r_i(t) - s_r(t)\frac{\partial L(\boldsymbol{r}, \boldsymbol{\lambda})}{\partial r_i}, \tag{3.9}$$

where $s_r(t)$ is the step size of the update at time t, much like $s_\lambda(t)$ in (3.5). However, the calculation of the inverse derivative of non-concave utilities is not possible because their derivatives are not one-to-one functions. In essence, the fact that $U_i'(\cdot)$ is not a one-to-one function implies that there might be more than one optimal rates for a single aggregate price λ^i. Moreover, an iterative equation such as (3.9) is impractical in many cases, since it may converge to local rather than global optima. This difficulty to calculate a general closed form solution for Problem (3.7) highlights the need for developing methods to approximate $r_i^*(\boldsymbol{\lambda})$ efficiently for specific non-concave utility shapes. With this motivation, Section 3.5 will present analytically an efficient approximation technique of the optimal rate for a novel mathematical representation of multi-sigmoidal utility functions. In essence, Algorithm 4 presented in Section 3.5 will provide a detailed procedure to solve the optimization problem in (3.4) for a specific family of multi-sigmoidal utilities.

The continuity properties of (3.4) are important for the convergence of any gradient based algorithm to the optimal resource rate allocation. Therefore the next section will shed some light on its discontinuity points and their connection to the exact shape of the utility function.

3.3.2. Discontinuity

So far, we know that the price based rate function $r_i^*(\boldsymbol{\lambda})$ is continuous for all price vectors if the utility is either concave or convex function of rates while

it is discontinuous at only one point for single-sigmoidal utilities. Continuity of $r_i^*(\boldsymbol{\lambda})$ is also important for convergence with multi-sigmoidal utilities and therefore any discontinuity points must be identified. Equation (3.8) shows that $r_i^*(\boldsymbol{\lambda})$ is in essence a function of the aggregate price per unit of traffic and does not depend on the individual values of λ_j, $j \in [1, J]$. Therefore, we will also refer to $r_i^*(\boldsymbol{\lambda})$ as $r_i^*\left(\lambda^i\right)$, where λ^i is the aggregate price for user i.

It turns out that the shape of a utility function determines the discontinuity points of the rate allocation function as well, and that the discontinuity points correspond to jumps from one concave region to another or from one concave region to zero. Moreover, we need to highlight the difference between a *candidate* discontinuity point and an *actual* discontinuity point. There are a number of candidate discontinuity points that may or may not appear as actual discontinuities of $r_i^*\left(\lambda^i\right)$. The methodology to identify these points involves the use of tangent lines to the utility function $U_i(r_i)$. Initially, we draw a tangent line $y = \alpha r_i + \beta$ that osculates the utility function at two or more points. Let r_i^n, with $n = 1, 2, ..., N$, be the rates at which the tangent line y osculates the utility function. We also name the touching points in ascending order, i.e. $r_i^1 < r_i^2 < \cdots < r_i^N$. Moreover, it is assumed that $U_i(r_i) \leq y$, which implies that the tangent line is graphically always above the utility function and therefore points r_i^n are all in the concave parts of the utility, i.e.

$$\frac{\partial^2 U_i(r_i)}{\left(\partial r_i^n\right)^2} < 0, \text{ at all points } r_i^n,\ n = 1, 2, \ldots, N. \tag{3.10}$$

In multi-sigmoidal utilities, a tangent line such as y can osculate the utility function at most at $N = K$ points, where K is the number of inflection

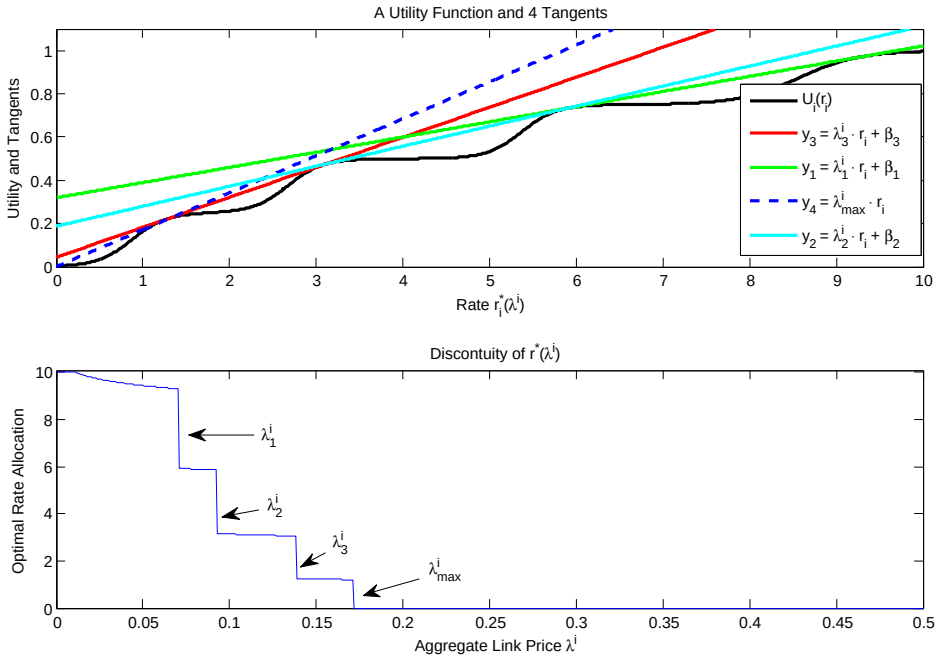

Figure 3.2.: Example of a multi-sigmoidal utility with four discontinuity points

points in the utility shape, and there can be at most $\frac{K(K-1)}{2}$ distinct tangents, in the case where each one of them osculates the utility function at exactly two points. Using the example of tangent y we can prove that the candidate discontinuity points are aggregate prices equal to the slopes of these tangent lines. To show that, we initially examine the properties of the points r_i^n, $n = 1, 2, \ldots, N$, where line y osculates the utility function. The next theorem proves that these points are all optimal rates of Problem (3.7) for aggregate price equal to the slope of line y, i.e. α.

Theorem 3. *If $\lambda^i = \alpha$ then the rates r_i^n, $n = 1, 2, \ldots, N$ are all globally optimal rates for user i and aggregate price λ^i.*

Proof. As explained in Section 3.3, user i is trying to maximize its Net Utility $NU_i(r_i)$ according to Problem (3.7).

The *Lagrangian* function of Problem $\mathbf{\Pi^{i}_{NU}}(\boldsymbol{\lambda})$ is:

$$L_i^{\boldsymbol{NU}}(r_i) = U_i(r_i) - \lambda^i \cdot r_i, \tag{3.11}$$

and its first derivative

$$\frac{dL_i^{\boldsymbol{NU}}(r_i)}{dr_i} = \frac{dU_i(r_i)}{dr_i} - \lambda^i. \tag{3.12}$$

Since, all points r_i^n, $n = 1, 2, \ldots, N$, belong to the same line, the derivative of the utility function at those points will have the same value. In other words,

$$\frac{dU_i(r_i)}{dr_i} = \alpha, \text{ for all points } r_i^n,\ n = 1, 2, \ldots, N. \tag{3.13}$$

Based on (3.12), (3.13) and the fact that $\lambda^i = \alpha$, we find that:

$$\frac{dL_i^{\boldsymbol{NU}}(r_i)}{dr_i} = 0, \text{ for all points } r_i^n,\ n = 1, 2, \ldots, N, \tag{3.14}$$

which is the *First Order Sufficient Condition* for optimality. Then, from (3.10) and (3.14) we find that points r_i^n, $n = 1, 2, \ldots, N$ satisfy the *Second Order Sufficient Condition* for optimality as well and therefore they are all locally optimal points.

In addition, since rates r_i^n, $n = 1, 2, \ldots, N$ are all points of the tangent line $y = \alpha \cdot r_i + \beta$, we know that:

$$\begin{aligned} &U_i(r_i^n) = \alpha \cdot r_i^n + \beta \Leftrightarrow \\ &U_i(r_i^n) - \lambda^i \cdot r_i^n = \beta \Leftrightarrow \\ &NU_i(r_i^n) = \beta, \text{ for } n = 1, 2, \ldots, N. \end{aligned} \tag{3.15}$$

Hence, all locally optimal rates r_i^n, $n = 1, 2, \ldots, N$ yield the same value at

the objective function of Problem (3.7) equal to β.

Global optimality of these points implies that there is no other rate that leads to higher value of the objective function, i.e. there is not any other rate that results in higher *Net utility* for user i. We will prove this part of the theorem by contradiction.

Assume that there is a rate r' that has higher *net utility* for user i than the locally optimal rates r_i^n, $n = 1, 2, \ldots, N$ for aggregate price $\lambda^i = \alpha$. In other words,

$$NU_i\left(r'\right) = \beta' > \beta. \tag{3.16}$$

Then, for this point r' we have that:

$$\begin{aligned} &NU_i\left(r'\right) = \beta' \Leftrightarrow U_i\left(r'\right) - \lambda^i \cdot r' = \beta' \Leftrightarrow \\ &U_i\left(r'\right) = \alpha \cdot r' + \beta'. \end{aligned} \tag{3.17}$$

In other words, rate r' belongs to the line $y' = \alpha \cdot r_i + \beta'$ which is also a tangent at the utility function at point r'. This implies that there is a second tangent at the utility function, other than $y = \alpha \cdot r_i + \beta$ with $\beta' > \beta$ that is also tangent at a concave point of it. However, this means that two different perpendicular lines are tangent to the same function which can not be true because, since $\beta' > \beta$, line $y = \alpha \cdot r_i + \beta$ can not be a tangent of $U_i\left(r_i\right)$ as well. This contradicts to our definition of line y. Therefore, there is no other point r' that yields higher net utility to user i for aggregate price $\lambda^i = \alpha$ and hence points r_i^n, $n = 1, 2, \ldots, N$ are globally optimal rates. $\square$

Theorem 3 shows that the price based rate function $r_i^*\left(\lambda^i\right)$ has multiple values for aggregate price λ^i equal to the slope of the tangent y and the multiplicity of the function at that point is equal to the number of points

N that the slope osculates the utility function. Examining the properties of $r_i^*\left(\lambda^i\right)$ around point $\lambda^i = \alpha$, it is possible to prove the following theorems in order to justify the discontinuity and monotonicity properties of $r_i^*\left(\lambda^i\right)$.

Theorem 4. *If $\lambda^i = \alpha + \delta$, where δ is a very small positive constant, then the globally optimal rate $r_i^*\left(\lambda^i\right)$ is smaller than the smallest optimal rate for $\lambda^i = \alpha$, i.e $r_i^*\left(\lambda^i\right) < r_i^1$.*

Proof. According to the *First Order Necessary Condition*, at the optimal rate r_i^*, $U_i'\left(r_i^*\right) = \lambda^i$. So, the candidate optimal points will be points where the family of tangent lines, $y = \lambda^i \cdot r_i + \beta$, for various values of β and $\lambda^i = \alpha + \delta$, touch the utility function. Assume that there are P such tangents. Then, it is easy to see that tangent p, with $p = 1, 2, \ldots, P$ and $P \leq K$, touches the utility function at exactly one point, let $r_{i,p}$. Without loss of generality we can assume that $r_{i,1} < r_{i,2} < \cdots < r_{i,P}$. Then, we have that:

$$U_i\left(r_{i,p}\right) = \lambda^i \cdot r_{i,p} + \beta \Rightarrow U_i\left(r_{i,p}\right) - \lambda^i \cdot r_{i,p} = \beta \Rightarrow$$

$$NU_i\left(r_{i,p}\right) = \beta. \tag{3.18}$$

Hence the rate that corresponds to the tangent with the largest value of β is the globally optimal rate. It is easy to verify graphically that the minimum of all $r_{i,p}$ points, i.e. $r_{i,1}$, is the one that corresponds to the tangent line with the larger constant β and using the concavity properties of the utility function around the candidate optimal rates, it is also easy to conclude $r_{i,1} < r_i^1$, where r_i^1 is the smallest optimal rate for aggregate price $\lambda^i = \alpha$ and hence $r_i^* < r_i^1$. □

Theorem 5. *If $\lambda_i = \alpha - \delta$, where δ is a very small positive constant, then the globally optimal rate r_i^* is larger than the largest optimal rate for $\lambda_i = \alpha$,*

i.e $r_i^* > r_i^N$.

Proof. Working in the same way as for Theorem 4, we need to find the corresponding rate for the tangent line with the largest constant β and $\lambda^i = \alpha - \delta$. If the candidate points are again denoted by $r_{i,p}$, $p = 1, 2, \cdots, P$, with $r_{i,p} < r_{i,p} < \cdots < r_{i,P}$, it easy to understand that the largest, i.e. $r_{i,P}$, is the rate that corresponds to the largest *Net Utility* and due to the concavity properties of the utility around the candidate rates $r_{i,P} > r_i^N$. This proves that the optimal rate r_i^* will be larger than the largest optimal value at $\lambda_i = \alpha$, i.e. $r_i^* > r_i^N$. □

In addition, regarding the monotonicity of $r_i^*\left(\lambda^i\right)$ with respect to λ^i, it is possible to prove the following theorem.

Theorem 6. *The optimal rate function of user i,* $r_i^*\left(\lambda^i\right)$*, is a decreasing function of* λ^i*.*

Proof. Let λ_1^i and λ_2^i be two aggregate prices with $0 \leq \lambda_1^i < \lambda_2^i \leq \lambda_{max}^i$ and let $x_1 = r_i^*\left(\lambda_1^i\right)$ and $x_2 = r_i^*\left(\lambda_2^i\right)$ the optimal rates for these aggregate prices respectively. User i is trying to optimize Problem (3.7) and therefore the *First Order Necessary Condition* must hold for the optimal rates x_1 and x_2. In other words, if $NU_i'(\cdot)$ and $U_i'(\cdot)$ are the derivatives of the *net utility* (i.e. the objective function of Problem (3.7)) and the *utility* function of user i respectively

$$\begin{array}{l} NU_i'(x_1) = 0 \\ NU_i'(x_2) = 0 \end{array} \Rightarrow \quad \begin{array}{l} U_i'(x_1) - \lambda_1^i = 0 \\ U_i'(x_2) - \lambda_2^i = 0 \end{array} \Rightarrow$$

$$U_i'(x_1) = \lambda_1^i < \lambda_2^i = U_i'(x_2) \Rightarrow$$

$$U_i'(x_1) < U_i'(x_2). \tag{3.19}$$

According to the *Second Order Necessary Conditions*, the optimal rates will always be in concave regions of the multi-sigmoidal utility function. Therefore, if points x_1 and x_2 are in the same concave region, then from (3.19) we conclude that $x_2 < x_1$. Moreover, even if points x_1 and x_2 are in different concave regions, theorems 4 and 5 imply that $r_i^* \left(\lambda^i\right)$ has decreasing "jump" discontinuities, i.e $x_2 < x_1$. Therefore, $r_i^* \left(\lambda^i\right)$ is a decreasing function of aggregate price λ^i. □

Apart from the discontinuity of $r_i^* \left(\lambda^i\right)$ around the points determined by the tangents at the utility function, these theorems imply that rates in the range $\left(r_i^1, r_i^N\right)$, excluding points r_i^n, $n = 2, \ldots, N-1$, can never be globally optimal rates and therefore the price-based rate function will "jump" from r_i^N to r_i^1. Another direct result from Theorem 6 is that there will be a maximum value for λ^i, let λ_{max}^i, above which the optimal rate will be zero. In other words, $r_i^* \left(\lambda^i\right)$ has a positive value for $0 \leq \lambda^i \leq \lambda_{max}^i$ and is zero for aggregate prices $\lambda^i \geq \lambda_{max}^i$. This maximum non-zero aggregate price λ_{max}^i is called *maximum willingness to pay* for user i and is a discontinuity point of $r_i^* \left(\lambda^i\right)$ for single-sigmoidal utilities [52][47]. The methodology to calculate λ_{max}^i in the multi-sigmoidal case shows that λ_{max}^i is a discontinuity point of $r_i^* \left(\lambda^i\right)$ for multi-sigmoidal utilities as well.

To calculate λ_{max}^i, we start by the fact that an aggregate price $\lambda^i = \lambda_{max}^i$ corresponds to two distinct values of rate, a positive one, let $\hat{r_i}$, and a zero rate. For these two rates, the net utility must be equal, i.e.

$$NU_i\left(\hat{r_i}\right) = NU_i\left(0\right) \Leftrightarrow$$

$$U_i\left(\hat{r_i}\right) - \lambda_{max}^i \cdot \hat{r_i} = 0. \tag{3.20}$$

Since $\hat{r_i}$ is an optimal rate for aggregate price λ_{max}^i, the *First Order Opti-*

mality Conditions must be met. In other words, the first derivative of the net utility function at point $\hat{r}_i$, $NU_i'(\hat{r}_i)$, must be zero, which leads to

$$U_i'(\hat{r}_i) = \lambda_{max}^i. \tag{3.21}$$

Substituting to (3.20), we get the differential equation:

$$U_i(\hat{r}_i) - U_i'(\hat{r}_i) \cdot \hat{r}_i = 0 \tag{3.22}$$

where $U_i'(\cdot)$ is the first derivative of the utility function. This differential equation clearly has more than one solutions, a zero rate solution and one or more positive ones, which shows that $r_i^*\left(\lambda^i\right)$ has multiple values for aggregate price λ_{max}^i. By solving (3.22) and calculating $\hat{r}_i$, it is also possible to calculate λ_{max}^i using (3.21) by selecting the largest of the positive solutions, which is a discontinuity point of $r_i^*\left(\lambda^i\right)$.

It is evident from the above that every tangent at two or more points of the utility function represents a candidate discontinuity point of the price based function $r_i(\boldsymbol{\lambda})$. Each one of these points represents a "jump" from one hyperbolic tangent component to another, while the discontinuity point around λ_{max}^i represents a "jump" from a hyperbolic tangent component to zero rate. The latter point will always appear in the rate function but the rest depend on their relative value compared to λ_{max}^i. For example, if λ_{max}^i is smaller than all the other candidate discontinuity aggregate prices, then none of them will appear and there will be only one discontinuity point. The maximum number of discontinuity points are K, as many as the inflection points of the utility. This can happen if there are $K-1$ distinct tangent lines, each one touching the utility at two points that belong to two consecutive hyperbolic tangent components. The K^{th} discontinuity point is

for $\lambda^i = \lambda^i_{max}$ when "jumping" from the first concave region to zero rate, and could graphically be represented by a tangent line that passes from point $(0, 0)$ and osculates the utility function at its first hyperbolic tangent component.

Figure 3.2 shows an example of a utility function that has four discontinuity points. The top sub-figure shows the utility function and the four tangent lines responsible for the four discontinuity points while the bottom one shows the optimal rate $r_i^*(\boldsymbol{\lambda})$ calculated exhaustively with the discontinuity points clearly shown. This figure illustrates the connection between the shape of the utility function and the discontinuity points of the price-based rate function $r_i^*(\boldsymbol{\lambda})$. Moreover, it evidently verifies that $r_i^*(\boldsymbol{\lambda})$ is a decreasing function of the aggregate price since it consists of decreasing continuous parts and decreasing jump discontinuity points.

Commenting on the feasibility of all K sigmoidal components to be selected as optimal choices, it is evident that this is possible only under the existence of K distinct discontinuity points[3]. In any other case, there will be at least one sigmoidal component that is unreachable during NUM, which is one of the shortcomings of the NUM framework. This observation leads to the interesting conclusion that the fully reachable multi-sigmoidal utilities are those with the maximum possible number of discontinuity points. Moreover, we can form the following theorem:

Theorem 7. *A multi-sigmoidal utility will have all levels reachable, and hence will have the maximum discontinuity points, iff the following condi-*

[3]Such utility function is hereafter referred as fully reachable multi-sigmoidal utility.

tions hold:

$$\begin{aligned} &(1)\ \lambda^i_{k,k-1} < \lambda^i_{k,j}, \quad \forall j \in [1,\ldots,k-2], \quad k \in [3,\ldots,K] \\ &(2)\ \lambda^i_{k,k-1} < \lambda^i_{max}, \quad \forall k \in [2,\ldots,K], \end{aligned} \tag{3.23}$$

where $\lambda^i_{k,l}$ is the slope of the tangent that osculates the utility of user i at the k^{th} and l^{th} sigmoidal component.

Proof. To prove its sufficiency, we assume a utility function for which conditions (1) and (2) hold. This implies that as the aggregate price λ^i of that user increases, it will first reach the discontinuity point $\lambda^i_{K,K-1}$ and the optimal rate will drop to the next concave region $K-1$. As the aggregate price increases further, $\lambda^i_{k,k-1}$ will always be reached before any discontinuity point to non consecutive regions and, therefore, optimal rate will move only to consecutive ones until it exceeds the user's maximum willingness to pay, λ^i_{max}, and becomes zero. This leads to the conclusion that the utility function is fully reachable.

To show that it is also a necessary condition, we assume first that condition (1) does not hold for a specific utility. This implies that there is an index m, with $m \in [1,\ldots,k-2]$, for which $\lambda^i_{k,m} < \lambda^i_{k,k-1}$. Therefore, when the aggregate price for user i exceeds $\lambda^i_{k,m}$, user i will drop from region k to the non consecutive region m. Hence there will be at least $k-m-1$ unreachable concave regions in the utility function and, therefore, $r^*_i\left(\lambda^i\right)$ will have at most $K-(k-m-1)$ discontinuity points. Now, assume that condition (2) does not hold. In that case, there is an index m, with $m \in [2,\ldots,k]$, for which $\lambda^i_{m,m-1} > \lambda^i_{max}$. This implies that when the aggregate price λ^i for user i reaches or exceeds λ^i_{max}, the optimal rate will drop to zero and the concave regions 1 to $m-1$ will be unreachable. Moreover, there can be at most $K-m+1$ discontinuity points in $r^*_i\left(\lambda^i\right)$. The aforementioned argu-

ments lead to the conclusion that conditions (1) and (2) are also necessary conditions to have the a fully reachable utility function. □

Determining a detailed procedure to design such multi-sigmoidal utilities is beyond the scope of this paper but always remains within our future research plans.

For an arbitrary multi-sigmoidal utility function it is possible to determine exactly the aggregate prices for which $r_i^*(\boldsymbol{\lambda})$ is discontinuous. The calculation of these points involves the calculation of λ_{max}^i and all candidate discontinuity points. The easiest way to calculate the candidate discontinuity rates is by assuming that each tangent osculates the utility function at exactly two points, let p_1 and p_2, and then to calculate the slope of this tangent. More specifically, for points p_1 and p_2 it is known that:

$$U_i'(p_1) = U_i'(p_2) \tag{3.24}$$

$$U_i'(p_2) = \alpha_y, \tag{3.25}$$

where α_y is the slope of the tangent line. Substituting α_y we reach the following system of equations that can be used to calculate points p_1 and p_2:

$$U_i'(p_1) - U_i'(p_2) = 0 \tag{3.26}$$

$$U_i'(p_2) - \frac{U_i(p_1) - U_i(p_2)}{p_1 - p_2} = 0. \tag{3.27}$$

After calculating p_1 and p_2, it is possible to calculate the slope of the tangent line, i.e. the aggregate price that is a candidate discontinuity point using:

$$\lambda_c^i = \frac{U_i(p_1) - U_i(p_2)}{p_1 - p_2}. \tag{3.28}$$

After calculating all the candidate discontinuity points by restricting the range of point p_1 and p_2 within all possible sigmoidal components, we create the symmetric matrix $\boldsymbol{S}^i$ of size $K \times K$, where $S^i(s_1, s_2)$ represents the slope of the tangent that osculates the ${s_1}^{th}$ and ${s_2}^{th}$ concave region of the utility. By convention, we assume that the elements of the main diagonal of matrix $\boldsymbol{S}^i$ contain some very large positive value. Consequently, Algorithm 3 can be used to determine which of these candidate discontinuity points will actually appear in $r_i(\boldsymbol{\lambda})$. Note that $S^i(ctr_1, 1 : ctr_1)$ denotes the first ctr_1 elements of the ${ctr_1}^{th}$ row of matrix $\boldsymbol{S}^i$. The resulting vector $\boldsymbol{disc}$ contains the discontinuity points of $r_i^*(\boldsymbol{\lambda})$. Algorithm 3 is an iterative algorithm that depends only on the choice of the utility function of each source node and therefore can be run independently by each node in order to determine the discontinuity points of its price based rate allocation function $r_i^*(\boldsymbol{\lambda})$. Note that in case that one of the tangents osculates the utility function at more than two points, then two or more elements of matrix $\boldsymbol{S}^i$ will be equal and the discontinuity point will appear in vector $\boldsymbol{disc}$ as a multiple discontinuity step for this aggregate price.

3.3.3. Oscillations

Equations (3.4) and (3.5) constitute a joint primal-dual distributed algorithm of NUM, which can converge to an optimal solution, even in the case of non-concave utilities, as long as (3.4) is continuous around the optimal price vector $\boldsymbol{\lambda}^*$. For instance, even though the optimal rate cannot be calculated for the general case, for the reasons explained earlier, it is possible to be approximated efficiently for a specific family of multi-sigmoidal functions presented in Section 3.5. The convergence of (3.5) to the optimal solution of the dual problem relies on the selection of the step size $s_\lambda(t)$ at each

Algorithm 3 – Calculation of discontinuity points of $r_i^*(\boldsymbol{\lambda})$

$ctr_1 = K$;
$ctr_2 = 1$;
Calculate λ_{max}^i using (3.21) and (3.22);
Calculate matrix $\boldsymbol{S}^i$ by solving the system of equations (3.26) and (3.27);
while true **do**
  $index = \text{argmin}\left\{S^i\left(ctr_1, 1 : ctr_1\right)\right\}$;
  $\lambda_{tmp}^i = \min\left\{S^i\left(ctr_1, 1 : ctr_1\right)\right\}$;
  if $\lambda_{max}^i < \lambda_{tmp}^i$ **then**
    break;
  else
    $disc\left(ctr_2\right) = \lambda_{tmp}^i$;
    $ctr_1 = index$;
  end if
  $ctr_2 = ctr_2 + 1$;
end while
$disc\left(ctr_2\right) = \lambda_{max}^i$;

iteration t. Ref. [7] presents various methods for determining constant or variable step sizes. In general, a method with diminishing step sizes, where $s_\lambda\left(\tau + 1\right) < s_\lambda\left(\tau\right)$, at any time $t = \tau$, and values that converge to zero, are suitable for most cases to converge to the optimal solution. Of course, the trade-off between convergence speed and proximity to the optimal solution should be taken into account in all practical applications and it is generally true that large step sizes accelerate the algorithm's convergence but increase the distance from the optimal solution.

The discontinuity points calculated by Algorithm 3 also play an important role in the convergence of the optimization method comprised of equations (3.4) and (3.5), as specified by the condition proved by Theorem 1 in Chapter 2. Specifically, the phenomenon of oscillation occurs when the optimal rate function of a specific user is a discontinuous function of the aggregate price λ^i and the optimal price vector $\boldsymbol{\lambda}^*$ leads to an aggregate price (for that specific user) equal to the discontinuity point. Moreover, as described in Chapter 1, [49] provides specific conditions regarding the aggregate in-

coming rate at a link that can lead to oscillations. This is the only case that the algorithm will not converge to the optimal solution and is necessary to apply an oscillation resolving technique such as the *Oscillation Resolving Heuristic - ORH* presented in the next section.

3.4. Resolving User Oscillations

In case the optimal price vector $\boldsymbol{\lambda}$ leads to one of the discontinuity points of some user, which can be calculated by Algorithm 3, then this user will oscillate between multiple rates and the algorithm will not converge to any solution.

As explained in Chapter 2, a user oscillation occurs when the user transmits at an excessive data rate in an iteration of the optimization process, and then in the next iteration, the user transmits at an exceedingly low rate. Moreover, even though the notion of oscillation refers to a specific user, a user oscillation affects other users as well and therefore leads to a network oscillation.

There are in general two approaches in order to resolve this user oscillation issue; an admission control mechanism, and a fixed rate allocation method. The former removes the oscillating users from the optimization process without allocating any rate to them [46], while the latter allocates a positive rate, let r_i^{osc}, to each oscillating user i before removing them from the optimization. Allocating some rate to oscillating users leads towards more fair resource allocations compared to an admission control technique since all users have access to the network resources.

The oscillation rates of user i are in fact very close to the optimal rates for the aggregate price λ^i for which the oscillation happens. More specifically,

if the optimal price vector $\boldsymbol{\lambda}^*$ leads to an aggregate price λ^i for user i, which is a discontinuity point as well, the gradient algorithm will move between aggregate prices $\lambda^i + \epsilon_1$ and $\lambda^i - \epsilon_2$, where ϵ_1 and ϵ_2 are small positive constants. However, as theorems 4 and 5 show, the former will lead to a rate smaller than (but very close to) the smallest touching point of the respective tangent and the latter to a rate larger than (but again very close to) the largest touching point. These rate values will lead to either underutilization or overutilization of links respectively and hence to network oscillations. It is therefore, evident that an efficient oscillation resolving heuristic should allocate a rate within the oscillating range. Based on this observation, we propose the *Oscillation Resolving Heuristic (ORH)*, an enhanced version of the heuristic proposed in Chapter 2, to assure the convergence of the gradient based distributed algorithm.

The Oscillation Resolving Heuristic (ORH) allocates a fixed non-zero rate to oscillating users and removes them from the rest of the optimization process, which continues for the remaining users in the network. The allocated rate r_i^{osc} to oscillating user i is equal to the smallest touching point of the tangent, r_i^1, with slope equal to the aggregate price λ^i for which the oscillation happens. In other words, user i is allocated rate equal to the smallest of the rates that it oscillates at[4]. This approach has several advantages over other methods in the literature. First, no users are restricted from accessing network resources contrary to admission control approaches such as [46]. Moreover, the allocated rate $r_i^{osc} = r_i^1$ satisfies the *Necessary Conditions* for optimality since it is one of the optimal rates for aggregate price λ^i. Finally, even though all touching points of the tangent with slope λ^i are optimal rates, by selecting the smallest of them we assure that the rest of

[4]The smallest oscillation rate is at most ϵ_{osc} away from r_i^1, which corresponds to aggregate price $\lambda^i + \epsilon_1$, where both ϵ_1 and ϵ_{osc} are small positive constants.

the users in the network will compete for largest amount of resources, thus leading to higher total network utility.

The implementation of ORH is very simple and the algorithm runs independently for each source node and requires a simple oscillation detection mechanism. In addition, there is no need for any centralized coordination, thus preserving the distributed nature of the algorithm consisted of (3.4) and (3.5). Once user i detects an oscillation, it starts transmitting at the smallest of the oscillation rates, instead of evaluating (3.4), while ignoring the aggregate price included in the acknowledge packets coming back from the destination node. All links continue updating their link prices according to (3.5).

The Oscillation Resolving Heuristic (ORH) does not represent a complete solution for solving Problem (3.1). In fact, equations (3.4) and (3.5), and Algorithms 4 and 5 for the specific multi-sigmoidal utilities presented later, are responsible for solving Problem (3.1) iteratively, while the ORH is merely part of the process for resolving an oscillation that might occur during the iterative optimization process. In addition, the use of ORH does not affect the convergence properties of the algorithm for the following reason. Assuming that there are initially M users in the network, if the heuristic is evoked to prevent oscillations for one of them, the optimization process will continue for the remaining $M-1$ users following the convergence properties of a gradient-based optimization algorithm [7]. This process can be repeated as long as there are oscillating users in the network. At the end, either no more oscillations will occur and the algorithm will converge to the optimal solution or, if oscillations keep occurring, we will end up with the trivial case of only 1 user.

The ORH leads towards more fair resource allocations and higher utility

for practical applications compared to an admission control mechanism such as the self-regulating heuristic proposed in [46]. The heuristic in [46] lacks fairness by unnecessarily preventing users from obtaining network resources, while the ORH follows a different approach and allocates resources to as many users as possible. Simulation comparison of ORH and the heuristic proposed in [46] also shows that the ORH can lead to significantly higher utility.

3.5. A Novel Multi-sigmoidal Function and its Application to NUM

Solving the non-concave maximization problem (3.7) in the general case is not possible for the reasons explained in Section 3.3. It is, however, possible to derive an efficient approximation of the closed form solution by exploiting the special structure of specific utility functions.

3.5.1. A Hyperbolic Tangent Based Utility Function

Based on the desired properties of a multi-sigmoidal utility, presented in Section 3.2, we propose the use of the following family of multi-sigmoidal functions:

$$U(r) = \frac{1}{2K}\left\{\sum_{k=1}^{K}\tanh\left(\frac{r-c_k}{b_k}\right)+K\right\}, \tag{3.29}$$

where r is the transmission rate, c_k is the k^{th} inflection point, with $c_1 > c_2 > \cdots > c_K$, and b_k is a positive design parameter that determines the steepness of the k^{th} component of the multi-sigmoidal function. K is the number of single sigmoidal components consisting the multi-sigmoidal function, each one of them having a single inflection point, which is the point where the

second derivative changes sign, from positive to negative sign. For example, the multi-sigmoidal function of Figure 3.1 consists of four hyperbolic tangent components. It is evident from (3.29) that a multi-sigmoidal utility can be characterized by two vectors; the *inflection vector* $\boldsymbol{c} = [c_1, c_2, \ldots, c_K]^T$ and the *steepness vector* $\boldsymbol{b} = [b_1, b_2, \ldots, b_K]^T$.

Hyperbolic tangent functions have been extensively used in neural networks research area [81] but their convenient properties make them also applicable within the context of multi-tiered multimedia applications for the following reasons:

- Hyperbolic tangent functions possess the five properties described in Section 3.2.
- They can be combined together to create multi-sigmoidal shapes of arbitrary number of rate levels. For example, the utility function in (3.29) is consisted of K hyperbolic tangent components.
- They can be calibrated using the inflection vector $\boldsymbol{c}$ and the steepness vector $\boldsymbol{b}$ to achieve the desired shape.
- Their first derivative can be easily inverted to calculate the optimal rate allocation for a specific price vector.

The aforementioned advantages of hyperbolic tangent functions will be discussed in detail in the remainder of this chapter.

The hyperbolic tangent function, $tanh\,(x)$, is a symmetric, continuous (property P5), differentiable and increasing (property P2) function, which is centered around its inflection point at $r = 0$ and has two horizontal asymptotes, the lines[5] $y = -1$ and $y = 1$. Each tangent component can

[5]We denote the values along the vertical and horizontal axes with y and x respectively.

be scaled and shifted appropriately so that the resulting utility has values within the range $[0, 1]$. More specifically, the center of each hyperbolic tangent component in (3.29) has been shifted around the respective inflection point c_k, whereas the addition of K and the multiplication with $\frac{1}{2K}$ restricts the utility's range. The resulting multi-sigmoidal function has horizontal asymptotes the lines $y = 0$ and $y = 1$ (property P1). Note that inflection points c_k can be used as design parameters to create the step-like behaviour of the utility around the rate values of each application quality level.

Parameters b_k, $k = 1, \ldots, K$, can be used to calibrate the steepness of the respective tangent components. In general, larger values for b_k lead to smoother shapes. In particular, they can be used to bring $U(0)$ and $U(r_{max})$ as close to the bounds (0 and 1 respectively) as necessary, where r_{max} is the maximum rate above which the utility is equal to 1. Regarding its physical meaning, r_{max} can be considered as the maximum transmission rate of the source. Specifically, for the first case of $U(0)$, equation (3.29), for $r_i = 0$ becomes:

$$\sum_{k=1}^{K} \tanh\left(-\frac{c_k}{b_k}\right) \approx -K. \tag{3.30}$$

Moreover, since each tangent component is bounded within the range $[0, 1]$, (3.30) is equivalent to:

$$\tanh\left(-\tfrac{c_k}{b_k}\right) \approx -1 \quad \text{, for } k = 1, 2, \ldots, K. \tag{3.31}$$

Since $y = -1$ is an asymptote, the above equation will never be satisfied in the equality but we can select variables b_k, $k \in \{1, 2, \ldots, K\}$, so that the maximum error ϵ_k of the k^{th} tangent component is bounded. More specifically, it is possible to calculate an upper bound for each b_k in order

to meet property P3 according to

$$\tanh\left(-\frac{c_k}{b_k}\right) \leq -1+\epsilon_k \Rightarrow b_k \leq -\frac{c_k}{\tanh^{-1}\left(\epsilon_k-1\right)} \tag{3.32}$$

and since $\tanh^{-1}(\cdot)$ is negative around $r=-1$,

$$b_k \leq \frac{c_k}{|\tanh^{-1}\left(\epsilon_k-1\right)|}. \tag{3.33}$$

By selecting the component bounds appropriately, it is possible to bound the total error $\epsilon = \sum_{k=1}^{K} \epsilon_k$ below a maximum threshold. In addition, due to the relative position of the tangent components, it can be shown that the effect of parameter b_1, i.e. the sigmoidal component that is closer to the point $r=0$, is dominant over the rest and therefore the calculated bound for b_1 is expected to be much tighter for the same error ϵ_k.

Working in the same way, it is possible to calculate the upper bounds for parameters b_k to assure that property P4 is also satisfied and then, by combining the two sets of inequalities, to calculate the final bounds of the calibrating parameters b_k in order to meet the required properties. Additional bounds for parameters b_k will be calculated later to minimize the approximation error of the optimal rate.

3.5.2. Approximation of the Optimal Rate

The family of multi-sigmoidal utilities described in (3.29) is a non-concave function with multiple concave and convex regions. Its first derivative is given by

$$V(r) = \frac{1}{2K}\left\{\sum_{k=1}^{K}\frac{1}{b_k} sech^2\left(\frac{r-c_k}{b_k}\right)\right\}, \tag{3.34}$$

which is not a one-to-one function since the same value $V(\cdot)$ corresponds to more than one rates. Figure 3.3 shows the utility derivative for the multi-sigmoidal example in Figure 3.1[6], in black solid line, which illustrates that a single value of $V(\cdot)$ corresponds to at most $2 \times K$ distinct rates and therefore it is not an invertible function. In addition, it is not possible to invert the function by splitting its domain to one-to-one parts due to the complexity of the calculations.

Despite the fact that these rates can not be calculated by inverting function $V(\cdot)$, it is possible to be approximated efficiently. The approximation methodology relies on the structure of $V(\cdot)$ in (3.34), which is a summation of a number of independent hyperbolic secant components. Moreover, those components are symmetric, they can be inverted separately, and by taking into account that the rate that maximizes Problem (3.7) can only be in a concave region or at zero rate, it is possible to calculate a single rate for each component. The hyperbolic secant components (depicted by coloured dashed lines in Figure 3.3) of the derivative function have the form

$$f_k(r) = \frac{1}{2Kb_k} sech^2\left(\frac{r-c_k}{b_k}\right), \quad k = \{1, 2, \dots K\}. \tag{3.35}$$

Using (3.35), the utility derivative $V(\cdot)$ can be approximated by:

$$V^a(r) = \begin{cases} \frac{sech^2\left(\frac{r-c_1}{b_1}\right)}{2Kb_1}, & 0 \leq r < x_1 \\ & \text{or} \\ \frac{sech^2\left(\frac{r-c_k}{b_k}\right)}{2Kb_k}, & x_{k-1} \leq r < x_k, k \in [2, K-1] \\ & \text{or} \\ \frac{sech^2\left(\frac{r-c_K}{b_K}\right)}{2Kb_K}, & r \geq x_{K-1} \end{cases} \tag{3.36}$$

[6]Neglect the individual hyperbolic secant components for the moment.

Algorithm 4 – Net Utility Maximization

At time slot t, every source node $i = 1, \ldots, M$:

1: receives the aggregate price $\lambda^i(t)$ for using the network resources;
2: **if** source i follows a multi-sigmoidal utility **then**
3: calculates the candidate optimal rates using (3.37);
4: chooses the optimal rate, $r_i^*(\boldsymbol{\lambda})$, as the rate that yields higher value for Problem (3.7) using (3.38);
5: **else if** source i follows a concave utility **then**
6: chooses the optimal rate $r_i^*(\boldsymbol{\lambda})$ using (3.8);
7: **end if**
8: starts transmitting at the next time slot, $t+1$, at rate $r_i^*(\boldsymbol{\lambda})$;

where x_k, $k \in \{1, \ldots, K-1\}$, are the intersection points of the hyperbolic secant components. For example, x_1 in Figure 3.3 is the intersection point between the first two components. If (3.36) is used to approximate the utility derivative of user i in (3.8), for a specific vector $\boldsymbol{\lambda}$, there will be K candidate optimal points, one at each concave part of a hyperbolic secant component. These candidate optimal rates are given by

$$r_i^c(\boldsymbol{\lambda}, k) = b_k^i \cdot sech^{-1}\left(\sqrt{2 \cdot K \cdot b_k^i \cdot \lambda^i}\right) + c_k^i, \tag{3.37}$$

where $sech^{-1}(\cdot)$ is the inverse hyperbolic secant, b_k^i, $k = 1, 2, \ldots, K$, form steepness vector $\boldsymbol{b^i}$ and inflection points c_k^i, $k = 1, 2, \ldots, K$, form inflection vector $\boldsymbol{c^i}$ of user i. An additional candidate solution that can be an optimal allocation is at zero rate and, therefore, the candidate rate $r_i^c(\boldsymbol{\lambda}, K+1) = 0$ must also be taken into account. Consequently, the optimal rate of user i for price vector $\boldsymbol{\lambda}$ will be the one that yields the maximum *net utility*, i.e.

$$r_i^*(\boldsymbol{\lambda}) = \text{argmax}\left\{NU_i\left(r_i^c(\boldsymbol{\lambda}, k)\right) | k = 1, 2, \ldots, K+1\right\}. \tag{3.38}$$

The use of equation (3.38) to approximate the optimal rate for any price vector $\boldsymbol{\lambda}$, and therefore for the optimal vector $\boldsymbol{\lambda}^*$ as well, leads to the de-

Algorithm 5 – Link Price Calculation

At time slot t, every network link $j = 1, \ldots, L$:

1: calculates the incoming aggregate rate;
2: calculates the new price using (3.5);
3: sends the new price $\lambda_j(t+1)$ to all sources that are using link j;

velopment of a distributed gradient based algorithm to solve Problem (3.1). The algorithm consists of two parts; one carried out by each source and one by each link. Algorithm 4 is used by each source node in the network to determine the transmission rate at each time slot based on the aggregate price that a source node has to pay in order to send its traffic to the destination node. At the same time, link j uses Algorithm 5 in order to determine the value of the dual variable λ_j, i.e. the price that every unit of traffic is charged, for the next iteration. The new prices are communicated back to the interested source nodes. As with Algorithm 1 in Chapter 2, this communication can be implemented efficiently by taking into account that each source node is interested in the aggregate price of the used path and not the price of each link individually. Therefore, every intermediate node of a specific path can add its price to the already aggregated price of the previous nodes in a specific field in the acknowledge (ACK) packets. When an ACK packet reaches the source node, it will contain the aggregate price of the path that can be used to calculate the optimal transmission rate for the next iteration. Alternatively, if the link price is viewed as the link delay, the aggregate price can be implicitly measured by the packet queuing delay in the network. The set of distributed algorithms 4 and 5 behave similarly to equations (3.4) and (3.5) regarding the discontinuity points of the optimal rate allocation function. Therefore, the oscillations that are likely to appear can be resolved using the heuristic presented in Section 3.4.

Algorithms 4 and 5 are extensions of the standard gradient-based iterative

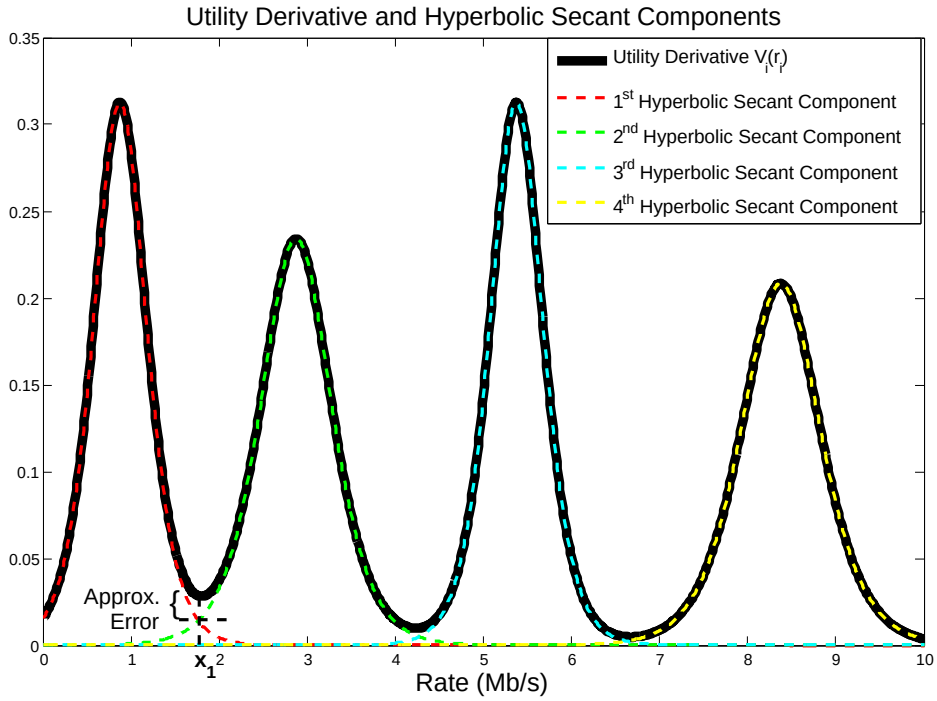

Figure 3.3.: Example of a multi-sigmoidal utility derivative and its four hyperbolic secant components

optimization algorithm and consist of low complexity operations. More specifically, at each iteration the optimal rate is calculated after evaluating (3.37) for each hyperbolic tangent component and choosing the rate that yields the highest net utility, while the new price is calculated using simple mathematical operations in (3.5). The convergence speed and optimality properties of the proposed algorithms are also extensions of the standard gradient algorithm. As explained earlier in Section 3.3.3, there is a trade-off between convergence speed and proximity to the optimal solution, which depends on the value of the step size s_λ, with larger values to help the method converge faster but at the expense of accuracy. In most practical cases, a small positive constant step size is sufficient but alternative methods of variable step size are also available in literature [7].

The procedure described above has transformed (3.4), which involves the solution of a non-convex optimization problem, into a simple selection (out of $K+1$ choices) of the rate that minimizes the net utility using (3.38). However, since it is an approximation method, it is necessary to determine the approximation error and propose methods to minimize it. It is easy to verify from Figure 3.3 that the approximation error has its maximum values at the intersection points x_k, $k = 1, 2, \ldots, K-1$ of two consecutive components. Specifically, the utility derivative at any intersection point x_k is

$$V(x_k) = f_k(x_k) + f_{k+1}(x_k) + \sum_{l=1, l\neq k, l\neq k+1}^{K} f_l(x_k), \tag{3.39}$$

where the intersecting secant components are equal, i.e. $f_k(x_k) = f_{k+1}(x_k) = \gamma_k$ and the rest are almost negligible, i.e. $\sum_{l=1,l\neq k,l\neq k+1}^{K} f_l(x_k) \ll \gamma_k$. However, since $V(x_k)$ is approximated by $f_k(x_k)$, it is clear that the approximation error is affected by the degree of overlap[7] of the hyperbolic secant components. In fact, the effects of this overlapping can be restricted efficiently.

The inflection points of the utility's sigmoidal components are determined by the technology used at the source node and they are assumed that can not be changed. However, there is often some freedom in selecting the steepness parameters of a multi-sigmoidal utility. In such cases, the steepness parameters b_k, $k = 1, 2, \ldots, K$, can be used as design parameters to assure that the approximation error is small. Recall that these parameters were also used earlier to assure that the utility function will be as close to 0 and 1 at points $r = 0$ and $r = r_{max}$ respectively as necessary. In this way, it is possible to calculate some additional bounds for the values of the parame-

[7] We assume that two hyperbolic secant components c_1 and c_2 are not overlapping if $f_{c1}(x_c) = f_{c2}(x_c) \approx 0$ at their intersection point x_c.

ters b_k of the utility function so that the hyperbolic secant components of the utility derivative are non-overlapping. In general, the smaller the values of b_k are, the more concentrated the respective hyperbolic secant component is around the inflection point. Clearly, the choice of b_k for component k affects the range of choices at the neighboring ones and therefore it is not possible to determine analytically a single steepness vector $\boldsymbol{b}$ to assure low approximation error. However, it is possible to formulate optimization problems that calculate the optimal steepness vector $\boldsymbol{b}$ according to various criteria that affect the objective function of the optimization problem. More precisely, we formulate the following optimization problem:

$$
\begin{aligned}
&\max_{\boldsymbol{x},\boldsymbol{b}} \quad g\left(\boldsymbol{x},\boldsymbol{b}\right), \quad \text{such that,} \\
&\text{for } k = 1,\ldots,K-1: \\
&\quad \text{a)} \quad \frac{1}{b_k} sech^2\left(\frac{x_k - c_k}{b_k}\right) \leq \sigma, \\
&\quad \text{b)} \quad \frac{1}{b_{k+1}} sech^2\left(\frac{x_k - c_{k+1}}{b_{k+1}}\right) \leq \sigma, \\
&\quad \text{c)} \quad c_k \leq x_k < c_{k+1}, \\
&\text{for } k = 1,\ldots,K: \\
&\quad \text{d)} \quad b_k > 0.
\end{aligned}
\tag{3.40}
$$

The first two constraints assure that the values of the secant components, and therefore the maximum approximation error as well, are below a maximum threshold σ at point x_k. There is no explicit constraint that points x_k, $k = 1, 2, \ldots, K-1$, are the intersection points of the secant components but the optimal solution for constraints a and b will always be at the intersection points and therefore the role of points x_k is implicitly defined. The objective function $g\left(\boldsymbol{x},\boldsymbol{b}\right)$ can be any concave function that describes

the optimization criteria. For example, during our simulations the function $g(\boldsymbol{x}, \boldsymbol{b}) = \sum_{k=1}^{K} b_k$ was used in order to get the upper bound of the steepness parameters. Constraint c makes sure that the intersection point of two consecutive components is always between their inflection points. Finally, constraint d assures that the optimization variables b_k stay within their domains. Note that one could also extend the set of constraints of Problem (3.40) with the bounds obtained in in the previous section. The resulting $\boldsymbol{b}^*$ vector includes the maximum steepness parameters $b_k^{max}, k = 1, \ldots, K$ for which the maximum approximation error is below the threshold σ. Any value smaller than that will result in smaller error. Problem (3.40) corresponds to each source node's utility function and can be solved independently using any optimization method [7].

3.6. Simulation Results

The algorithms described in the previous sections were simulated in a MATLAB environment in order to study their performance and compare against other approaches. In addition, several examples where network oscillations occurred were examined in order to evaluate the efficiency of the Oscillation Resolving Heuristic (ORH) to stabilize the network. For illustrative purposes, the simulation results are organized in two sections; a single bottleneck network case and a multiple bottleneck network one. There were some common assumptions followed in all simulations with respect to the following parameters. First, the simulation setup included a variety of types of applications, including FTP, HTTP and multimedia applications. This dictated the use of different utility functions, concave or multi-sigmoidal, according to the type of application. All multimedia applications were mod-

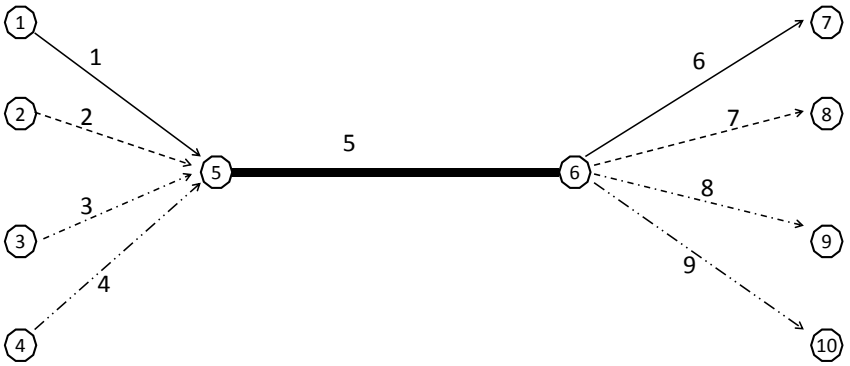

Figure 3.4.: Example of a network topology with a single bottleneck link

elled using multi-sigmoidal utilities according to (3.29) for different inflection and steepness vectors. Furthermore, the calculation of the steepness parameter vector $\boldsymbol{b}^i$ for each multi-sigmoidal utility was done by solving Problem (3.40) for a maximum approximation error $\sigma = 10^{-4}$ using the Global Optimization Toolbox in MATLAB, and, last but not least, all utilities where designed so that their maximum transmitted rate r_{max} is $10Mb/s$ and $U_i(r_{max}) = 1$ for all source nodes $i = 1, 2, \ldots, M$.

3.6.1. Single bottleneck link

Figure 3.4 shows an example topology of a network that has a single bottleneck link. It illustrates a network of ten nodes and nine links, where four nodes act as sources generating traffic towards four destination nodes. The traffic flows are designated by a different line style. The capacities of links $1-4$ and $6-9$ where selected to be $10Mb/s$ so that they do not restrict the bit rate that the respective source nodes transmit. On the other hand, the capacity of link 5 was chosen so that the capacity is inadequate for all sources to transmit at their maximum rate r_{max}, thus creating a bottleneck.

Source nodes 2 and 4 have multi-sigmoidal utilities of four hyperbolic

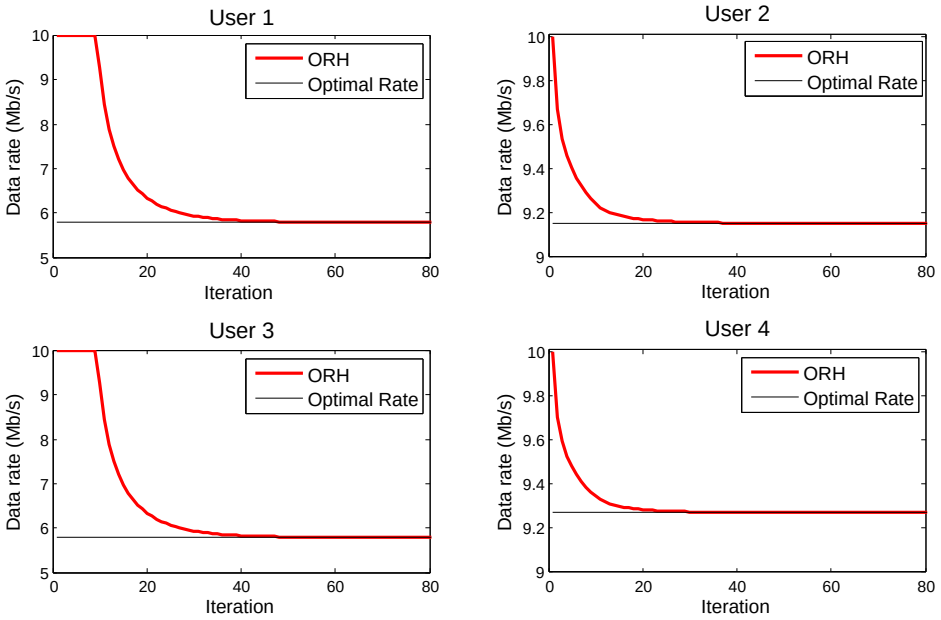

Figure 3.5.: Convergence of rate without oscillation

tangent components while sources 1 and 3 represent HTTP and FTP traffic respectively and are modelled using logarithmic utility functions [45]. Several different values for the capacity of the bottleneck link were used in order to examine cases of network oscillation or stability. In essence, by increasing the bottleneck link capacity, one can decrease the optimal link price due to the availability of more resources and the weakening of the competition among users.

Figure 3.5 shows the convergence of the rates of the source nodes for bottleneck capacity $C_5 = 30Mb/s$. In this case, all links apart from link 5 have zero price and $\lambda_5 = 0.061444$. The resulting aggregate price is not one of the discontinuity points of the multi-sigmoidal utilities and therefore the distributed algorithm converges to the optimal rates as expected. In addition, the algorithm manages to converge to the optimal solution in relatively small number of iterations, around 50, using a constant step size

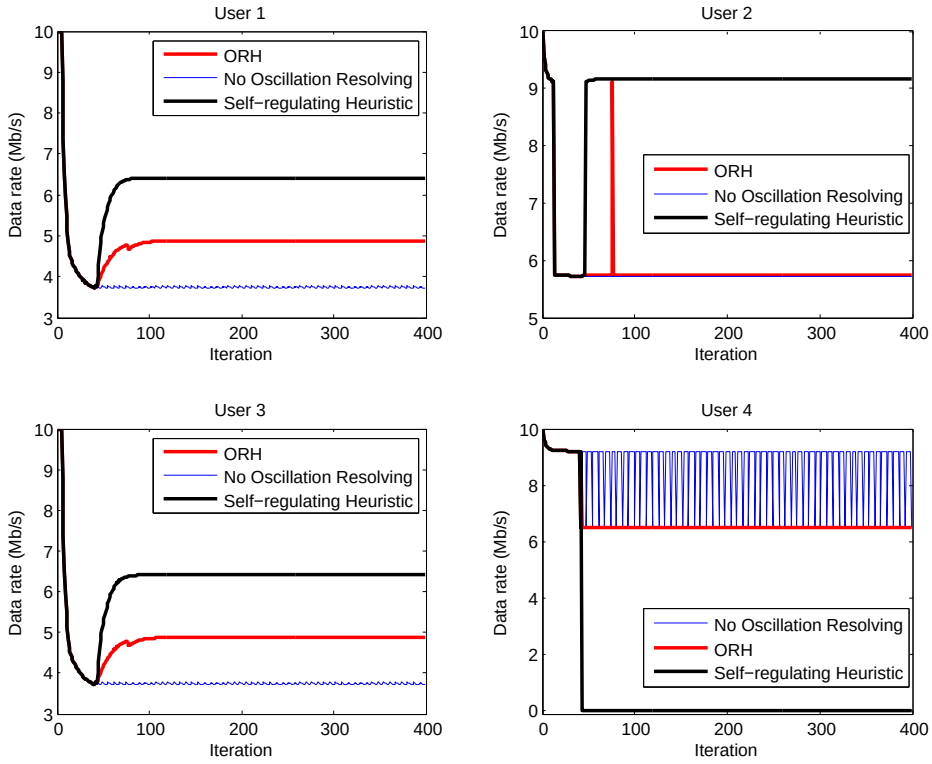

Figure 3.6.: Convergence of rate allocation when oscillation occurs

$s_\lambda = 5 \cdot 10^{-4}$ at all iterations[8].

Figures 3.6 and 3.7 show the convergence of rates and objective function respectively in the case of oscillations and compare the ORH with the self-regulating heuristic presented in [46] in resolving oscillations. The bottleneck link capacity in this case has been set to $C_5 = 22Mb/s$. Once the aggregate link price reaches 0.0722 user 4 starts oscillating. The ORH algorithm then is used to set the rate of user 4 equal to the minimum of its oscillation rates and the optimization continues for users $1-3$. Later, in iteration 77, the aggregate price reaches the discontinuity point of user 2

[8]Faster convergence is also possible with the use of other step methods [7] but the performance evaluation of the standard gradient algorithm is beyond the scope of this paper.

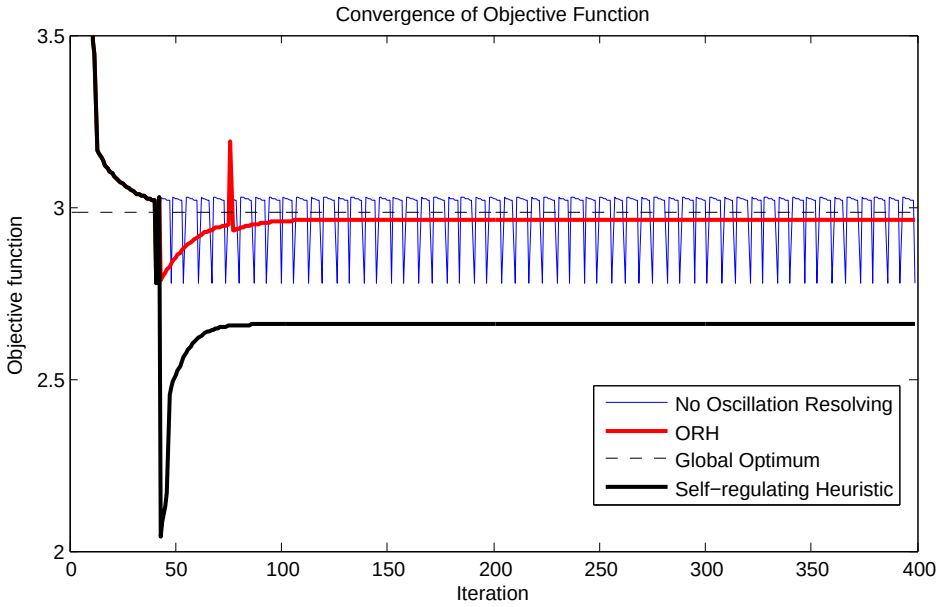

Figure 3.7.: Convergence of the objective function with oscillation

and the *ORH* is again used to resolve the oscillations and allow the distributed algorithm to converge. The self-regulating heuristic sets both rates to zero and removes these users from the rest of the optimization process, which means that only half of the users in the network are allocated some rate. On the other hand, ORH allocates rate to all users and, in addition, the solution that ORH converges to is shown to be very close to the actual optimal value and significantly higher than that achieved by the algorithm proposed in [46] as shown in Figure 3.7.

3.6.2. Multiple bottleneck links

Figure 3.8 illustrates a network topology with three bottleneck links where eight traffic flows are competing for network resources. The different traffic flows are distinguished by a different line style and colour combination. Links 5, 8 and 13 are the bottleneck links while the rest are sufficiently large

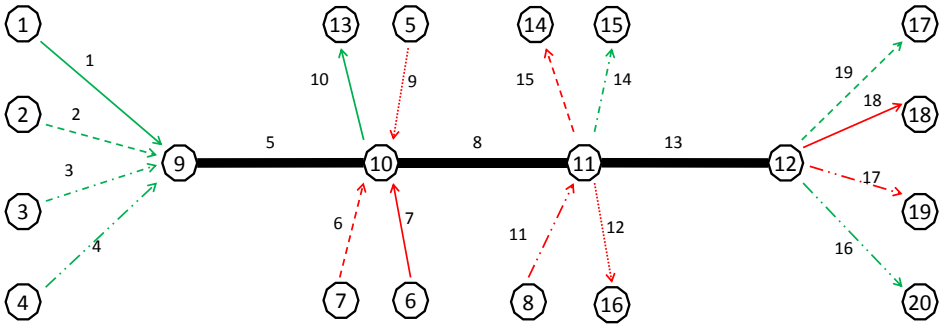

Figure 3.8.: Example of a network topology with multiple bottleneck links

to accommodate traffic even at the maximum transmission rate r_{max}. A combination of elastic and inelastic applications were selected to compete for resources in the network corresponding to both concave and multi-sigmoidal utility functions. Specifically, nodes 2, 3 and 6 measure user satisfaction using concave utilities, while the remaining five flows model multi-tiered multimedia applications.

Figures 3.9 and 3.10 show the convergence of the rate and objective function respectively of ORH, the self-regulating heuristic [46] and the case where no oscillation resolving method is used. For brevity, only the convergence of the first four users is shown. Soon after the initiation of the optimization process, user 4 starts oscillating. *ORH* and the self-regulating heuristic are then evoked to resolve this oscillation and allow the optimization algorithm to converge. Again, the maximum value of the objective function after the application of *ORH* is very close to the optimal one, as calculated using the Global Optimization Toolbox in MATLAB, and significantly higher than the one achieved by the self-regulating heuristic. In addition, ORH allocates rate to all eight users while the self-regulating heuristic to only six of them. Note that the values of the objective function in Figures 3.7 and 3.10 that are larger than the global optimal, in the cases

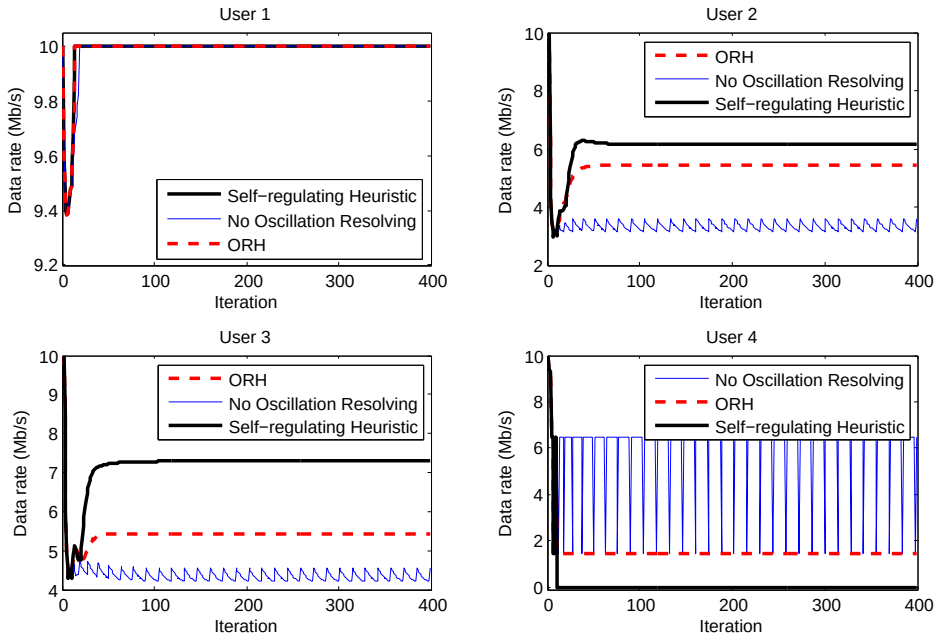

Figure 3.9.: Convergence of rate allocation with oscillation

of ORH and when no oscillation resolving is attempted, correspond to infeasible rate allocations and should be neglected during the comparison of the methods. Finally, the simulation results verify the fast convergence of the algorithm in around 70 iterations. This can be further improved, depending on the application, by using more elaborate gradient update methods, which however, would increase the complexity of the algorithm [7].

3.7. Concluding Remarks

This chapter studied the problem of efficient network resource allocation motivated by the fast growing number of multimedia applications in current communications networks. We introduced the concept of multi-sigmoidal utilities in the context of resource allocation and discussed in depth the challenges that these utilities impose to network utility maximization and

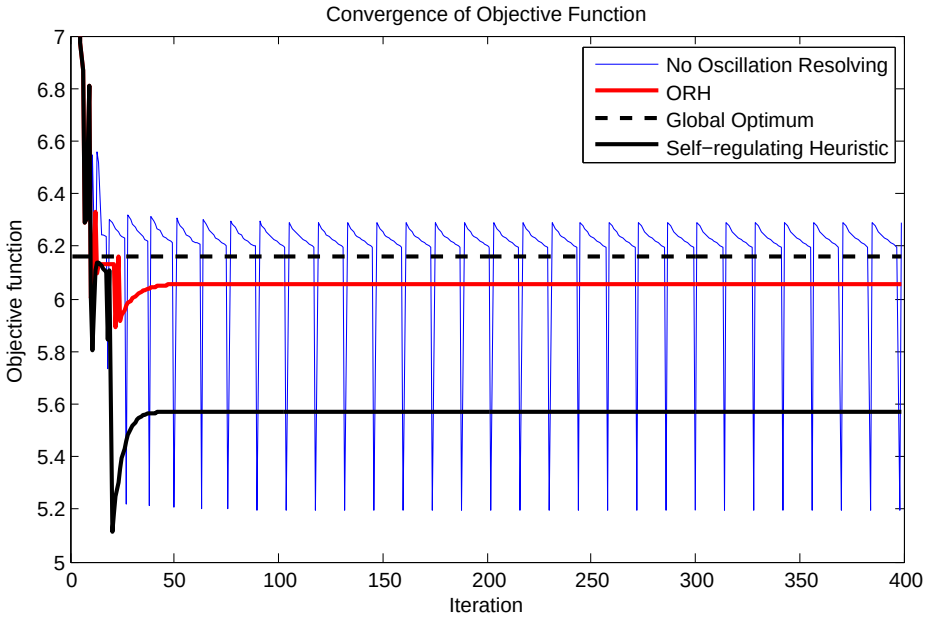

Figure 3.10.: Convergence of the objective function with oscillation

proposed efficient methods to overcome them. To this purpose we proposed a novel mathematical representation of such utility functions and a distributed gradient-based algorithm to optimize the allocation of network bandwidth by exploiting the special structure of the utility function. Then, an efficient heuristic was developed to assure network stability in all cases. Finally, the performance and robustness of the proposed techniques were evaluated through extensive simulations for various network topologies and conditions.

This chapter can be the basis for further research that will consider the foundations of a novel optimization-based Transport Layer protocol. Such a protocol will be able to optimize network performance by taking into account the heterogeneous QoS/QoE requirements of each user and assure efficient use of resources even under the existence of wireless links in the network.

4. Utility-Proportional Fairness for Multimedia Applications in Wireless Networks

4.1. Introduction

As explained earlier in Chapter 1, the proposed *Network Utility Maximization (NUM)* framework [17][18] has found numerous applications in communication networks since it made clear that expressing the network resource allocation process as an optimization problem can be solved by low-complexity distributed algorithms. Such algorithms optimize the resource allocation under two major assumptions; the utilities are all concave functions of rate and all links have fixed capacity, e.g. they model wired links. As explained analytically in Chapters 2 and 3, these two assumptions are responsible for a number of shortcomings of current NUM approaches.

Concave utilities are ideal to model applications that generate *elastic* traffic [42] that relates to applications that can adapt easily to changes in the network conditions. However, the majority of the traffic in current networks is generated by real-time applications that are considered *inelastic*. In practice until now, inelastic applications are handled in the same way as elastic ones. For example, TCP [37][38], the most widely used resource

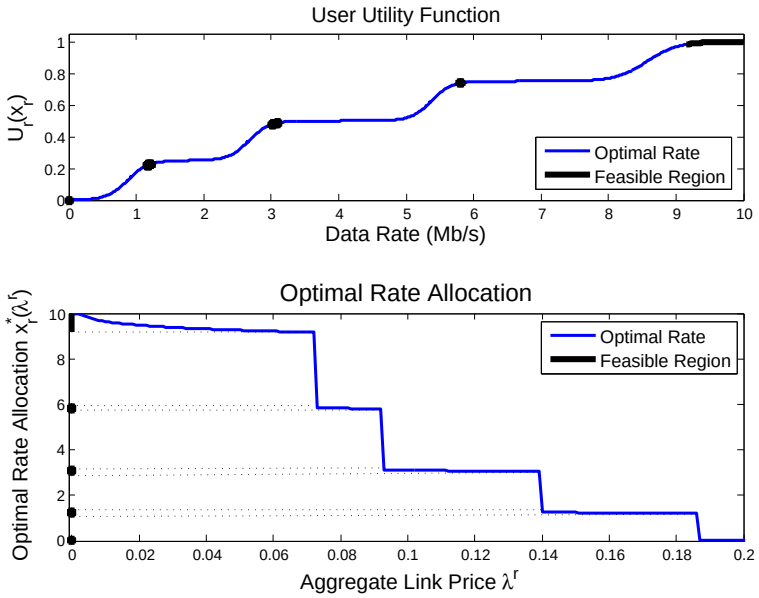

Figure 4.1.: The feasible rate region of a sigmoidal utility function

allocation protocol, models all applications independently of their elasticity properties using the same concave utility function, which varies depending on the TCP version.

Existing research work models inelastic applications using non-concave sigmoidal utility functions [46][48] that turn the resulting formulation into a non-convex problem. In addition, in Chapter 3 we proposed a multi-sigmoidal utility to model modern multimedia applications that support multiple regions of data rate and create a step-like evolution of user Quality of Experience (QoE) with respect to the data rate. An example of such utility function is shown in blue at the top subplot in Figure 4.1.

Despite the existence of an analytic methodology to solve or approximate the optimal solution for such problems in a distributed way, this approach has significant disadvantages:

- The optimal rate allocation function of a source node, $r_i^*\left(\lambda^i\right)$[1], is hard to be calculated in a closed form and therefore numerical gradient-based approaches must be used, which however increase convergence time, don't have guaranteed convergence and are less accurate.

- Function $r_i^*\left(\lambda^i s\right)$ is discontinuous for some values of aggregate link price. This causes oscillations in the network that can prevent the algorithm from converging. An example of $r_i^*\left(\lambda^i\right)$ for a multi-sigmoidal utility function is shown in blue at the bottom subplot of Figure 4.1. The displayed $r_i^*\left(\lambda^i\right)$ has four discontinuity points.

- The heuristics proposed in literature to resolve these oscillations offer approximations that in some cases can be far from the optimal solution.

- Despite the fact that the utility function $U_i\left(r_i\right)$ is defined for rates within the range $\left[r_i^{min}, r_i^{max}\right]$, only a small part of this range can be achieved. This restricts the applicability of such approaches in practical problems. For example, the rate for the utility of Figure 4.1 takes values within the range $[0, 10]$ but the feasible range region (shown in black) is restricted only to either zero or values within the regions $[1.12, 1.16]$, $[3.02, 3.09]$, $[5.79, 5.82]$ and $[9.16, 10]$.

The traditional NUM formulation maximizes the aggregate utility in the network. Moreover, it has been shown [17] that the resulting bandwidth allocations follow the so-called *(bandwidth) proportional fairness*. While this type of fairness seems to perform well when all users follow the same utility, this approach is responsible for some contradictory behaviors in cases that users have different QoS needs, i.e. when they follow different utilities.

[1] where λ^i is the aggregate price in the network

In such cases, proportional fairness favors users which require low rate to achieve high utility [77]. More specifically, a bandwidth proportional fair optimization algorithm favors users with low demand, i.e. those with rapidly increasing utility function. This happens because allocating a unit of rate to a utility with large derivative leads to larger increase in the aggregate utility than when allocating to users with high demand, i.e. with small value of utility derivative. To resolve this contradictory behavior, authors in [77] define a new type of fairness, called *utility proportional fairness*.

As mentioned in Chapter 1, a *utility proportional fair* allocation can be achieved if the utility function of each user is transformed according to:

$$\mathcal{U}_i\left(r_i\right) = \int_{m_i}^{r_i} \frac{1}{U_i\left(y\right)} dy, \quad m_i \leq r_i \leq M_i, \tag{4.1}$$

where m_i and M_i are the minimum and maximum transmission rates for user i respectively, and the objective function of the resource allocation problem[2] is updated to include the summation of all transformed utility functions.

Authors in [77] propose a distributed algorithm to solve the resource allocation problem in order to achieve *utility proportional fairness* in *wired* networks shared by various types of applications. However, current communication networks are often consisted of wireless networks, whose capacity is not constant but depends on the interference of other links. This need highlights the necessity of extending the current *utility proportional fairness* framework to be able to adjust link powers according to the channel conditions in the network.

Motivated by the aforementioned shortcomings of current bandwidth and

[2]See the problem formulation in (1.18) in Chapter 1.

utility proportional fairness mechanisms in wireless networks, this chapter proposes an alternative approach in allocating network resources and makes the following contributions:

- Proposes a utility proportional fair optimization formulation for high-SINR wireless networks. Utility proportional fairness can prevent the oscillations caused when a utility function is non-concave, allow the use of the full range of possible rate values and calculate the optimal rate.

- Derives analytical solutions for the optimal rate allocation function for a number of widely used application types including multi-sigmoidal utilities used to model multi-tiered multimedia applications.

- Proposes a distributed utility proportional fair algorithm to jointly optimize transmission powers and data rates in high-SINR wireless networks.

The rest of this chapter is organized as follows. First, Section 4.2 presents a utility proportional fair optimization formulation for high-SINR wireless networks and gives and insight on a distributed algorithm to solve this problem. Consequently, Section 4.3 provides closed form solutions of the optimal rate for a number of application types, discusses how these formulas can be used to prevent oscillations and presents a distributed gradient-based algorithm. Section 4.4 presents numerical results illustrating the convergence and performance of the proposed approach compared to other approaches in literature and, finally, Section 4.5 concludes the work presented in this chapter.

4.2. Problem Formulation

This section focuses on the development of an optimization formulation for wireless networks that achieves *utility proportional fairness* while taking into account the interference among wireless links and the different QoS requirements of various applications.

4.2.1. Network Model

The system model of this work is similar to those used in the previous chapters. However, we will describe it analytically again to facilitate the reader. Consider a multi-hop wireless network where each node can operate either as data traffic source, destination or relay that just forwards traffic to its neighbors. We define the transmission rate vector $\boldsymbol{r} = [r_1, r_2, \ldots, r_M]^T$, which includes the transmission rates of all M source nodes in the wireless network, and the link l as the tuple (T_l, R_l), where T_l is the transmitting and R_l the receiving node, respectively. We also define $\boldsymbol{p} = [p_1, p_2, \ldots, p_L]^T$ as the vector which includes the transmission powers of the L links. The wireless channel is modelled as follows. Let $\boldsymbol{G}$ be a matrix of size $L \times L$, where G_{km}, with $k, m \in 1, 2, \ldots, L$, represents the path loss coefficient for the path between the transmitter of link k and the receiver of link m. The elements of the path loss matrix $\boldsymbol{G}$ depend on the physical characteristics of the wireless links.

Similarly to the channel model in the previous chapters, a hybrid TDMA/ CDMA scheme is assumed to operate. More specifically, we consider Orthogonal - CDMA (OCDMA) for transmissions towards the same receiver, and pseudo-noise-CDMA (PN-CDMA) between different receivers. This means that the transmitted signal is first spread through multiplication by

a Welsh-Hadamard (WH) sequence with N chips per symbol. Then a PN sequence is overlayed either without further spreading, i.e., with the same chip rate, or with further spreading by a factor S, i.e. number of chips per WH chip. Note that, all users transmitting towards the same receiver employ the same PN sequence, and N orthogonal sequences are reused at each receiver. Moreover, TDMA is employed throughout the multi-hop routes. This implies that time is divided into frames, each of them comprises of two equally sized slots, where transceivers alter from transmitting to receiving mode.

As explained earlier, each source node i is associated with a utility function $U_i(r_i)$, which represents the degree of satisfaction that a user enjoys when sending at a specific rate. In other words, the user utility function reflects the QoE of a user when data content is delivered at a specific rate. This QoE cannot be determined precisely for each user but prior work in the literature [45][46] and our work in Chapter 3 has identified approximate forms/shapes for various applications, such as HTTP, FTP and video streaming applications. Finally, we also associate each wireless link l with a convex cost function $V_l\left(p_l\right)$. This function represents the cost of using the limited power resources of the wireless channel. The incorporation of this cost function leads towards more energy efficient resource allocations for the reasons explained earlier in the previous chapters.

4.2.2. Optimization Problem

The network performance optimization process of a multi-hop wireless network is formulated as the following maximization problem:

$$\begin{aligned} \max_{\boldsymbol{r},\boldsymbol{p}} \quad & \sum_{i=1}^{M} \mathcal{U}_i(r_i) - \gamma \sum_{l=1}^{L} V_l(p_l) \\ \text{s. t.} \quad & \sum_{i=1}^{M} \alpha_{il} r_i \leq C_l\left(\boldsymbol{p}\right), \forall \text{ links } l \end{aligned} \tag{4.2}$$

where $\mathcal{U}_i(r_i)$ is the transformed utility function given by (4.1) for rate r_i, parameter α_{il} is one if the traffic of user i is passing through link l, and zero otherwise. Parameters a_{il}, with $i \in \{1, M\}$ and $l \in 1, L$, form the routing matrix $\boldsymbol{A}$ which is known a priori and fixed throughout the optimization process. The rates r_i, with $i \in 1, 2, \ldots, M$, and powers p_l, with $l \in 1, 2, \ldots, L$, are positive quantities and γ is a positive weighting parameter. Based on the aforementioned channel model, the capacity of a link follows the Shannon's capacity formula, $C_l\left(\boldsymbol{p}\right) = B \cdot \log_2\left(1 + SINR_l\right)$ and is a function of the Signal to Noise plus Interference Ratio (SINR) at the receiver of the link. To avoid the non-concavity of the capacity function, we will restrict ourselves, in this chapter as well, only to high SINR environments where the approximation $SINR_l \gg 1$, the formula $C_l\left(\boldsymbol{p}\right) = B \log_2\left(SINR_l\right)$ can provide a sufficiently accurate approximation of link capacity [21].

Duality Theory [6] provides an efficient methodology to solve optimization problems distributedly. For this reason, one should initially form the

Lagrangian function as

$$L(\boldsymbol{r},\boldsymbol{p},\boldsymbol{\lambda}) = \sum_{i=1}^{M}\left\{\mathcal{U}_i(r_i) - r_i\cdot\lambda^i\right\} + \sum_{l=1}^{L}\lambda_l B\log\left(\frac{NSp_lG_{ll}}{\sum_{k\neq l}p_kG_{kl}+n_l}\right) - \gamma\sum_{l=1}^{L}V_l(p_l),$$

where $\lambda^i = \sum_{l=1}^{L}\alpha_{il}\lambda_l$ is the aggregate price of user i to send a unit of rate through the network and N, S are the chip rate and spreading gain respectively as explained earlier. Consequently, the *dual* optimization problem is defined as:

$$\begin{aligned} \min_{\boldsymbol{\lambda}} \quad & d(\boldsymbol{\lambda}) = L(\boldsymbol{x}^*,\boldsymbol{p}^*,\boldsymbol{\lambda}) \\ \text{s. t.} \quad & \boldsymbol{\lambda}\geq 0. \end{aligned} \tag{4.3}$$

It is evident that Problem (4.2) consists of two subproblems coupled by the dual variable vector $\boldsymbol{\lambda}$. The first one determines the optimal rate to maximize the net revenue of the *source* node, while the second determines the transmission power of the *links*. Consequently, according to *duality theory* every source i can calculate its optimal rate $r_i^*(\boldsymbol{\lambda})$ using

$$r_i^*(\boldsymbol{\lambda}) = \arg\max\left[\mathcal{U}_i(r_i) - r_i\cdot\lambda^i\right]. \tag{4.4}$$

The power and dual variables can be calculated iteratively using:

$$\lambda_l(t) = \lambda_l(t-1) - \delta_\lambda(t)\frac{\partial L(\boldsymbol{r},\boldsymbol{p},\boldsymbol{\lambda})}{\partial\lambda_l} \tag{4.5}$$

$$p_l(t) = p_l(t-1) + \delta_p(t)\frac{\partial L(\boldsymbol{r},\boldsymbol{p},\boldsymbol{\lambda})}{\partial p_l}, \tag{4.6}$$

where $\delta_\lambda(t)$ and $\delta_p(t)$ are small positive step sizes and

$$\frac{\partial L(\boldsymbol{r},\boldsymbol{p},\boldsymbol{\lambda})}{\partial \lambda_l} = B \cdot \log_2 \left(\frac{NSp_l G_{ll}}{\sum_{k \neq l} p_k G_{kl} + n_l} \right) - \sum_{i=1}^{M} \alpha_{il} r_i \tag{4.7}$$

$$\frac{\partial L(\boldsymbol{r},\boldsymbol{p},\boldsymbol{\lambda})}{\partial p_l} = \frac{1}{p_l \ln(2)} \left[\lambda_l - \sum_{m \neq l} \lambda_m \frac{G_{lm} P_l}{\sum_{k \neq m} G_{km} P_k + n_m} \right] - \gamma V_l^{'}(p_l). \tag{4.8}$$

Equations (4.4)-(4.6) constitute a joint primal-dual distributed algorithm, which will be described in detail in the next section along with how utility proportional fairness can lead to the calculation of closed form solutions for (4.4).

4.3. The Price-based Rate Allocation Function

Eq. (4.4) calculates the optimal rate of user i based on the aggregate price λ_i of the path that user i is using to send its traffic. As we proved in Chapter 2, the convergence of any gradient based algorithm using (4.4) depends on its continuity around the optimal price vector $\boldsymbol{\lambda}$. Such continuity can not be guaranteed for *bandwidth proportional* fair algorithms and heuristic techniques such as the *Oscillation Resolving Heuristics* (ORH), or the self-regulating heuristic proposed in [46], must be utilized to approximate the optimal solution. However, the transformation of (4.1) to achieve *utility proportional fairness* can also guarantee continuity of (4.4) and lead to the calculation of analytical solutions. The derivation of such analytical solutions and the development of a distributed optimization algorithm will

be the focus of this section.

4.3.1. Calculating a general rate equation

According to optimization theory [7], the optimal rate will be at the point where the first derivative of the objective function diminishes and therefore

$$r_i^* (\boldsymbol{\lambda}) = \mathcal{U}'^{-1}_i \left(\lambda^i\right). \tag{4.9}$$

In the traditional NUM framework $\mathcal{U}_i (\cdot) = U_i (\cdot)$, where $U_i (\cdot)$ is the utility function of user i as defined earlier. The optimal rate can be calculated using (4.9) only if the utility function is a concave function of rates. If $U_i (\cdot)$ is partially convex and partially concave, as with sigmoidal utilities, the first derivative cannot be inverted since it is not a one-to-one function. For sigmoidal utilities, one should use alternative methods with a negative impact on the algorithm convergence speed. Such an alternative could be a gradient based iterative equation of the form:

$$r_i(t) = r_i(t-1) + \delta_r(t) \frac{\partial L(\boldsymbol{r}, \boldsymbol{p}, \boldsymbol{\lambda})}{\partial r_i} \tag{4.10}$$

where $\delta_r(t)$ is a positive step size and $\frac{\partial L(\boldsymbol{r},\boldsymbol{p},\boldsymbol{\lambda})}{\partial r_i}$ is the gradient of the Lagrangian function with respect to r_i. However, such an approach will not always converge to the global optimum. In fact, according to the condition we proved in Theorem 1, Chapter 2, the algorithm will converge only if (4.4) is continuous around the optimal price vector $\boldsymbol{\lambda}$. If this condition does not hold, there can be oscillations in the network that will prevent the algorithm from converging and an oscillation resolving heuristic is necessary to ensure stability but at the cost of loosing optimality. In addition, (4.10) is a gradient-based iterative equation that may get trapped in local

optima in the case of utilities with multiple concave regions, such as the multi-sigmoidal utility presented in Chapter 3 in (3.29).

Considering *utility proportional fairness*, however, by using the transformation of (4.1), the problem becomes convex even for sigmoidal utilities and (4.4) always satisfies the condition in Theorem 1. More importantly, this allows to calculate a closed form solution for (4.9) directly or, in some cases, use the iterative equation in (4.10). In utility proportional fairness, where the user utility function is transformed according to (4.1), the first derivative can be easily calculated as:

$$\mathcal{U}_i^{'}\left(r_i\right) = \frac{1}{U_i\left(r_i\right)}. \tag{4.11}$$

Eq. (4.11) is invertible as long as it is continuous and monotonic, which are both true for any concave utility and any sigmoidal utility of arbitrary number of inflection points following the shape shown in Figure 4.1. In this case, the optimal rate is given by:

$$r_i^*\left(\lambda^i\right) = U_i^{-1}\left(\frac{1}{\lambda^i}\right). \tag{4.12}$$

Based on (4.12), we can calculate a closed form solution for utilities that satisfy these two properties. This is a significant advantage of the utility proportional fairness approach which leads to the development of algorithms that calculate the optimal solution even for non-concave utilities and converge significantly faster than the traditional approach that uses (4.10).

4.3.2. Application specific forms

The existence of various types of user applications complicates the process of calculating a general closed form solution for the optimal rate allocation

Application Type	Optimal Rate Allocation Function
HTTP	$r_i^*(\boldsymbol{\lambda}) = r^{min} \cdot \left(\frac{r^{max}}{r^{min}}\right)^{\frac{1}{\lambda^i}}$
FTP	$r_i^*(\boldsymbol{\lambda}) = (r^{max}+1)^{\frac{1}{\lambda^i}} - 1$
Single-tiered Video Application	$r_i^*(\boldsymbol{\lambda}) = \frac{\alpha \cdot \beta - \ln(\lambda^i - 1)}{\alpha}$
Multi-tiered Video Application	$r_i^*(\lambda^i) = b_j \cdot \operatorname{arctanh}\left(2\left(\mathrm{K}\frac{1}{\lambda^i} - \mathrm{j}\right) + 1\right) + \mathrm{c_j}$

Table 4.1.: The Optimal Resource Allocation Function for Widely Used Types of Applications

function of a user. It is however possible to derive application-specific analytical solutions for (4.4) that can be used in a distributed algorithm to jointly optimize the transmission rates and powers.

Based on the analysis above, it is possible to derive the optimal rate allocation for browsing, file transfer and video streaming applications using the suggested utility functions in [45], [46] and the multi-sigmoidal utility function we proposed in Chapter 3, when *utility proportional fairness* is applied. These optimal rate allocation functions are demonstrated in Table 4.1. r^{min} and r^{max} represent the minimum and maximum transmission rate of a user, and parameters α and β are calibration parameters of the single-sigmoidal utility. The calculation of analytical solutions for concave and single-sigmoidal utilities is relatively easy and is provided in Appendix A. However, the calculation for multi-sigmoidal utilities such as those described in (3.29) is more complicated and will be described in detail in the remainder of this section.

A multi-sigmoidal utility consists of K hyperbolic tangent components. As explained earlier, by definition, the hyperbolic tangent has values in the range $(-1, 1)$ but the components in (3.29) have been scaled and shifted so that the resulting utility has values in the range $[0, 1]$. Therefore, each of the scaled components is restricted in a different non overlapping region. For

example, values in the range $(0.5, 0.75)$ correspond to the third hyperbolic tangent component of the utility in the top plot of Figure 4.1. This implies that a value of utility corresponds to a single point[3] and belongs to only one of the hyperbolic tangent components, while the rest of the components have value either 1 or -1. To calculate its inverse we write:

$$y = \frac{1}{2K}\left\{\sum_{k=1}^{K}\tanh\left(\frac{r_i - c_k}{b_k}\right) + K\right\} \Leftrightarrow 2Ky - K = \sum_{k=1}^{K}\tanh\left(\frac{r_i - c_k}{b_k}\right) \Rightarrow$$

$$2Ky - K = \mu + \tanh\left(\frac{r_i - c_j}{b_j}\right) - \varphi. \tag{4.13}$$

Index j represents the index of the hyperbolic tangent component that corresponds to the requested point. Term μ represents the components before j that have value 1, i.e. $\mu = j - 1$, and term φ represents the components after j that have value -1, i.e. $\varphi = K - j$. Based on these, (4.13) becomes:

$$2\left(Ky - j\right) + 1 = \tanh\left(\frac{r_i - c_k}{b_k}\right), \tag{4.14}$$

and by solving with respect to r_i, we find that:

$$r_i^*\left(y\right) = b_j \cdot \operatorname{arctanh}\left(2\left(\mathrm{Ky} - \mathrm{j}\right) + 1\right) + \mathrm{c_j}. \tag{4.15}$$

Moreover, by combining (4.12) and (4.15) we calculate the optimal rate allocation of user i with respect to the aggregate network price for i as

$$r_i^*\left(\lambda^i\right) = b_j \cdot \operatorname{arctanh}\left(2\left(\mathrm{K}\frac{1}{\lambda^{\mathrm{i}}} - \mathrm{j}\right) + 1\right) + \mathrm{c_j}. \tag{4.16}$$

Eq. (4.16) is a closed form of the optimal rate allocation for a specific aggregate price λ^i when the utility function has multi-sigmoidal shape, i.e.

[3] i.e., it is a one-to-one function.

Component	Utility Value Region	Aggregate Price Region
1	$\left[0, \frac{1}{4}\right]$	$(4, \infty)$
2	$\left[\frac{1}{K}, \frac{2}{K}\right]$	$(2, 4)$
3	$\left[\frac{2}{K}, \frac{3}{K}\right]$	$\left(\frac{4}{3}, 2\right)$
4	$\left[\frac{3}{K}, 1\right]$	$\left[0, \frac{4}{3}\right)$

Table 4.2.: Tangent Components and the Respective Utility and Aggregate Price Value Regions for a Utility with 4 sigmoidal components

when it models multi-tiered multimedia applications. In order to evaluate (4.16), it is necessary to determine the hyperbolic tangent component that the specific aggregate price λ^i corresponds to, i.e. determine the value of j. According to the *first order necessary condition for optimality* [7], at the optimal solution $\mathcal{U}_i^{'}\left(r_i^*\right) = \lambda^i$, which leads to

$$U_i\left(r_i^*\right) = \frac{1}{\lambda^i}. \tag{4.17}$$

Eq. (4.17) shows that the regions of utility values can be easily mapped to regions of aggregate price values. Specifically, for a multi-sigmoidal utility with K inflection points, the hyperbolic component j is within region $\left[\frac{j-1}{K}, \frac{j}{K}\right]$, with $j = 1, 2, \ldots, K$, of the utility values and corresponds to prices in the region $\left(\frac{K}{j}, \frac{K}{j-1}\right)$, with $\frac{K}{0} \to \infty$. In other words, depending on the value of the aggregate price λ^i, we can determine the component that the optimal rate belongs to and specify j. For example, Table 4.2 shows the utility value regions and their respective aggregate price regions in utility proportional fairness for a multi-sigmoidal utility given by (3.29) for $K = 4$. Note, that aggregate prices within $[0, 1)$ correspond to $U_i = 1$ and therefore to component $j = K$.

By splitting the summation of hyperbolic tangent components and calculating the inverse of only one of them, we create some discontinuities on the

boundaries of the aggregate price regions. These discontinuities are caused by the fact that arctanh(x) $\rightarrow \pm\infty$ when $x \rightarrow \pm 1$ respectively. Specifically for (4.16) the discontinuities appear on the intermediate boundaries since, by definition of the utility function, $r_i^*(0) = r_i^{max}$ and $r_i^*(\infty) = 0$. For example, in the case of a multi-sigmoidal utility with $K = 4$, the discontinuities exist for $\lambda^i = \frac{4}{3}$, $\lambda^i = 2$ and $\lambda^i = 4$. In order to handle these discontinuities and assure continuity of the rate allocation function, one could assign an approximation of the optimal rate for these boundary cases based on neighboring rate values. In other words, the optimal rate $r_i^*\left(\lambda^i\right)$ for the boundary aggregate prices can be calculated by a transformation of the form:

$$r_i^*\left(\lambda^i\right) = f\left(r_i^*\left(\lambda_-^i\right), r_i^*\left(\lambda_+^i\right)\right), \tag{4.18}$$

where $\lambda_-^i = \lambda^i - \epsilon$, $\lambda_+^i = \lambda^i + \epsilon$ and ϵ is a very small positive constant. A potential approach could be a weighted average of the rates for prices λ_-^i and λ_+^i according to:

$$r_i^*(\lambda^i) = \frac{w_1 \cdot r_i^*(\lambda_-^i) + w_2 \cdot r_i^*(\lambda_+^i)}{w_1 + w_2}, \tag{4.19}$$

where w_1 and w_2 are weighting parameters with $w_k > 0$, $k \in \{1, 2\}$. The relative values of the parameters w_1 and w_2 can be used to select a rate value that is closer to one or the other discontinuity end. For example, $w_1 > w_2$ implies that $r_i^*(\lambda^i)$ will be closer to $r_i^*(\lambda_-^i)$ than to $r_i^*(\lambda_+^i)$. For the numerical results later in this paper, we will use (4.19) to calculate the optimal rate for boundary aggregate prices with $w_1 = w_2 = \frac{1}{2}$ and $\epsilon = 10^{-8}$.

This weighted averaging of neighboring points for the estimation of the optimal rate assures that (4.16) is a continuous function of the aggregate price. This continuity for all aggregate prices also implies that when us-

Algorithm 6 – Optimal Rate Calculation

At time t, source $i = 1, \ldots, M$:

1: receives the aggregate price $\lambda^i(t)$;
2: calculates the optimal rate, $r_i^* (\boldsymbol{\lambda})$, using (4.4) or the formulas in Table 4.1;
3: starts transmitting at time $t+1$ at rate $r_i^* (\boldsymbol{\lambda})$;

Algorithm 7 – Link Price and Power Calculation

At time t, a link $l = 1, \ldots, L$:

1: calculates the incoming aggregate rate;
2: calculates the new price using (4.5);
3: calculates the new power using (4.6);
4: sends the new price $\lambda_l (t+1)$ to all sources that are using link l and starts transmitting using $p_l (t+1)$;

ing *utility proportional fairness* all rates within the range $\left[r^{min}, r^{max}\right]$ are feasible contrary to the bandwidth proportional fairness case, where only a small part of the total rate range is feasible, as illustrated in Figure 4.1. This shows that the rate allocation function has the robustness and elasticity to adjust to any changes in the link prices and take advantage of the full range of the available rate region in order to maximize user satisfaction in the network.

4.3.3. Distributed Algorithm

Having formulated the proportional fair optimization problem for wireless networks and derived analytical solutions of the optimal rate allocation function for some of the most common applications, the next step is to develop a distributed algorithm to jointly optimize transmission powers and data rates in the aforementioned wireless network.

The iterative equations (4.4)-(4.6) and the application-specific results summarized in Table 4.1 can be used to create a distributed algorithm to jointly optimize rates, powers and prices with minimum information ex-

change between users. This algorithm consists of two parts; Algorithm 6 is carried out by each source node and Algorithm 7 in each link. This joint algorithm is an extension of the standard gradient-based algorithm and will converge to the optimal solution for sufficiently small values of the step sizes $\delta_\lambda(t)$ and $\delta_p(t)$ [7], since Problem (4.2) has been convexified using the utility transformation of (4.1) and the High-SINR Shannon capacity approximation formula. Regarding the information exchange of the algorithm, users need to know the aggregate link price λ^i in order to determine the optimal transmission rate for the next iteration of the algorithm execution. Similarly to the information exchange of the proposed algorithms in the previous chapters, the aggregate price can be either stored in the ACK packets sent by the destination to the source node, or, if the link price is viewed as the link delay, it can be implicitly measured by the packet queuing delay in the network, which can be easily obtained from the lower layers of the protocol stack with no additional signaling overhead. Finally, as with the distributed algorithm in Chapter 2, using the cost function $V_l(p_l)$ is a natural way of assuring both energy efficiency and convergence of the distributed power control algorithm.

4.4. Numerical Results

The *Utility Proportional Fairness (UPF)* approach was applied to various network scenarios in MATLAB where the network resources were being used by a number of elastic and inelastic applications to send traffic to a set of destination nodes. The goal of our simulations was twofold; first, to quantify the improvement that a closed form solution for the optimal rate allocation (such as those presented in Table 4.1) can offer compared to an

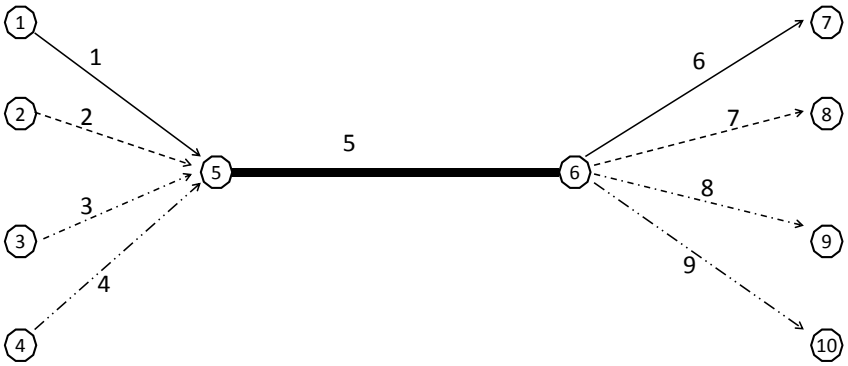

Figure 4.2.: Simple Network Topology Example

iterative gradient based equation, such as (4.10), and, second, to compare UPF against the widely used *Bandwidth Proportional Fairness (BPF)* resource allocation policy. Therefore, the simulation results presented in this section are organized in two parts. In subsection 4.4.1, we compare the convergence performance of the closed form rate allocation algorithm in UPF against the respective iterative algorithm, and, then, the advantages of the UPF approach against the BPF are highlighted in subsection 4.4.2.

4.4.1. Convergence Comparison of Iterative and Analytical Solution Methods

Algorithms 6 and 7 describe an iterative approach to calculate the optimal values of three sets of variables; the transmission rates of the sources, the transmission powers of the wireless links and the link prices. To isolate the performance of the rate allocation mechanism in these algorithms, we simulated them in the simple symmetric *wired* network of Figure 4.2. This implies that all links in the network have fixed capacity that is known a priori, and Problem (4.2) reduces to a rate only allocation optimization

Application Type	User Utility Function
HTTP	$U_i(r_i) = \frac{\log\left(\frac{r_i}{r^{min}}\right)}{\log\left(\frac{r^{max}}{r^{min}}\right)}$
FTP	$U_i(r_i) = \frac{\log(r_i+1)}{\log(r^{max}+1)}$
Single-tiered Video Appl.	$U_i(r_i) = \frac{1}{1+e^{-\alpha(r_i-\beta)}}$
Multi-tiered Video Appl.	$U_i(r_i) = \frac{1}{2K}\left\{\sum_{k=1}^{K}\tanh\left(\frac{x_r-c_k}{b_k}\right)+K\right\}$

Table 4.3.: The Utility functions of Some Widely Used Types of Applications

formulation of the form:

$$\begin{aligned} \max_{\boldsymbol{r}} \quad & \sum_{i=1}^{M}\mathcal{U}_i(r_i) \\ \text{s. t.} \quad & \sum_{i=1}^{M}\alpha_{il}r_i \leq C_l,\ \forall \text{ links } l \end{aligned} \tag{4.20}$$

that was covered extensively for the BPF approach in Chapter 3. This allows us to evaluate the impact of using the closed form solutions summarized in Table 4.1 without the effects in convergence of the power allocation mechanism.

To solve this problem, we ran Algorithm 6 and Algorithm 7 after omitting line 3. The link capacity vector during these simulations was $\boldsymbol{C} = [15, 15, 15, 15, 25, 15, 15, 15, 15]^T$, source nodes 1 and 4 were assumed to have multi-sigmoidal utilities, and source 2 and 3 to host elastic applications, i.e. user satisfaction for these applications to be modelled using concave utilities. The two non-concave utilities followed the multi-sigmoidal shape of (3.29) with the following parameters; $\boldsymbol{b} = [0.0722, 0.1957, 0.2237, 0.3980]^T$, $\boldsymbol{c} = [0.875, 2.675, 5.375, 8.575]^T$ for the first user and parameter vectors $\boldsymbol{b} = [0.1269, 0.2960, 0.1274, 0.3366]^T$, $\boldsymbol{c} = [1.25, 3.75, 6.25, 8.75]^T$ for the second one respectively. Finally, the concave utilities followed the FTP utility function of Table 4.3 for $r^{max} = 10$. With this particular choices for utility

functions, link 5 turns into a bottleneck for the network and the rate allocation of this link among the competing users will determine the final rate vector. The convergence speed in both cases depends also on the selection of the step size for the link price update equation, $\delta_\lambda(t)$, which was set as $\delta_\lambda(t) = 0.03$ in both cases. The step size for the iterative rate allocation approach, described by (4.10), was also set to $\delta_r(t) = 0.03$. These parameters were set empirically to achieve fast converge according to the step size limitations of the general gradient algorithm [7].

Figure 4.3 shows the convergence of the rate allocation of user 1, which follows a multi-sigmoidal utility. Both methods converged to the same rate allocation vector but the use of the analytical form of Table 4.1 is improving the convergence speed of the algorithm significantly. Specifically, the purely iterative approach needs in this example around 5800 iterations, while the closed form solution needs less than 1/6 of these, around 900. These iterations are in essence the iterations needed by Algorithm 7 to calculate iteratively the optimal link prices. Once this is done, a single evaluation of the rate allocation function yields the optimal rate for a specific user. Due to the significant improvement in convergence time of the analytical approach, this method was used during the comparison of UPF and BPF methods, the results of which are presented in the next subsection.

4.4.2. Comparison of Bandwidth and Utility Proportional Fairness Methods

The two approaches were simulated in various network scenarios in a MATLAB environment in order to compare their convergence properties, the differences in the allocation mechanism and show that UPF can successfully avoid the occurrence of rate oscillations and can lead to fair allocation

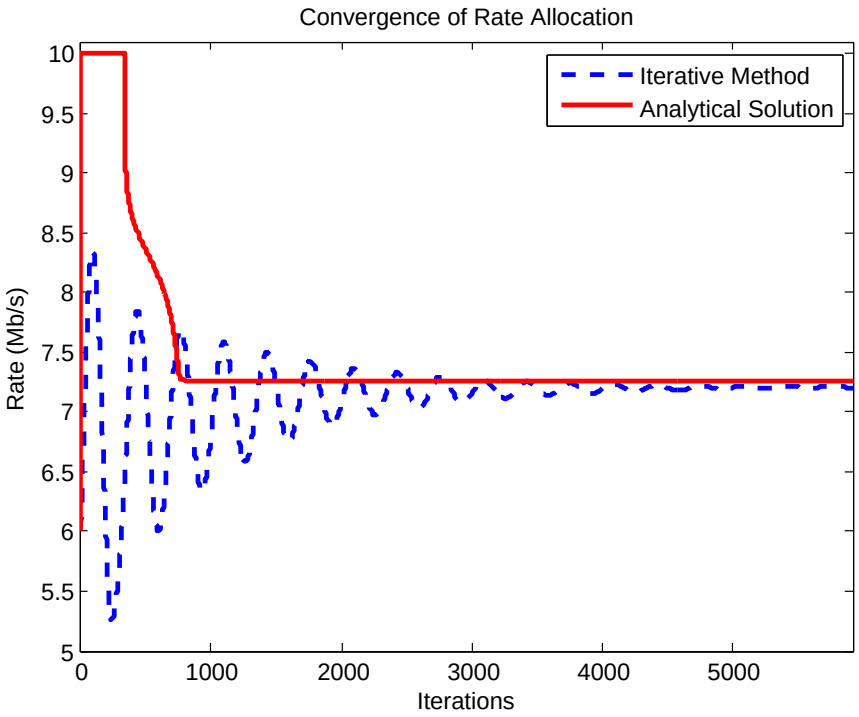

Figure 4.3.: Convergence Speed Comparison of Iterative and Analytical Solution Rate Allocation Methods

of resources when heterogeneous applications compete. For a more complete comparison, this section is organized in two parts. In the first part, the two approaches are compared in a wireless network scenario under the existence of only concave and single-sigmoidal utilities, while, in the second part, multi-sigmoidal utilities will be also used to model multi-tiered multimedia applications.

Concave and Single-sigmoidal Utilities

The *utility proportional fairness (UPF)* approach was applied to various network scenarios, an example of which is the network topology shown in Figure 4.4 for illustration purposes. The wireless network consists of 6 source nodes, 3 intermediate nodes and a set of 3 destination nodes. The

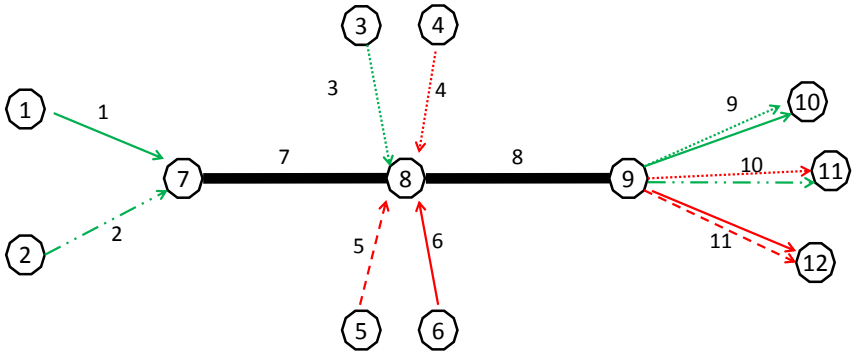

Figure 4.4.: Network Topology Example

simulation setup consisted of a variety of types of applications, including FTP, HTTP and single-tiered multimedia applications. This dictated the use of different utility functions, concave or sigmoidal, according to the type of application. All applications were modelled using the utilities of Table 4.3 for various values of parameters. More specifically for the example of Figure 4.4, source nodes 1-3 and 5 serve real-time applications, whereas source nodes 4 and 6 serve elastic applications modelled by concave utilities. The path loss coefficients G_{ll} were significantly larger than these of the interfering channels, i.e. terms G_{kl} for $k = 1, \ldots, L$ and $k \neq l$, in order to allow the use of the high-SINR channel capacity approximation formula with low approximation error.

The performance of the UPF approach is compared against the traditional *bandwidth proportional fairness (BPF)* [46] approach used in prior work in order to show that UPF can successfully avoid the occurrence of rate oscillations and can lead to fair allocation of resources when heterogeneous applications compete for them. During the BFP optimization process, the *self-regulating* heuristic [46] was used in order to resolve any oscillations that might occur.

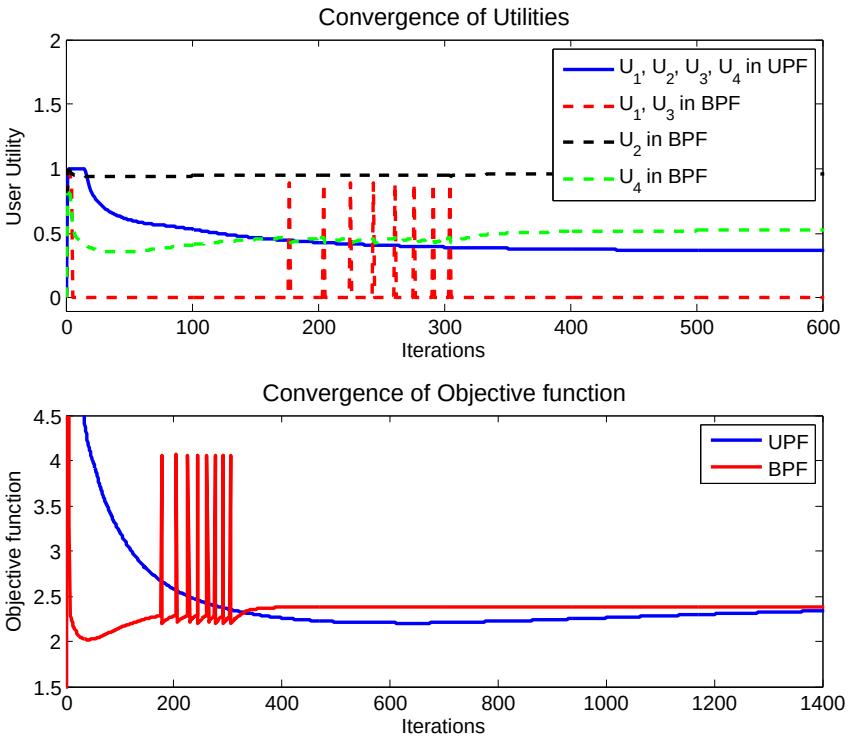

Figure 4.5.: Convergence of Utility and Objective Functions

Figure 4.5 shows the convergence of both the objective function of the optimization problem and the utility functions of sources $1-4$. When BPF is used, users 1 and 3 follow a sigmoidal utility and start to oscillate after about 180 iterations, as the spikes indicate. The *self-regulating* heuristic removes them from the optimization process and therefore their utility is 0. The remaining users compete for all the network resources which leads to higher individual utilities for these users. On the other hand, there are no oscillations when UPF is used and the resulting rate allocation leads to the same degree of satisfaction for all sources. In general, UPF gives priority to users with higher rate requirements while BPF allocates more rate to users that are satisfied easier in an attempt to achieve higher aggregate utility in the network. For example, at the final rate allocation in BPF, all the

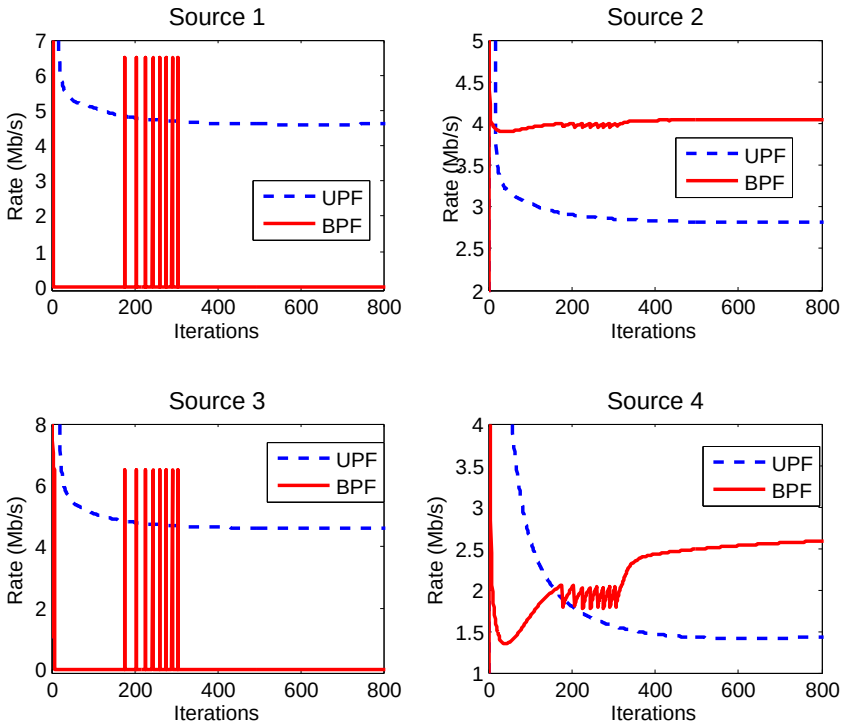

Figure 4.6.: Convergence of Rate Allocation

elastic applications are allocated some rate while only two out of the four multimedia applications are allowed to transmit.

The convergence of the rate allocation of the first four sources for both UPF and BPF approaches is illustrated in Figure 4.6. It is evident that for BPF the oscillations occurring at the rate allocation of sources 1 and 3 cause spikes in the allocations of the rest as well, while in UPF rate are converging smoothly to the optimal solution. Finally, Figure 4.7 shows the convergence of the power allocation for links 1 to 4. It is evident from the peaks around iteration 190 that the existence of oscillations in the BPF approach affects the convergence of powers as well, whereas in UPF the powers converge smoothly to their optimal values. In addition, it is clear that the different allocation policy between UPF and BPF also leads to

different values of transmission powers due to the difference in the traffic passing through each link.

Concave and Multi-sigmoidal Utilities

An example topology is shown in Figure 4.8 for illustration purposes. The wireless network consists of 4 source nodes, 4 intermediate nodes and 1 destination node. As with the previous subsection, the simulation setup consisted of a variety of types of applications, including FTP, HTTP and multimedia applications. This dictated the use of different utility functions, concave or sigmoidal, according to the type of application. All applications were modelled using the utilities of Table 4.3 for various values of parameters. More specifically for the example of Figure 4.8, we used the utilities described in subsection 4.4.1. The hybrid TDMA/CDMA scheme described

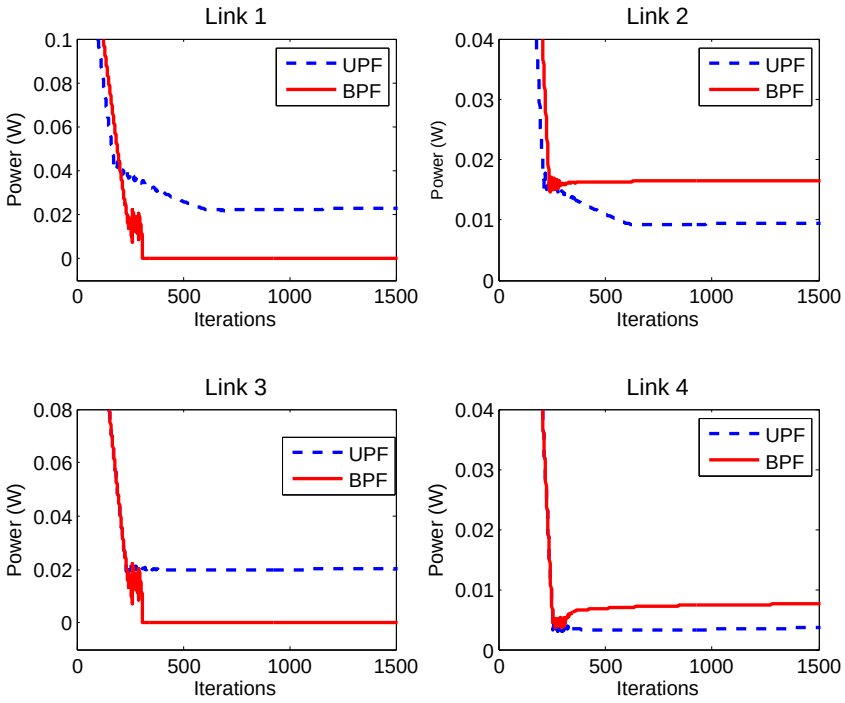

Figure 4.7.: Convergence of Transmission Power Allocation

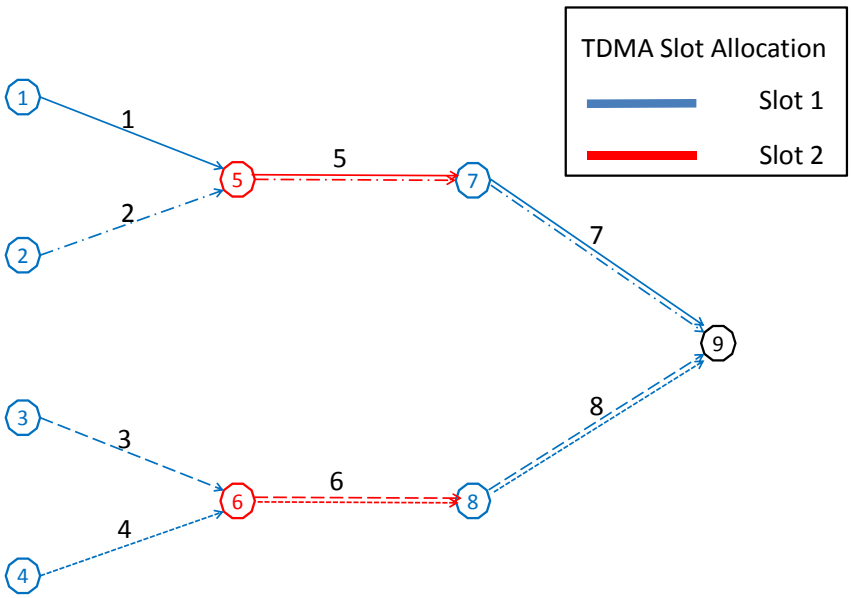

Figure 4.8.: Wireless Network Topology Example

in Section 4.2 was deployed with $N = 2$ chips per symbol, a spreading gain $S = 4$ and channel bandwidth of $B = 2$MHz. The path loss coefficients G_{ll} were significantly larger than these of the interfering channels, i.e. terms G_{kl} for $k = 1, \ldots, L$ and $k \neq l$, in order to allow the use of the high-SINR channel capacity approximation formula with low approximation error. For the BPF approach, any oscillations that occurred during the optimization process were resolved using the *Oscillation Resolving heuristic (ORH)* presented in Chapter 3. Another option would be to use the *self-regulating heuristic* in [46]. Regarding the power vectors, it is assumed that there is a feasible power vector to achieve capacity adequate to accommodate the non-concave utilities when transmitting at rate equal to the minimum of their oscillating rates.

Figure 4.9 shows the convergence of the aggregate utilities in the network and the individual utility functions of sources 3 and 4. The convergence of

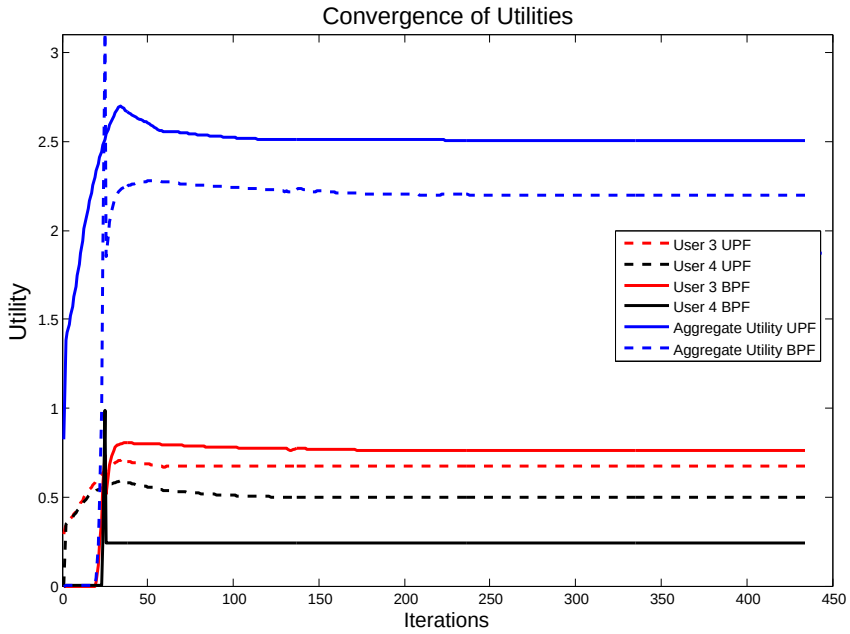

Figure 4.9.: Convergence of User Utility Functions

utilities for users 1 and 2 were similar and omitted for clarity. When BPF is used, users 1 and 4, who follow a multi-sigmoidal utility, start to oscillate after around 25 iterations, as the spike indicate. The oscillation resolving heuristic allocates to each one of them rate equal to the minimum of their oscillating rates and, consequently, removes them from the optimization process. Therefore, their utility remains constant from that point and until the end of the optimization process. The remaining users compete for the rest of the network resources. On the other hand, there are no oscillations when UPF is used since the problem in this case is convex. In addition, by comparing the individual utilities one can observe that the resulting rate allocation leads to almost the same degree of satisfaction for all sources[4]. In general, UPF gives priority to users with higher rate requirements, i.e. users

[4]If the wireless network topology had been exactly symmetric, the individual utilities would have been identical

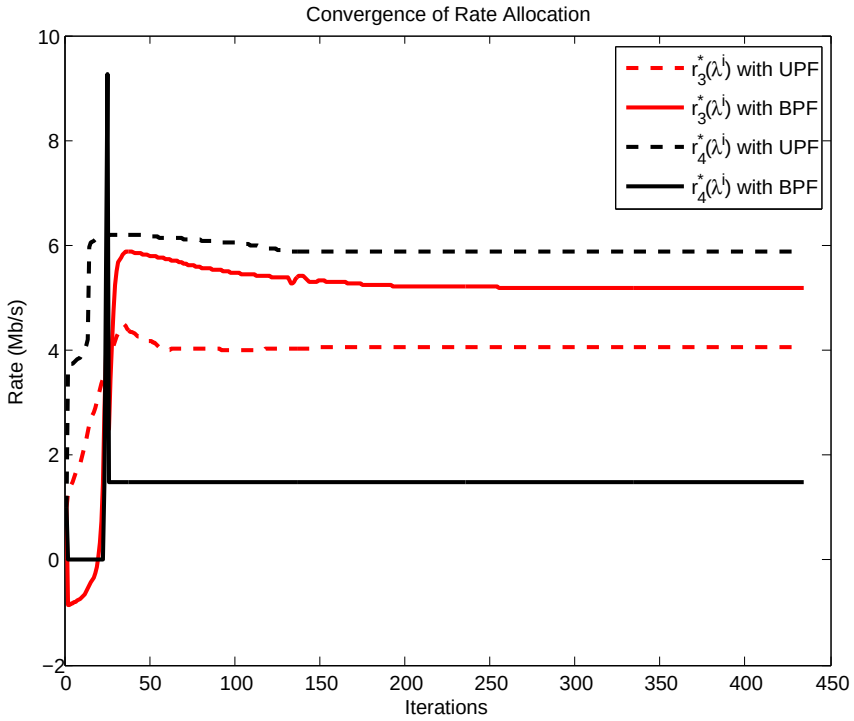

Figure 4.10.: Convergence of Rate Allocation

that need larger amount of rate to achieve a specific value of satisfaction, while BPF allocates more rate to users that are satisfied easier, i.e. users with larger value of derivative. This stems from the fact that BPF tries to achieve higher aggregate utility in the network while UPF tries to balance the degree of satisfaction among users.

The convergence of the rate allocation for sources 3 and 4 for both UPF and BPF approaches is illustrated in Figure 4.10. In the case of BPF, the oscillations occurring at the rate allocation of source 4 affect also the allocations of the concave users. Once these oscillations are resolved, the BPF algorithm converges. On the other hand, in UPF, rates are converging smoother since the optimization problem is convex and hence no oscillations occur. As explained for the utility convergence, comparing the resulting rate

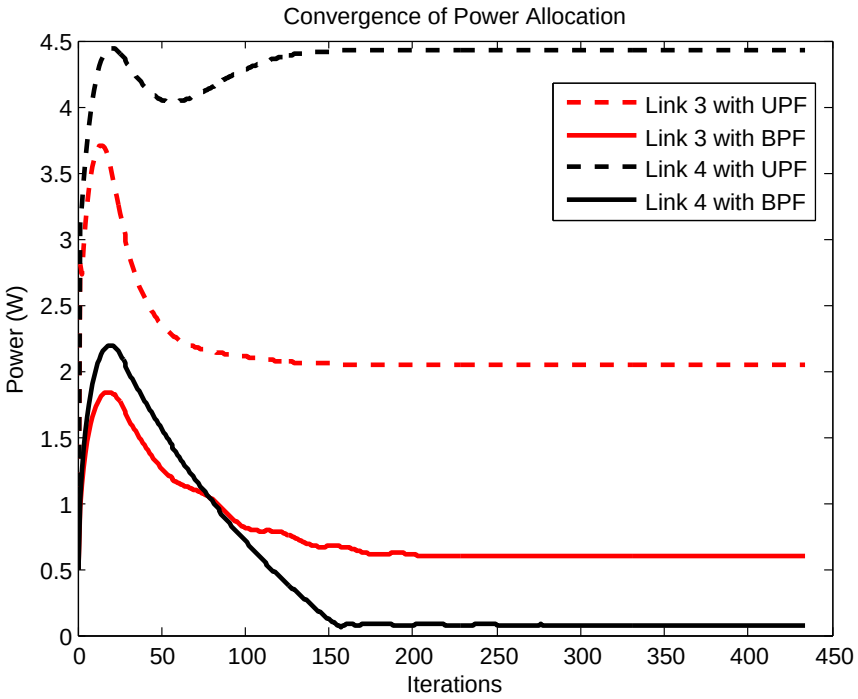

Figure 4.11.: Convergence of Transmission Power Allocation

allocations, it is evident that the UPF approach allocates more rate to the two multi-tiered multimedia applications, i.e. to users 1 and 4, rather than to concave applications.

Figure 4.11 shows the convergence of the power allocation for links 3 and 4. The algorithm behavior for the remaining links is similar and the respective plots where omitted for clarity. It is clear that the different allocation policy between UPF and BPF also leads to different values of transmission powers due to the difference in the traffic passing through each link. Moreover, the resulting power allocations for UPF are significantly higher that in BPF in order to accommodate the increased total network capacity achieved. Finally, power convergence in the case of UPF is smoother due to the convexity of the problem and the absence of rate discontinuities.

4.5. Concluding Remarks

This chapter discussed how *utility proportional fairness* can be used to resolve many of the shortcomings of traditional NUM approaches in wireless networks. More specifically, we proposed a utility proportional-fair optimization formulation for high-SINR wireless networks and developed a joint distributed rate and power allocation algorithm to solve this problem. In addition, it was shown that the use of utility proportional fairness allows the calculation of closed form solutions for the optimal rate allocation for a wide range of popular applications, including multi-tiered multimedia applications, prevents oscillations in the network and assures that all applications will be treated equally in terms of the rate allocation. Our approach was also simulated and compared against the traditional bandwidth proportional fair approach.

5. Conclusion and Future Work

The aim of this chapter is twofold. First, it will provide a summary of the research methodologies and results presented in the previous chapters and, then, it will outline possible directions for future work in this very promising research area.

5.1. Summary of the Results

The methodologies, algorithms and results that were presented in this PhD Thesis contribute to both fundamentals of optimization theory in general and to the development of efficient engineering techniques to solve practical problems that occur in current communication problems.

Chapter 2 demonstrates the shortcomings of TCP, the most widely used Transport layer protocol in the internet, when dealing with inelastic traffic that is generated by multimedia applications. This inability of TCP to allocate resources efficiently highlights the need to design a novel optimization-based Transport layer protocol that can take into account the evolution of user satisfaction for each type of application and be able to optimize the resource allocation so that the total user satisfaction is maximized under some well-defined objective. However, current optimization frameworks cannot provide the foundations of such an optimization-based protocol since the resulting non-convex formulations are difficult to be solved by in a dis-

tributed way. However, there are cases of non-convex problems that can be solved distributedly as easy as a convex one.

Towards the development of such a Transport layer protocol, Chapter 2 describes a non-convex optimization framework. At the core of this framework, theorems 1 and 2 prove a sufficient (and in some cases necessary as well) condition to identify the non-convex optimization problems that can be solved distributedly. This theorem connects the convergence of the distributed algorithm with the continuity properties of the rate allocation function. The great advantage of this framework is its generality and its wide applicability, which is demonstrated in solving the resource allocation problem in TDMA/CDMA ad-hoc networks. The proposed distributed algorithm is shown to converge to the optimal solution when Theorem 1 holds. In the opposite case, the discontinuity of the rate allocation function of a user might cause this user to oscillate between two rate values and consequently prevent the distributed algorithm from converging. To stop this oscillation behavior we proposed a simple oscillation resolving heuristic that assures the convergence of the algorithm. Moreover, our simulations show that this heuristic provides an efficient approximation of the optimal solution.

Chapter 3 provides significant contribution regarding the necessity for the use of multi-sigmoidal utilities in resource allocation. First, we argue on the reasons why single-sigmoidal utilities are not adequate to model modern multi-tiered multimedia applications and provide analytical results on the effect of a multi-sigmoidal utility on the continuity properties of the rate allocation function. These results are necessary in order to investigate the applicability of the non-convex framework provided in Chapter 2. Therefore, we prove that the rate allocation function of a user can have as many

discontinuities as the number of inflections points of its utility and that these points can cause a distributed algorithm to oscillate similarly to the single-sigmoidal case. Furthermore, these discontinuity points can be determined efficiently using the detailed steps of the proposed algorithm. Moreover, we propose a specific family of multi-sigmoidal utility functions that is appropriate to model user satisfaction for multi-tiered multimedia applications. The efficient calibration of the parameters of this function is shown and the development of a very efficient approximation method of the optimal rate is described. Finally, the case of oscillations is thoroughly examined and an efficient heuristic algorithm for oscillations with multi-sigmoidal utilities is also proposed and applied to various network topologies.

Chapter 4 proposes an alternative allocation policy that can provide improvements in shortcomings of the traditional resource allocation methods. These shortcomings can be summarized in the following two reasons; one that relates to the fairness characteristics of the allocation policy applied and one relating to the oscillations that occur and the range of feasible rates. Traditional resource allocation methods, including those presented in Chapters 2 and 3, despite the fact that lead to the maximum possible aggregate rate, lead to some unfair behavior towards inelastic applications. The applied Bandwidth Proportional Fairness policy applied gives priority to users that are easier satisfied. This leads applications which need more rate, such as multimedia applications, to receive less resources, thus creating users that are very satisfied and users with almost zero utility. Moreover, as explained analytically in Chapters 2 and 3, the discontinuity of the rate allocation function can lead to cases of oscillations and restricts the range of feasible rates that each user can be allocated, and thus removing most of the elasticity of the proposed algorithms.

The Utility Proportional Fairness approach presented in 4 can resolve successfully the aforementioned problems since, with the appropriate utility transformation, it can assure the convexity of the problem, and hence the continuity of the rate allocation function, and, moreover, can lead towards more fair resource allocations by giving priority to applications that need resources the most. Therefore, we propose an analytic optimization framework for resource allocation in wireless networks under the existence of multi-sigmoidal utility functions and prove closed form solutions for the rate allocation function of a number of widely used application types and show that an optimization-based algorithm will converge to an optimal and fair solution that will attempt to satisfy all applications at the same extend. Moreover, the continuity of the rate allocation function provides our algorithm with the necessary robustness by allowing the use of the full rate range and not just a small portion of it.

5.2. Future Work

Despite the aforementioned contributions of the current PhD work, there are significant research challenges yet to be answered, which are also in our future research plans and will be outlined in this section.

The non-convex optimization framework described in Chapter 2, can identify the non-convex problems that can be solved distributedly using a gradient-based algorithm. In addition, in case that the condition of Theorem 1 does not hold and oscillations occur, the proposed heuristic can assure stability of the system and provide an efficient approximation of the optimal solution. However, at the moment, it is not possible to know whether the condition of Theorem 1 holds for a problem before trying to solve it.

Therefore, it is within our research plans to develop a detailed procedure to evaluate whether a non-convex problem can be solved distributedly with a gradient-based algorithm before attempting to solve it. This will prevent cases of oscillation and will possibly lead to the development of more sophisticated techniques to solve these problems distributedly.

The applicability of the proposed framework will be further improved if more cases of utility function are examined. As our research showed, the continuity properties of the rate allocation function depends solely on the utility function of that user. This offers the opportunity to examine more types of utilities in order to determine the discontinuity points and act faster in case of oscillations. Moreover, this will lead to more accurate modelling of the satisfaction of users with respect to the transmission rate and, hence, to more efficient resource allocations.

Our work regarding Utility Proportional Fairness showed that an allocation policy apart from optimal can also be fair towards network users. In addition, oscillation policies can act as an efficient convexification tool for non-convex problem formulations. It is within our future plan to investigate the effect of other fairness policies in the resource allocation. The incorporation of policies such as the Utility Max-min Fairness or the Bandwidth Max-min Fairness will be further examined.

The problem formulations where the applicability of our optimization techniques was demonstrated were mainly regarding wireless ad-hoc networks. Some aspects of energy efficiency were taken into account but energy efficiency is not the main consideration in wireless networks consisted of non-battery powered hosts. Therefore, we plan to investigate novel problem formulations that describe the resource allocation problem in wireless sensor networks. The energy considerations in battery-operated sensor networks

are of significant importance and can take the form of lifetime maximization or selective operation of sensors based on the spatial characteristics of the measured phenomena. In addition, alternative, and more accurate, channel models will be considered that will allow the application of our research even to low-SINR environment.

Another limitation of the work that relates to NUM is the fact that the routing matrix of the network is considered fixed and known a priori. This is supported by an implicit assumption that a routing algorithm is run before the application of the resource allocation algorithm and that the network is static enough to allow the distributed optimization-based algorithm to converge to the optimal solution. However, there are cases, especially in wireless networks, where links change and therefore the routing should be updated during the optimization process. Therefore, other approaches should be examined such as the case of joint rate, power and routing optimization.

Last but not least, our greatest motivation so far and our ultimate goal of our future research is to combine all the previous items and work towards the development of an optimization-based resource allocation protocol that will substitute current suboptimal protocols, such as TCP, and will be able to optimize the performance of a network while taking into account the unique characteristics of the applications utilizing it.

A. Appendix

A.1. Proof of Optimal Rate Allocation Function for HTTP Application

Using the utility function for the HTTP application of Table 4.3, we have that:

$$y \cdot \log\left(\frac{r^{max}}{r^{min}}\right) = \log\left(\frac{x}{r^{min}}\right) \Leftrightarrow \log\left(\left\{\frac{r^{max}}{r^{min}}\right\}^{y}\right) = \log\left(\frac{x}{r^{min}}\right) \Leftrightarrow$$

$$x = r^{min} \cdot \left\{\frac{r^{max}}{r^{min}}\right\}^{y}. \tag{A.1}$$

From (A.1) we conclude that the inverse of the utility function is:

$$U_i^{-1}(x) = r_{min} \cdot \left\{\frac{r^{max}}{r^{min}}\right\}^{x}, \tag{A.2}$$

and substituting in (4.12) we prove that:

$$r_i^*(\boldsymbol{\lambda}) = r^{min} \cdot \left(\frac{r^{max}}{r^{min}}\right)^{\frac{1}{\lambda^i}}. \tag{A.3}$$

A.2. Proof of Optimal Rate Allocation Function for FTP Application

Working in a similar way for the utility function of the FTP application according to Table 4.3, we have that:

$$\log(x+1) = y \cdot \log(r^{max}+1) \Leftrightarrow \log(x+1) = \log(\{r^{max}+1\}^{y})$$

$$x = \{r^{max}+1\}^{y} - 1. \tag{A.4}$$

Consequently, from (A.4) we conclude that the inverse of the utility function is:

$$U_i^{-1}(x) = \{r^{max}+1\}^{x} - 1, \tag{A.5}$$

and substituting in (4.12) we prove that:

$$r_i^*(\boldsymbol{\lambda}) = (r^{max}+1)^{\frac{1}{\lambda^i}} - 1. \tag{A.6}$$

A.3. Proof of Optimal Rate Allocation Function for Single-tiered Video Application

Similarly with the other two cases above, we have that:

$$y = \frac{1}{1+e^{-\alpha(r_i-\beta)}} \Leftrightarrow \frac{1-y}{y} = e^{-\alpha(r_i-\beta)} \Leftrightarrow$$

$$x = \frac{\alpha\beta - \ln\left(\frac{1}{y}-1\right)}{\alpha}, \tag{A.7}$$

which leads to the following inverse utility function:

$$U_i^{-1}(x) = \frac{\alpha\beta - \ln\left(\frac{1}{x} - 1\right)}{\alpha} \tag{A.8}$$

Then, in combination with (4.12) we conclude that the optimal rate allocation function is:

$$r_i^*(\boldsymbol{\lambda}) = \frac{\alpha \cdot \beta - \ln\left(\lambda^i - 1\right)}{\alpha}. \tag{A.9}$$

Bibliography

[1] L. Kleinrock, "On communications and networks," *IEEE Transactions on Computers*, vol. C-25, no. 12, pp. 1326 –1335, December 1976.

[2] J. Day and H. Zimmermann, "The osi reference model," *Proceedings of the IEEE*, vol. 71, no. 12, pp. 1334 – 1340, December 1983.

[3] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update, 20102015," Cisco Systems Inc., Tech. Rep., February 2011. [Online]. Available: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360.pdf

[4] ——, "Cisco visual networking index: Forecast and methodology, 20102015," Cisco Systems Inc., Tech. Rep., June 2011. [Online]. Available: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.pdf

[5] C. A. Floudas, I. G. Akrotirianakis, S. Caratzoulas, C. A. Meyer, and J. Kallrath, "Global optimization in the 21st century: advances and challenges," *Computers and Chemichal Engineering*, vol. 29, pp. 1185–1202, May 2005.

[6] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[7] D. P. Bertsekas, *Nonlinear Programming.* Athena Scientific, 1999.

[8] R. Horst, P. M. Pardalos, and N. V. Thoai, *Introduction to Global Optimization.* Kluwer Academic Publishers, 1995.

[9] P. Pardalos and H. Romeijn, "Handbook of global optimization," *Heuristic approaches, Kluwer, Dordrecht*, 2002.

[10] C. A. Floudas and C. E. Gounaris, "A review of recent advances in global optimization," *Journal of Global Optimization*, vol. 45, pp. 3–38, 2009.

[11] T. Weise, *Global Optimization Algorithms Theory and Application.* it-weise.de (self-published), 2009, [Online]. Available: http://www.it-weise.de/.

[12] D. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods.* Prentice Hall, 1989.

[13] Z. quan Luo, W. kin Ma, A.-C. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 20 –34, May 2010.

[14] J. B. Lasserre, "An explicit exact sdp relaxation for nonlinear 0-1 programs," in *Proceedings of the 8th International IPCO Conference on Integer Programming and Combinatorial Optimization.* London, UK, UK: Springer-Verlag, 2001, pp. 293–303.

[15] M. Fazel and M. Chiang, "Network utility maximization with nonconcave utilities using sum-of-squares method," in *IEEE CDC-ECC 2005*, December 2005, pp. 1867 – 1874.

[16] X. L. Sun, K. McKinnon, and D. Li, "A convexification method for a class of global optimization problems with applications to reliability optimization," *Journal of Global Optimization*, vol. 21, pp. 185–199, 2001.

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

[18] S. Low and D. Lapsley, "Optimization flow control. i. basic algorithm and convergence," *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 861 –874, December 1999.

[19] D. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1439 –1451, August 2006.

[20] D. P. Palomar and M. Chiang, "Alternative distributed algorithms for network utility maximization: Framework and applications," *IEEE Transactions on Automatic Control*, vol. 52, no. 12, pp. 2254 –2269, December 2007.

[21] M. Chiang, "To layer or not to layer: balancing transport and physical layers in wireless multihop networks," in *IEEE INFOCOM 2004*, vol. 4, March 2004, pp. 2525 – 2536.

[22] M. Chiang, S. Low, A. Calderbank, and J. Doyle, "Layering as optimization decomposition: A mathematical theory of network architectures," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 255 –312, January 2007.

[23] J. Postel, "Transmission control protocol," Internet Engineering Task Force, RFC 793, September 1981. [Online]. Available: http://www.rfc-editor.org/rfc/rfc793.txt

[24] H. Balakrishnan, V. Padmanabhan, S. Seshan, and R. Katz, "A comparison of mechanisms for improving tcp performance over wireless links," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 756 –769, December 1997.

[25] S. F. M. Mathis, J. Mahdavi and A. Romanow, "Selective acknowledgement options," Internet Engineering Task Force, RFC 2018, October 1996. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2018.txt

[26] H. Balakrishnan and R. Katz, "Explicit loss notification and wireless web performance," in *IEEE GlobeCom mini-conference*, Sydney, Australia, November 1998.

[27] K. Ramakrishnan, S. Floyd, and D. Black, "The addition of explicit congestion notification (ecn) to ip," Internet Engineering Task Force, RFC 3168, September 2001. [Online]. Available: http://tools.ietf.org/html/rfc3168

[28] S. S. H. Balakrishnan and R. Katz, "Improving reliable transport and handoff performance in cellular networks," *ACM Wireless Networking*, November 1995.

[29] P. Karn, "The qualcomm cdma digital cellular system," in *Mobile & Location-Independent Computing Symposium on Mobile & Location-Independent Computing Symposium*, ser. MLCS. Berkeley, CA,

USA: USENIX Association, 1993, pp. 4–4. [Online]. Available: http://dl.acm.org/citation.cfm?id=1287073.1287077

[30] S. Nanda, R. Ejzak, and B. Doshi, "A retransmission scheme for circuit-mode data on wireless links," *IEEE Journal on Selected Areas in Communications*, vol. 12, no. 8, pp. 1338 –1352, October 1994.

[31] E. Ayanoglu, S. Paul, T. F. LaPorta, K. K. Sabnani, and R. D. Gitlin, "Airmail: a link-layer protocol for wireless networks," *Journal of Wireless Networks*, vol. 1, no. 1, pp. 47–60, February 1995.

[32] V. Jacobson, "Congestion avoidance and control," in *Symposium proceedings on Communications architectures and protocols*, ser. SIGCOMM '88. New York, NY, USA: ACM, 1988, pp. 314–329. [Online]. Available: http://doi.acm.org/10.1145/52324.52356

[33] L. S. Brakmo and L. L. Peterson, "Tcp vegas: End to end congestion avoidance on a global internet," *IEEE Journal on selected Areas in communications*, vol. 13, pp. 1465–1480, 1995.

[34] T. H. S. Floyd and A. Gurtov, "The newreno modification to tcp's fast recovery algorithm," Internet Engineering Task Force, RFC 3782, April 2004. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3782.txt

[35] A. Kumar, "Comparative performance analysis of versions of tcp in a local network with a lossy link," *IEEE/ACM Transactions on Networking*, vol. 6, no. 4, pp. 485 –498, August 1998.

[36] S. H. Low, L. L. Peterson, and L. Wang, "Understanding tcp vegas: Theory and practice," Princeton University, Tech. Rep. TR 616-00, November 2000. [Online]. Available: http://www.cs.princeton.edu/research/techreps/TR-616-00

[37] ——, "Understanding tcp vegas: a duality model," *Journal of the ACM*, vol. 49, no. 2, pp. 207–235, 2002.

[38] S. H. Low, "A duality model of tcp and queue management algorithms," *IEEE/ACM Transactions on Networking*, vol. 11, no. 4, pp. 525–536, 2003.

[39] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397 –413, August 1993.

[40] D. Lapsley and S. Low, "Random early marking for internet congestion control," in *Global Telecommunications Conference, 1999. GLOBECOM '99*, vol. 3, 1999, pp. 1747 –1752 vol.3.

[41] R. T. Rockafellar, "Lagrange multipliers and optimality," *SIAM Rev.*, vol. 35, pp. 183 –283, 1993.

[42] W. Stallings, *Data and Computer Communications*, 9th ed. Pearson Custom Publishing, 2010.

[43] S. Shenker, "Fundamental design issues for the future internet," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 7, pp. 1176–1188, September 1995.

[44] J. He, J. Rexford, and M. Chiang, "Don't optimize existing protocols, design optimizable protocols," *SIGCOMM Computer Communications Review*, vol. 37, no. 3, pp. 53–58, July 2007.

[45] C. Liu, L. Shi, and B. Liu, "Utility-based bandwidth allocation for triple-play services," in *Universal Multiservice Networks, 2007. ECUMN '07. Fourth European Conference on*, feb. 2007, pp. 327 –336.

[46] J.-W. Lee, R. R. Mazumdar, and N. B. Shroff, "Non-convex optimization and rate control for multi-class services in the internet," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 4, pp. 827–840, August 2005.

[47] M. Chiang, S. Zhang, and P. Hande, "Distributed rate allocation for inelastic flows: optimization frameworks, optimality conditions, and optimal algorithms," in *IEEE INFOCOM 2005*, vol. 4, March 2005, pp. 2679 – 2690 vol. 4.

[48] P. Hande, S. Zhang, and M. Chiang, "Distributed rate allocation for inelastic flows," *IEEE/ACM Transactions on Networking*, vol. 15, no. 6, pp. 1240 –1253, December 2007.

[49] J.-W. Lee, R. Mazumdar, and N. Shroff, "Nonconvexity issues for internet rate control with multiclass services: stability and optimality," in *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 1, March 2004, pp. 24–34.

[50] M. Chiang and J. Bell, "Balancing supply and demand of bandwidth in wireless cellular networks: utility maximization over powers and rates," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 4, March 2004, pp. 2800 – 2811 vol.4.

[51] M. Chiang, "Balancing transport and physical layers in wireless multihop networks: Jointly optimal congestion control and power control," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 104–116, January 2005.

[52] J. W. Lee, R. Mazumdar, and N. Shroff, "Downlink power allocation for multi-class cdma wireless networks," in *IEEE INFOCOM 2002*, vol. 3, November 2002, pp. 1480 – 1489 vol.3.

[53] Y. Hou, K. K. Leung, and A. Misra, "Mission-based joint optimal resource allocation in wireless multicast sensor networks," in *Annual Conference of ITA*, Maryland, USA, September 2009.

[54] G. Tychogiorgos, K. Leung, A. Misra, and T. LaPorta, "Distributed network utility optimization in wireless sensor networks using power control," in *IEEE PIMRC 2008*, September 2008, pp. 1 –6.

[55] S. Ulukus and R. Yates, "Stochastic power control for cellular radio systems," *IEEE Transactions on Communications*, vol. 46, no. 6, pp. 784 –798, June 1998.

[56] A. Ribeiro and G. Giannakis, "Layer separability of wireless networks," in *CISS 2008*, March 2008, pp. 821 –826.

[57] J. Papandriopoulos, S. Dey, and J. S. Evans, "Optimal and distributed protocols for cross-layer design of physical and transport layers in manets," *IEEE/ACM Transactions on Networking*, vol. 16, no. 6, pp. 1392–1405, 2008.

[58] D. O'Neill, A. Goldsmith, and S. Boyd, "Wireless network utility maximization," in *IEEE Military Communications Conference, MILCOM*, November 2008, pp. 1 –8.

[59] D. Fouskakis and D. Draper, "Stochastic optimization: A review," *International Statistical Review*, vol. 70, pp. 315–350, 2002.

[60] Y. Yi and M. Chiang, "Stochastic network utility maximization," *European Transactions on Telecommunications*, vol. 22, pp. 1–22, 2008.

[61] M. Chiang, D. Shah, and A. Tang, "Stochastic stability under network utility maximization: General file size distribution," in *In Proceedings of Allerton Conference*, 2006, pp. 49–77.

[62] L. Massoulie, "Structural properties of proportional fairness: Stability and insensitivity," *Annals of Applied Probability*, vol. 17, no. 3, pp. 809 –839, 2007.

[63] R. Srikant, "On the positive recurrence of a markov chain describing file arrivals and departures in a congestion-controlled network," in *IEEE Computer Communications Workshop*, 2005.

[64] F. Baccelli, D. R. McDonald, and J. Reynier, "A mean-field model for multiple tcp connections through a buffer implementing red," *Performance Evaluation*, vol. 49, no. 1-4, pp. 77–97, September 2002.

[65] S. Shakkottai and R. Srikant, "Mean fde models for internet congestion control under a many-flows regime," *IEEE Transactions on Information Theory*, vol. 50, no. 6, pp. 1050 – 1072, June 2004.

[66] S. Deb, S. Shakkottai, and R. Srikant, "Asymptotic behavior of internet congestion controllers in a many-flows regime," *Mathematics of Operation Research*, vol. 30, no. 2, pp. 420–440, May 2005.

[67] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936 –1948, December 1992.

[68] M. Neely, E. Modiano, and C.-P. Li, "Fairness and optimal stochastic control for heterogeneous networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 2, pp. 396 –409, April 2008.

[69] M. Neely, "Delay-based network utility maximization," in *IEEE INFOCOM*, March 2010, pp. 1 –9.

[70] ——, "Stochastic network optimization with non-convex utilities and costs," in *Proc. Information Theory and Applications Workshop (ITA)*, February 2010, pp. 1 –9.

[71] L. Georgiadis, M. J. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," *Foundations and Trends in Networking*, vol. 1, no. 1, pp. 1–144, April 2006.

[72] D. P. Bertsekas and R. Gallager, *Data Networks*. Prentice-Hall, 1992.

[73] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 556 –567, October 2000.

[74] B. Radunovic and J.-Y. L. Boudec, "Why max-min fairness is not suitable for multi-hop wireless networks," Ecole Polytechnique Federale de Lausanne (EPFL), Tech. Rep., 2003.

[75] T. Bonald, L. Massoulié, A. Proutière, and J. Virtamo, "A queueing analysis of max-min fairness, proportional fairness and balanced fairness," *Queueing Syst. Theory Appl.*, vol. 53, no. 1-2, pp. 65–84, June 2006.

[76] L. Tassiulas and S. Sarkar, "Maxmin fair scheduling in wireless networks," in *INFOCOM 2002. Twenty-First Annual Joint Conference*

of the IEEE Computer and Communications Societies. Proceedings. IEEE, vol. 2, 2002, pp. 763 – 772.

[77] W.-H. Wang, M. Palaniswami, and S. H. Low, "Application-oriented flow control: Fundamentals, algorithms and fairness," *IEEE/ACM Transactions on Networking*, vol. 14, no. 6, pp. 1282 –1291, December 2006.

[78] Z. Cao and E. Zegura, "Utility max-min: an application-oriented bandwidth allocation scheme," in *IEEE INFOCOM*, vol. 2, March 1999, pp. 793 –801.

[79] J. Kyparisis, "On uniqueness of kuhn-tucker multipliers in nonlinear programming," *Mathematical Programming*, vol. 32, pp. 242–246, 1985.

[80] J. S. P. V. Hutson and M. J. Cloud, *Applications of Functional Analysis and Operator Theory*. Academic Press, 1980.

[81] J. Drakapoulos, "Multi-sigmoidal neural networks and back-propagation," in *International Conference on Artificial Neural Networks*, June 1995, pp. 154 –159.