UNIVERSIDAD DE LA REPÚBLICA

Facultad de Ingeniería

Tesis para optar al Título de

Doctor en Ingeniería Eléctrica

RESOURCE ALLOCATION IN NETWORKS FROM A CONNECTION-LEVEL PERSPECTIVE

(ASIGNACIÓN DE RECURSOS EN REDES DESDE LA PERSPECTIVA DE LAS CONEXIONES)

Autor:

Andrés Ferragut

Director de Tesis:

Dr. Fernando Paganini

Montevideo, Uruguay

2011

UNIVERSIDAD DE LA REPÚBLICA ORIENTAL DEL URUGUAY
INSTITUTO DE INGENIERíA ELÉCTRICA

Los abajo firmantes certificamos que hemos leído el presente trabajo titulado "**Resource allocation in networks from a connection-level perspective**" hecho por **Andrés Ferragut** y encontramos que el mismo satisface los requerimientos curriculares que la Facultad de Ingeniería exige para la tesis del título de **Doctor en Ingeniería Eléctrica**.

Fecha: 30 de agosto de 2011

Director Académico: ______________________
Dr. Pablo Monzón

Director de Tesis: ______________________
Dr. Fernando Paganini

Tribunal examinador: ______________________
Dr. Pablo Belzarena

____
Dr. Matthieu Jonckheere

____
Dr. Ernesto Mordecki

ISSN: XXXX-XXXX (printed version)
ISSN: XXXX-XXXX (electronic version)

Andrés Ferragut (ferragut@ort.edu.uy)

Tesis de Doctorado en Ingeniería Eléctrica
Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay, 2011.

UNIVERSIDAD DE LA REPÚBLICA ORIENTAL DEL URUGUAY

Fecha: **30 de agosto de 2011**

Autor:	**Andrés Ferragut**
Titulo:	**Resource allocation in networks from a connection-level perspective**
Instituto:	**Instituto de Ingeniería Eléctrica**
Grado:	**Doctor en Ingeniería Eléctrica** (**Dr. Ing.**)

Se autoriza a través de la presente a la Universidad de la República Oriental del Uruguay a hacer circular y copiar esta tesis con propósitos no comerciales por requerimientos de individuos o instituciones.

Firma del autor

Contents

Abstract

In this thesis, we analyze several resource allocation problems arising in the study of telecommunication systems. In particular, we focus on data networks, of which the most important example is the global Internet. In such networks, the scarce resource to be allocated is the amount of bandwidth assigned to each ongoing connection. This allocation is performed in real time by the underlying protocols, splitted across several logical layers.

From this point of view, the network can be thought as a large scale control system, where individual entities must follow given control laws in order to find a suitable resource allocation. Since the seminal work of [Kelly et al., 1998], this problem has been expressed in economic terms, through the Network Utility Maximization framework. This formulation proved to be a valuable tool to analyze existing mechanisms and design new protocols that enhance the network behavior, and provided a crucial link between the traditional layer analysis of network protocols and convex optimization techniques, leading to what is called the cross-layer design of networks.

In this work we focus on the analysis of the network from a connection-level perspective. In particular, we study the connection-level performance, efficiency and fairness of several models of network resource allocation. We do so in several settings, both single and multipath, and both wired and wireless scenarios.

We analyze in detail two important problems: on one hand, the resource allocation provided by congestion control protocols whenever multiple connections per user are allowed. We identify problems with the current paradigm of flow-rate fairness, and we propose a new notion of *user-centric fairness*, developing along the way decentralized algorithms that can be implemented at the edge of the network, and drive the system to a suitable global optimum.

The second important problem analyzed here is the resource allocation provided by congestion control algorithms over a physical layer that allows multiple transmission rates, such as wireless networks. We show that the typical algorithms in use lead to important inefficiencies from the connection-level perspective, and we propose mechanisms to overcome these inefficiencies and enhance the resource allocation provided by such networks.

Throughout this work, several mathematical tools are applied, such as convex optimiza-

tion, control theory and stochastic processes. By means of these tools, a model of the system is constructed, and several control laws and algorithms are developed to reach the desired performance objective. As a final step, these algorithms were tested via packet-level simulations of the networks involved, providing validation for the theory, and evidence that they can be implemented in practice.

Resumen

En esta tesis, se analizan varios problemas de asignación de recursos que surgen en el estudio de los sistemas de telecomunicaciones. En particular, nos centramos en las redes de datos, de los cuales el ejemplo más importante es la Internet global. En este tipo de redes, el recurso escaso que debe ser asignado es la cantidad de ancho de banda de cada conexión en curso. Esta asignación se realiza en tiempo real por los protocolos subyacentes, que típicamente se encuentran divididos en varios niveles lógicos o capas.

Desde este punto de vista, la red puede ser pensada como un sistema de control a gran escala, donde cada entidad debe seguir un conjunto dado de leyes de control, a fin de encontrar una asignación adecuada de recursos. Desde el influyente trabajo de [Kelly et al., 1998], este problema se ha expresado en términos económicos, dando lugar a la teoría conocida como *Network Utility Maximization* (maximización de utilidad en redes). Este marco ha demostrado ser una herramienta valiosa para analizar los mecanismos existentes y diseñar protocolos nuevos que mejoran el comportamiento de la red. Proporciona además un vínculo crucial entre el tradicional análisis por capas de los protocolos de red y las técnicas de optimización convexa, dando lugar a lo que se denomina análisis multi-capa de las redes.

En este trabajo nos centramos en el análisis de la red desde una perspectiva a nivel de conexiones. En particular, se estudia el desempeño, eficiencia y justicia en la escala de conexiones de varios modelos de asignación de recursos en la red. Este estudio se realiza en varios escenarios: tanto *single-path* como *multi-path* (redes con un único o múltiples caminos) así como escenarios cableados e inalámbricos.

Se analizan en detalle dos problemas importantes: por un lado, la asignación de los recursos realizada por los protocolos de control de congestión cuando se permiten varias conexiones por usuario. Se identifican algunos problemas del paradigma actual, y se propone un nuevo concepto de *equidad centrada en el usuario*, desarrollando a su vez algoritmos descentralizados que se pueden aplicar en los extremos de la red, y que conducen al sistema a un óptimo global adecuado.

El segundo problema importante analizado aquí es la asignación de los recursos realizada por los algoritmos de control de congestión cuando trabajan sobre una capa física que permite

múltiples velocidades de transmisión, como es el caso en las redes inalámbricas. Se demuestra que los algoritmos usuales conducen a ineficiencias importantes desde el punto de vista de las conexiones, y se proponen mecanismos para superar estas ineficiencias y mejorar la asignación de los recursos prestados por dichas redes.

A lo largo de este trabajo, se aplican varias herramientas matemáticas, tales como la optimización convexa, la teoría de control y los procesos estocásticos. Por medio de estas herramientas, se construye un modelo del sistema, y se desarrollan leyes de control y algoritmos para lograr el objetivo de desempeño deseado. Como paso final, estos algoritmos fueron probados a través de simulaciones a nivel de paquetes de las redes involucradas, proporcionando la validación de la teoría y la evidencia de que pueden aplicarse en la práctica.

Agradecimientos

Como todo trabajo de largo aliento, y vaya si esta Tesis lo fue, cabe agradecer a muchas personas que contribuyeron y pusieron de su parte para que este esfuerzo llegue a buen puerto a lo largo de estos años. En primer lugar, quiero agradecer a los miembros del tribunal de Tesis, por haber dedicado tiempo y esfuerzo a recorrer estas páginas y contribuir con sus comentarios a mejorar la versión final de este documento. En particular, quiero agradecer a Ernesto Mordecki, quien hace ya bastante tiempo me dio el empujón inicial para que oriente mi trabajo hacia la investigación, combinando las herramientas matemáticas con la ingeniería.

Quiero agradecer también a mis compañeros de trabajo a lo largo de todos estos años. Empezando por el núcleo de investigación que formamos y bautizamos "ARTES" hace ya varios años, y donde intentamos plantar la semilla de la investigación matemática aplicada a redes. Muy especialmente, quiero agradecer a Pablo Belzarena por su apoyo constante en su doble rol de orientador y también de jefe, como director del IIE, por haber facilitado tantas cosas. Otro agradecimiento muy especial para Laura Aspirot, con quien las discusiones (académicas o no tanto), y las tardes de trabajo, mate de por medio, siempre fueron enriquecedoras. Más en general, a mis compañeros de la barra del IIE y del IMERL, que siempre estuvieron presentes a lo largo del camino. En particular a Darío Buschiazzo, porque con su constante "¿Samu, cuándo te vas a doctorar?" empujó siempre para llegar a este punto.

Otro agradecimiento especial a mis compañeros del Grupo MATE: Enrique Mallada, Martín López, Marcos Cardozo, Diego Feijer, Juan Pablo Saibene, José García, Fabián Kozynski y Martín Zubeldía. Todos ellos, en mayor o menor medida, contribuyeron con este trabajo. Con ellos he compartido el trabajo diario durante estos años y sus propuestas, ideas y críticas crearon siempre un entorno especial donde llevar adelante la investigación. También a Patricia Corbo, que confió en mí para seguir desarrollando trabajo de investigación en el Uruguay.

Un agradecimiento especial para Pablo Monzón, mi Director Académico, por siempre estar atento y disponible, facilitando muchas cuestiones formales, y sugiriendo y aportando siempre.

Otro agradecimiento muy especial para mi Director de Tesis: Fernando Paganini. No alcanzan las palabras para describir el rol que Fernando tuvo en este trabajo. No fue un orientador más, sino que se arremangó y trabajó constantemente para que este proyecto llegue a buen puerto. Gracias, Fernando, por darme la oportunidad, y porque con tu trabajo y solidaridad permanente aprendí que hacer investigación de calidad es posible desde cualquier lugar. Solo hace falta (como si fuera poco) perseverancia, rigurosidad, mucho estudio y espíritu crítico. Todo esto lo aprendí compartiendo las horas de trabajo durante estos años. Aprendí además, que todo esto puede hacerse sin perder la sonrisa, ni la humildad.

Un agradecimiento también muy especial a mi familia, mis padres Ariel y Alicia, mis hermanos Pablo, Gabriel, Javier y Matías, por su apoyo desde siempre. Y a mi familia ampliada, Cristina, Pablo y Matías que siempre estuvieron presentes dándome fuerza y consejos.

Y por último, el agradecimiento más importante es para la coautora espiritual de esta tesis: Natalia, nunca habría llegado hasta acá sin vos. Nunca sería lo mismo sin vos. Siempre creíste en mí aún en los momentos más difíciles de este camino. Me empujaste a perseguir mis metas, y me enseñaste a disfrutarlas, y sobre todo a disfrutarlas juntos. Me enseñaste que el camino importa, y que el camino es más lindo recorrerlo juntos, y con una sonrisa. Ojalá que las horas (o días, o meses) separados por esta empresa, no empañen la construcción permanente que día a día hacemos de nuestra vida juntos, sino que sirva para hacerla más fuerte, y sobre todo, más linda de recorrer entre los dos.

1

Introduction

The Internet has evolved from a loose federation of academic networks to a global entity spanning millions of end users, equipments and interconnecting technologies. The exponential growth of available communication devices and applications have revolutionized human communication, commerce and computing.

Early in this evolution, it was recognized that unrestricted access to network resources resulted in poor performance, mainly because of high packet loss rates and the resulting low network utilization. Users transferring data over the network in an open loop fashion will inevitably lead, as the network grows, to what is called "congestion collapse".

This phenomenon, first observed in the mid 1980s, brought forward the need of studying the *resource allocation* that the network shall provide to its users. The main challenge is to allow data flows over the network to regulate themselves in order to find an *efficient* and *fair* operating point. The solution calls for feedback mechanisms that allow the end users to regulate their demands based on some congestion signals provided by the network. Moreover, and due to the scale of the Internet, it is necessary to have *decentralized* control algorithms, that is, we cannot assume that a central dispatcher will provide the resource allocation on the fly, based on current demand. Such an approach would require an unaffordable amount of computing and signalling resources.

The first step in this direction came with the work of [Jacobson, 1988], in which the author proposed to add a congestion control mechanism to the Transmission Control Protocol (TCP) which was and is used on the Internet to transfer data files. The algorithm was based on the sliding window mechanism that was already in use for *flow control*, but with the possibility to adapt the window size based on network congestion. The main assumption is that packet losses in a given communication were caused by some resources of the network becoming congested, and thus upon detection of a lost segment in the data transfer, the sender must restrain from sending too much data, by reducing its transmission window. The proposed algorithm, which we will describe later, as well as its successive updates have been undoubtedly successful in steering the traffic of the Internet through its global expansion.

The importance of the congestion control problem has attracted a large research community. The main breakthrough was the formulation in the seminal work of [Kelly et al., 1998]. Kelly poses the congestion control problem as a suitable convex optimization problem, where connections are represented as economic actors in a "bandwidth market". The network congestion signals are interpreted as *prices*, and the market equilibrium becomes the resource allocation obtained through the decentralized mechanisms. The resulting formulation has become known in the research community as the Network Utility Maximization (NUM) problem.

Throughout this work, we shall apply the NUM framework to analyze several important problems of network resource allocation. One of the main issues with economic models for network congestion control is the fact that the "utility" a connection derives in the aforementioned "bandwidth market" is associated with the protocol behavior. It is thus only a metaphorical description, not related with real user willingness for bandwidth. In general, typical users are not aware of this, because the congestion control algorithms are deep inside the operating systems of their hosts. However, several applications started to "cheat" the bandwidth allocation by opening multiple connections.

If a user or entity is allowed to open several connections, through the same or multiple paths, the resource allocation problem becomes more complicated. We shall show in particular that the end user has incentives to open multiple connections, thus leading to potential inefficiencies. This problem motivates the first part of this work, which focuses on finding a suitable NUM problem where utilities reflect the real user preferences, and where we develop decentralized algorithms that allow the system to operate in a fair and efficient operating point. We developed two kind of algorithms, cooperative and non cooperative. The first ones are based on end systems having fine-grained control of the number of connections, and cooperating among them to reach the common goal of maximizing network total utility. The second class of algorithms focuses on the case where users may not be cooperative, and the network must protect itself from greedy users by performing admission control. In our analysis, both single-path and multi-path situations are considered, thus modelling different types of user connections across the network.

Another recent development of networks is the ongoing growth of wireless local area networks (WLAN). In the last decade, the technology to provide wireless Internet connectivity has become ubiquitous. In particular, WLANs are often the final hop in network communication, providing connectivity to home and office users, college campuses, and even recreational areas. This growth of wireless networks has been accompanied by research on the resource allocation provided by such systems. In particular, the NUM framework has been applied often to model different types of wireless *Medium Access Control* (MAC) layers. Moreover, extensive research has been devoted to the analysis of collisions and its interaction with the above transport protocols and congestion control.

In the second part of the thesis, we focus on the resource allocation of wireless networks, with a special emphasis on the effect of having multiple transmission rates, which is a common feature in the technologies involved. The fact that multiple transmission rates coexist in the same cell has been often overlooked, with the main focus being placed on collisions. In our work, we analyze the impact of having multiple rates, showing that it can lead to important inefficiencies, not explained by collisions alone. We extend the NUM theory to include the multiple rates found in WLANs and analyze its interaction with the upper layer protocols, in particular congestion control, and how the resource allocation affects the connection level performance. We propose mechanisms to enhance the current resource allocation in order to improve the efficiency of these networks, through a simple and decentralized packet level algorithm. We also analyze the performance of the algorithms in single cell and mixed wired-wireless scenarios, which are common deployments in practice.

1.1 Outline of this work and main contributions

The document is organized as follows. In Chapter 2 we introduce the reader to congestion control in the Internet, and the NUM framework used to model it. This chapter presents all the main results in the literature related to congestion control and optimization. In particular, the different approaches to the resource allocation problem are presented, and we also describe the algorithms used in practice to perform congestion control, and how they relate to optimization algorithms.

The thesis is then split into two main parts. Part I introduces the *user-centric* notion of fairness we propose to enhance network resource allocation. In Chapter 3 we describe the models used for congestion control when multiple connections are involved, and we prove that the end-user has incentives to increase its number of connections to get more bandwidth. These results motivate the analysis of Chapter 4, where our notion of user-centric fairness is defined, in order to overcome this limitation of current protocols. We also describe how to achieve this notion of fairness by suitable modifications of standard congestion control algorithms, provided that the end user is capable of controlling the aggregate rate of its connections. We also analyze how to achieve the proposed resource allocation by using the number of ongoing connections as a control variable, which is clearly more practical. This analysis assumes cooperation between users. In Chapter 5 we lift this hypothesis by moving the function of connection-level control to the network. We develop decentralized algorithms for connection admission control that drive the system to the desired allocation. The analysis of this chapter is mainly based on fluid limits for the underlying stochastic processes. In Chapter 6 we include the possibility of routing at the connection-level timescale, and analyze the achievable stability region for these kind of algorithms, generalizing previous results. Throughout these chapters, fluid model and packet level simulations are presented to illustrate the behavior of the algorithms in practice. Conclusions of this part are given in Chapter 7.

In Part II, we analyze the resource allocation problem in wireless local area networks, focusing specifically on the multirate capabilities of these technologies. In Chapter 8, we develop a model for standard congestion control on top of a multirate wireless network, showing that the resulting allocation can be described through a NUM problem. However, this allocation is highly biased against the users with better rates, thus leading to important inefficiencies. This motivates the analysis of Chapter 9, where a new NUM problem is proposed, more suited to the particular nature of these networks. Also, packet level algorithms are developed in order to drive the system to this more efficient allocation. These algorithms are decentralized and can be implemented within current technologies. We also analyze the possibility of extending the notion of fairness of this NUM problem to the case of multiple hop wireless and wired networks, in particular providing an important generalization of the NUM framework to include both cases simultaneously. In Chapter 10 we turn our attention to the connection level timescale, and develop models that enable us to determine the stability region and performance of these systems when both the congestion control and the lower medium access control layer are taken simultaneously into account. Stability results and performance metrics are determined for both the current allocation and the proposed enhancements. In Chapter 11, we apply the results to the important case of IEEE 802.11 networks, commonly known as WiFi. These networks, which are now present in most deployments around the world, provide example of the inefficiencies described. We show how the developed algorithms can be applied in such a setting, and provide simulation examples of these algorithms in practice. Conclusions of this part are given in Chapter 12. Finally, in Chapter 13 we present the main conclusions of this work and describe the future lines of research. Appendix A reviews some of the mathematical preliminaries used in the developments of this thesis.

1.2 Associated publications

Below, we list the publications associated with this thesis:

- A. Ferragut and F. Paganini, *"Achieving network stability and user fairness through admission control of TCP connections"*, in Proceedings of the 42nd Annual Conference on Information Sciences and Systems (CISS 2008), Princeton, NJ, USA, 19-21 March 2008.

- A. Ferragut and F. Paganini, *"Utility-based admission control: a fluid limit analysis"*, in Stochastic Networks Conference, Paris, France, June 2008.

- A. Ferragut and F. Paganini, *"A connection level model for IEEE 802.11 cells"*, in Proceedings of the 5th. IFIP/ACM Latin American Networking Conference (LANC'09), Pelotas, Brazil, September 2009.

- A. Ferragut and F. Paganini, *"User-centric network fairness through connection-level control"*, in Proceedings of the 29th IEEE Infocom Conference, San Diego, USA, March 2010.

- A. Ferragut, J. García and F. Paganini, *"Network utility maximization for overcoming inefficiency in multirate wireless networks"*, in Proceedings of the 8th Intl. Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt'10), Avignon, France, May-June 2010.

- A. Ferragut and F. Paganini, *"Connection-level dynamics in networks: stability and control"*, in Proceedings of the 49th IEEE Conference on Decision and Control, Atlanta, USA, December 2010.

- A. Ferragut and F. Paganini, *"Resource allocation over multirate wireless networks: a Network Utility Maximization perspective"*, in Computer Networks, **55** pp. 2658-2674, 2011.

- A. Ferragut and F. Paganini, *"Network resource allocation for users with multiple connections: fairness and stability"*. Submitted to IEEE/ACM Transactions on Networking.

2

Resource allocation and congestion control in networks

In this chapter, we will present the main ideas behind the Network Utility Maximization (NUM) framework, which serve as a basis for the development of the contributions of the thesis. The NUM theory has evolved since the work of [Kelly et al., 1998] and is nowadays used to model different problems in networking. Below we introduce the reader to the main results and we establish some important definitions and notations used throughout this work. A main reference for NUM theory is [Srikant, 2004].

2.1 Resource allocation as an optimization problem

We begin by a simple example that illustrates the relationship between resource allocation and optimization, and provides intuition about some of the concepts involved later. The basic results on convex optimization theory that we shall use are summarized in Appendix A.1.

Consider a set of *entities*, which we may call "users" and we index by i. These users want to share some scarce resource. Assume that the total available quantity of the resource is limited by some amount $c > 0$. If a given user i is allocated a given amount x_i of the total resource, it derives a *utility* $U_i(x_i)$.

From now on, we make the following assumption on utility functions:

Assumption 2.1. *The utility function $U_i(\cdot)$ is a continuously differentiable, non decreasing, strictly concave function $U_i : \mathbb{R}^+ \to \mathbb{R}$.*

The utility function expresses the benefit a user obtains from the resources. The non decreasing hypothesis states simply that the user is "happier" whenever more resources are allocated to it. The concave assumption is standard, and expresses the fact the amount of *marginal* utility provided by extra resources is decreasing.

A *resource* allocation is a vector $x = (x_i)$, with $x_i \geqslant 0$, that indicates the amount of resources given to each one of the users. Since the total amount is limited by c, a resource allocation would be *feasible* if and only if:

$$\sum_i x_i \leqslant c. \tag{2.1}$$

Clearly, there are several possible resource allocations. We are interested in finding a resource allocation that gives an optimal trade-off between all the participating entities. A reasonable choice is to maximize the *social welfare*, which is given by the following convex optimization problem:

Problem 2.1.

$$\max_x \sum_i U_i(x_i)$$

subject to the constraint:

$$\sum_i x_i \leqslant c.$$

The above problem is the optimization of a concave function, due to Assumption 2.1, subject to the linear (and therefore convex) inequality constraint (2.1). Applying the techniques of Appendix A.1, we write the Lagrangian of the problem, which is:

$$\mathscr{L}(x,p) = \sum_i U_i(x_i) - p\left(\sum_i x_i - c\right),$$

where p denotes the Lagrange multiplier associated with the constraint.

The Karush-Kuhn-Tucker optimality conditions for this problem state that any solution (x^*, p^*) must satisfy:

$$\left.\frac{\partial \mathscr{L}}{\partial x_i}\right|_{x=x^*} = U_i'(x_i^*) - p^* = 0,$$

$$p^*\left(\sum_i x_i^* - c\right) = 0,$$

where $p^* \geqslant 0$, and x^* satisfies the constraint (2.1).

We can discuss two cases. In the first case, optimality is achieved with strict inequality in the constraint, i.e. $\sum_i x_i^* < c$. In this case, the second condition above implies that $p^* = 0$.

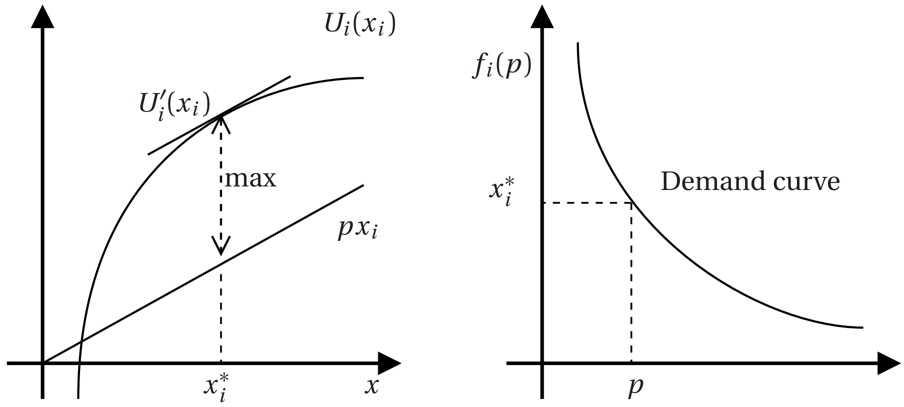

FIGURE 2.1: USER UTILITY AND DEMAND CURVE.

This happens when there is a point in the interior of the region defined by (2.1) where all users achieve maximal individual satisfaction. In this case, we say that the constraint is not *active* and the associated Lagrange multiplier will be 0.

A more interesting case is when all users cannot be fully satisfied, and some trade-off must occur. This is the case when all the utilities are increasing and strictly concave, as in Assumption 2.1. In this situation we will have $\sum_i x_i^* = c$ and $p^* > 0$, with the resulting allocation satisfying:

$$U_i'(x_i^*) = p^* \quad \forall i. \tag{2.2}$$

The above condition can be interpreted in economic terms, as we shall see below. Consider for the moment that some external entity is available, that has the power to fix a *price* p for the usage of the scarce resource. In such a situation, a user allocated an amount x_i of the resource will perceive a utility $U_i(x_i)$, but will have to pay px_i for its usage. A rational user would choose to individually make the following optimization:

$$\max_{x_i} U_i(x_i) - px_i.$$

Note that, since U_i is differentiable, the maximum above is attained by choosing x_i such that $U_i'(x_i) = p$, or equivalently:

$$x_i = f_i(p),$$

where $f_i(p) = (U')^{-1}(p)$ is called the user *demand curve*. For a given price p, $f_i(p)$ is the optimal amount of resource a rational user is willing to purchase. This interpretation is depicted in Figure 2.1. The demand curve $f_i(p)$ is a decreasing function of the price, as expected.

The economic interpretation of equation (2.2) is now clear: assume that, given a fixed price for the shared resource, every user purchases its own optimal amount at this price. Then, the KKT optimality conditions ensure that there exists an optimal price $p^* > 0$, which is exactly the Lagrange multiplier for the system, such that the overall allocation will be optimal for the original resource allocation Problem 2.1. By choosing this optimal price, we can decentralize the solution of the problem, since each user will purchase the amount of resources dictated by its demand curve, but the overall allocation will satisfy the KKT optimality conditions of the resource allocation Problem 2.1, and therefore we achieve maximal total welfare.

The key to solving the above problem in a decentralized way is therefore finding the right price for the resources. If the utility functions of each user are known, we can readily find the optimum of Problem 2.1, but in that case there would not be any real advantage from the decentralization property. A simpler way to achieve this is to adapt the price by the following dynamics:

$$\dot{p} = \sum_i x_i - c,$$

with adequate provisions to prevent the price from becoming negative. Note that the above equation simply states that the price must be increased whenever the offered resource cannot satisfy the current total demand, and lowered when there is an excess of resources. This algorithm is called the *dual* algorithm and we shall explore it later in more general settings. The decentralization property applies also to the external entity choosing the price, which does not have to know the individual user utilities, neither the amount of resource purchased by each one of them. It just needs to know the *total demand* for the resource in order to adapt the price accordingly.

The simple example presented here illustrates the main concepts and intuition behind the economic interpretation of resource allocation. We now move on to rate allocation in telecommunication networks, where the main ideas remain the same, but the constraints involving the offered resources become more elaborate.

2.2 Rate allocation in communication networks

Consider now a network composed of shared *resources* which are the network links. We index these links by l and associate with them a *capacity* c_l, namely the amount of data they can carry in bits per second (bps). Over this network, we want to establish connections that can use several links along a path or route denoted by r. For the moment, we consider that each connection uses a single set of resources or route, so we can also use r to identify the connection. We shall use the notation $l \in r$ if link l belongs to route r.

The resource allocation problem consists in determining a set of *rates* x_r at which each connection can transmit data, taking into account the *capacity constraints* of the network, that is, the total amount of traffic carried by each link must not exceed the link capacity c_l. However, the allocation should be *efficient*, i.e. it should carry as much traffic as possible across

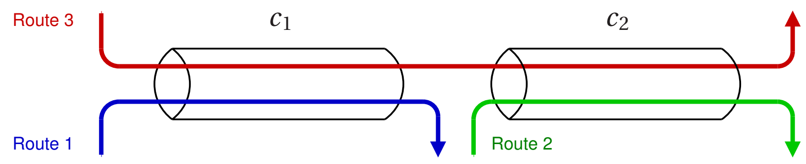

FIGURE 2.2: PARKING LOT NETWORK OF EXAMPLE 2.1.

the network, without violating the capacity constraints. Moreover, we want this allocation to be *fair*, i.e. it does not starve some users in benefit of some other ones.

For a given network, there are multiple possible resource allocations. It turns out that the two objectives stated before can be conflicting. To illustrate this we consider the following example, which will be used many times along this work.

Example 2.1 (Linear "parking-lot" network). Consider a network with two links of capacities c_1 and c_2 and three routes. Connections 1 and 2 travel through a single link, while connection 3 uses both resources. This network is depicted in Figure 2.2.

A suitable resource allocation $x = (x_1, x_2, x_3) \in \mathbb{R}^3_+$ must satisfy the network capacity constraints, which are:

$$x_1 + x_3 \leqslant c_1$$

$$x_2 + x_3 \leqslant c_2$$

Among the multiple possible allocations, we want to leave out those that under-utilize resources. We have the following definition:

Definition 2.1 (Pareto efficient allocation)**.** *For a given network, a feasible resource allocation $x \in \mathbb{R}^+_n$ is called* Pareto efficient *if and only if for any other feasible resource allocation $y \neq x$ we have:*

$$\textit{If } y_r > x_r \textit{ for some } r \Rightarrow \exists s : y_s < x_s. \tag{2.3}$$

The rationale behind the above definition is that, we cannot increase the rate allocated to some route without decreasing the rate allocated to another one.

An alternative characterization of Pareto efficiency in the case of networks is given by the following condition:

$$\forall r, \exists l \in r : \sum_{s:l \in s} x_s = c_l, \tag{2.4}$$

that is, each route traverses a saturated link.

We now present two possible allocations for the network of Example 2.1.

Example 2.2 (Max-throughput allocation). In the context of Example 2.1, consider the allocation $x_1 = c_1$, $x_2 = c_2$, $x_3 = 0$. This allocation is the most efficient in terms of total network throughput, however it is extremely unfair, starving the long route completely.

Example 2.3 (Max-min fair allocation). Consider the same network and assume that $c_1 \leqslant c_2$. A possible allocation is:

$$x_1 = x_3 = \frac{c_1}{2},$$
$$x_2 = c_2 - \frac{c_1}{2}.$$

This allocation is *fair* in a sense we will define clearly below. Note that the most congested resource is equally split between its two routes, while route 2 is allocated the remaining capacity in the second link. Note also that the total throughput obtained by this allocation is strictly less that the one in Example 2.2.

The above example illustrates the concept of *max-min fairness*, which is a notion borrowed from the economic theory, and introduced by [Bertsekas and Gallager, 1991] in the context of networks. Below is the formal definition.

Definition 2.2 (Max-min fairness). *A resource allocation $x = (x_r)$ is* max-min fair *if, for any other allocation y that satisfy the capacity constraints of the network, the following is true: if $y_r > x_r$ for some route r, then there exists another route s such that $x_s \leqslant x_r$ and $y_s < x_s$.*

In other words, when the allocation is max-min fair, any other allocation that increases the rate given to some route (r) must also strictly decrease the rate allocated to some other route s that was already in a poorer condition to begin with.

An alternative characterization of max-min fairness is the following proposition, whose proof can be found in [Bertsekas and Gallager, 1991, Section 6.5]

Proposition 2.1. *An allocation x is max-min fair if and only if for every r there exists a link l such that:*

$$\sum_{s:l\in s} x_s = c_l \quad \textit{and} \quad x_r \geqslant x_s \; \forall s : l \in s,$$

i.e., each route traverses a saturated link, and in that link it is among the ones that use most of the resources.

It is easy to see that in Example 2.3, the above condition is verified.

In order to generalize the above allocations, [Kelly, 1997] introduced the following framework, which is called Network Utility Maximization (NUM), and we will use it as a base for the analysis in the rest of our work.

Consider that each connection in the network derives a *utility* or benefit from the rate allocated to it, which we characterize by a function $U_r(x_r)$. In the rest of our work, unless otherwise stated, we assume that utilities satisfy Assumption 2.1.

In order to summarize the network capacity constraints for the general case, we make use of the following:

Definition 2.3 (Routing matrix)**.** *We call the* routing matrix *the matrix $R = (R_{lr})$ whose entries satisfy:*

$$R_{lr} = \begin{cases} 1 & \text{if } l \in r \\ 0 & \text{otherwise.} \end{cases}$$

With this notation, the total rate y_l that goes through link l verifies:

$$y_l = \sum_{r:l\in r} x_r = \sum_r R_{lr} x_r,$$

which can be written in matrix notation as:

$$y = Rx,$$

where $y = (y_l)$ is the vector of link rates.

We are now in position to state the Network Utility Maximization problem:

Problem 2.2 (Network Problem)**.** *Given a network composed of links of capacity $c = (c_l)$, a routing matrix R and where each source has a utility function $U_r(\cdot)$ allocate the resources such that the following optimum is attained:*

$$\max_{x_r \geqslant 0} \sum_r U_r(x_r)$$

subject to:

$$Rx \leqslant c,$$

where the last inequality is interpreted componentwise.

The resulting allocation is unique due to Assumption 2.1 and the fact that the capacity constraints are linear. The solution will be such that the total utility or *social welfare* of the connections is maximized without violating the capacity constraints. Note that under the above assumptions, the resulting allocation will be automatically Pareto-efficient, since if it is not the case, we can increase total utility by increasing the rate of the flow that is not saturated in condition (2.4).

By choosing different utility functions, we can model different resource allocations. An interesting parametric family of utility functions, proposed by [Mo and Walrand, 2000], is the following:

Definition 2.4 ($\alpha-$family of utility functions)**.** *Let $\alpha_r \geqslant 0$ be a parameter, and $w_r > 0$ a weight, the following function is called $\alpha-$*fair utility*:*

$$U_r(x_r) = \begin{cases} w_r \frac{x_r^{1-\alpha_r}}{1-\alpha_r} & \alpha_r \neq 1, \\ w_r \log(x_r) & \alpha_r = 1. \end{cases}$$

Note that the above functions satisfy $U_r'(x_r) = w_r x_r^{-\alpha_r}$ for all $\alpha_r \geqslant 0$. When all connections share the same $\alpha_r = \alpha$, we call the allocation resulting from Problem 2.2 the *(weighted) α−fair allocation.*

Note also that, for $\alpha = 0$, the resulting allocation if the one that maximizes network total throughput, as in Example 2.2. However, this choice of utility function is not strictly concave, as required by Assumption 2.1. For $\alpha \to \infty$ we have the following:

Proposition 2.2 ([Mo and Walrand, 2000])**.** *If $w_r = w > 0\ \forall r$, the α−fair resource allocation tends to the max-min fair allocation as $\alpha \to \infty$.*

If the weights are different for each route, the limit allocation is called weighted max-min fairness.

The role of the parameter α is to enable us to consider different intermediate situations between the maximum efficiency ($\alpha \to 0$) and maximum fairness ($\alpha \to \infty$). An important case is $\alpha = 1$ which is called *(weighted) proportional fairness* by [Kelly et al., 1998].

To futher analyze the properties of the resource allocation given in Problem 2.2, we will make use of Lagrangian duality (see Appendix A.1). Let $p = (p_l)$ denote the Lagrange multipliers for the link capacity constraints, then the Lagrangian of Problem 2.2 is:

$$\mathscr{L}(x,p) = \sum_r U_r(x_r) - \sum_l p_l \left(\sum_r R_{lr} x_r - c_l \right).$$

The Karush-Kuhn-Tucker (KKT) conditions for optimality in this problem are:

$$\frac{\partial \mathscr{L}}{\partial x_r} = U_r'(x_r) - \sum_l R_{lr} p_l = 0 \quad \forall r,$$
$$p_l \left(\sum_r R_{lr} x_r - c_l \right) = 0 \quad \forall l,$$

where as usual we assume that x is feasible and $p_l \geqslant 0$.

With the economic interpretation described in Section 2.1 in mind, we shall call p_l the *link price.* It is also convenient to rewrite the preceding equations in terms of *link rates* and *route prices.* Recall that $y_l = \sum_r R_{lr} x_r$ is the link rate. Also we define:

$$q_r = \sum_l R_{lr} p_l$$

as the *route price* for route r. Due to the definition of the routing matrix, it is simply the sum of the Lagrange multipliers of the link the route traverses. In matrix notation we have:

$$q = R^T p,$$

where ^{T} denotes matrix transpose.

With these notations, the KKT conditions can be rewritten as:

$$U_r'(x_r) - q_r = 0 \quad \forall r, \tag{2.5a}$$

$$p_l(y_l - c_l) = 0 \quad \forall l. \tag{2.5b}$$

Equation (2.5a) can be interpreted as a *demand curve*. The bandwidth allocated to route r must be such that the marginal utility of the route matches this route price. We shall also write this equation as $x_r = f_r(q_r)$ being $f_r(\cdot) = U_r'^{-1}(\cdot)$, with f_r being a decreasing function due to Assumption 2.1. Equation (2.5b) is the *complementary slackness* condition that tells that either resource l is saturated, or its price must be 0. Since the objective function is concave and the constraints are convex, the conditions (2.5) are necessary and sufficient for optimality (c.f. Appendix A.1), and thus characterize the resource allocation.

To illustrate the different resource allocations defined up to now, we show a numerical example based on the network of Example 2.1.

Example 2.4 ($\alpha-$fairness in a parking lot network). Consider the setting of Example 2.1, and take $c_1 = 10$ and $c_2 = 5$. Suppose the utilities come from the $\alpha-$family with the same α, i.e. $U_r'(x_r) = x_r^{-\alpha}$. The KKT conditions for this problem are:

$$x_1^{-\alpha} = q_1 = p_1, \quad x_2^{-\alpha} = q_2 = p_2, \quad x_3^{-\alpha} = q_3 = p_1 + p_2,$$
$$p_1(x_1 + x_3 - c_1) = 0, \quad p_2(x_2 + x_3 - c_2) = 0.$$

Since $x^{-\alpha} > 0$ we must have $p_1, p_2 > 0$ and thus the links will be saturated. We can reduce the above equations to:

$$p_1^{-1/\alpha} + (p_1 + p_2)^{-1/\alpha} = c_1,$$
$$p_2^{-1/\alpha} + (p_1 + p_2)^{-1/\alpha} = c_2.$$

Solving the above equations we find the link prices for different values of α. Then, setting $x_i = q_i^{-1/\alpha}$ we can find the resource allocation. In Table 2.1 we illustrate the results for different values of α. As we mentioned before, the cases $\alpha \to 0$ and $\alpha \to \infty$ correspond to the max-throughput and max-min allocations respectively.

So far, we have seen how different resource allocations schemes can be achieved with different choices of utility functions. If a global entity has access to the utility functions of all the users and knows the capacities of all the links, the resource allocation could be calculated by solving the optimization problem presented above. However, this is clearly unreasonable since this requires a centralized knowledge of all the parameters of the network. A more reasonable set of assumptions is the following:

- Each connection knows its own utility function.

α	x_1	x_2	x_3	p_1	p_2
0	10	5	0	1	1
1	7.87	2.89	2.11	.13	.35
2	7.57	2.57	2.43	1.75×10^{-2}	.15
5	7.50	2.50	2.50	4.21×10^{-5}	1.02×10^{-2}
∞	7.50	2.50	2.50	undef.	undef.

TABLE 2.1: $\alpha-$ FAIR ALLOCATIONS FOR THE NETWORK OF EXAMPLE 2.1 FOR DIFFERENT VALUES OF α.

- Each network link knows the total input rate.
- There is a protocol that allows the network to convey some information about the resource congestion on a given route to the connections along this path.

In the next Section we will present several algorithms that solve the resource allocation problem precisely, and verify the assumptions above. The key idea in all of these algorithms is that each link computes the Lagrange multipliers involved in its capacity constraint, an the source reacts by adapting its transmission rate to the sum of these link prices.

2.3 Congestion control as a decentralized optimization algorithm

The discussion of the preceding Section emphasizes the need of *decentralization*. As we shall see, the particular structure of the network Problem 2.2 enables us to use some simple algorithms to solve it. Below we discuss three different approaches from a theoretical perspective and then we proceed to relate them to current congestion control algorithms used in practice.

2.3.1 The primal algorithm

Consider that each link in the network is able to measure its input traffic y_l and can generate a *congestion price* according to some static map $y_l \mapsto f_l(y_l)$. We assume that $f_l(\cdot)$ is a non-decreasing function such that the following condition holds:

$$\int_0^y f_l(u)du \to \infty \quad \text{as } y \to \infty.$$

The functions $f_l(\cdot)$ play the role of a *penalty function*. Intuitively, the value of $f_l(\cdot)$ must grow quickly when the link capacity constraint is violated, i.e. as soon as $y_l > c_l$.

Consider now the following unconstrained optimization problem in the positive orthant:

Problem 2.3.

$$\max_{x_r \geqslant 0} \sum_r U_r(x_r) - \sum_l \int_0^{y_l} f_l(u)du,$$

where as before $y = Rx$.

Under the assumptions for U_r and f_l, the objective function is strictly concave and thus Problem 2.3 is a convex optimization problem. Under some mild extra assumptions on the penalty functions, the optimum of Problem 2.3 lies in the interior of the positive orthant.

The first order optimality conditions of Problem 2.3 are:

$$U_r'(x_r) - \sum_l R_{lr} f_l(y_l) = 0 \quad \forall r.$$

Identifying $p_l = f_l(y_l)$ we have that, as in the previous Section, the optimum is attained when the marginal utility of all users matches the sum of the link prices over those links traversed by the route. Note that these link prices are determined by the penalty function, and do not necessarily coincide with the Lagrange multipliers defined before. It is however convenient to overload the notation in this way, since they act as a price variable.

As before, with centralized information of all the utilities and link price functions of the network, a global entity could be capable of solving the optimal allocation. However, we would like to define an adaptation algorithm that enables the sources to find the optimum without the need of centralized information. Consider now the following dynamics for the source rates:

$$\dot{x}_r = k_r \left(U_r'(x_r) - q_r \right),$$

where as before $q_r = \sum_l R_{lr} p_l$ and $k_r > 0$ is a constant (step size). This is a gradient search algorithm, where the sources try to follow the direction of the gradient (with respect to x) of the objective function. The complete dynamics are:

$$\dot{x}_r = k_r \left(U_r'(x_r) - q_r \right) \quad \forall r, \tag{2.6a}$$

$$y = Rx, \tag{2.6b}$$

$$p_l = f_l(y_l) \quad \forall l, \tag{2.6c}$$

$$q = R^T p. \tag{2.6d}$$

Equations (2.6) define what is called the *primal algorithm*. Its name comes from the fact that adaptation is performed on the primal variables of Problem 2.2. Note that the equilibrium of (2.6) satisfies the optimality conditions of Problem 2.3. Moreover, we have the following:

Proposition 2.3 ([Kelly et al., 1998])**.** *The equilibrium of* (2.6) *is the optimum of Problem 2.3. This equilibrium is globally asymptotically stable.*

Note that this algorithm is decentralized: each source must adapt its rate following its own utility function and with the knowledge of the prices *along its route*. Links generate prices according to some static map of its own utilization, irrespective of which routes pass through. Therefore, the algorithm satisfies the hypotheses stated at the end of the previous Section. The main drawback is that, since prices are statically generated, the allocation attained is the solution of Problem 2.3, which is only an approximation of Problem 2.2.

In Section 2.4 we shall show that the typical TCP congestion control algorithm can be modelled as an implementation of this primal algorithm. In the next Section we shall show another algorithm that allows us to solve exactly Problem 2.2.

2.3.2 The dual algorithm

In order to solve Problem 2.2 exactly we must adapt not only the source rates but also the link prices. To derive a suitable algorithm, we turn to results on convex optimization theory. Consider again the Lagrangian of Problem 2.2:

$$\mathcal{L}(x,p) = \sum_r U_r(x_r) - \sum_l p_l \left(\sum_r R_{lr} x_r - c_l \right). \tag{2.7}$$

The dual function of Problem 2.2 consists on maximizing $\mathcal{L}(x,p)$ over x for fixed p. It is convenient to rewrite (2.7) as:

$$\mathcal{L}(x,p) = \sum_r \left(U_r(x_r) - q_r x_r \right) + \sum_l p_l c_l. \tag{2.8}$$

It is now clear that, to maximize $\mathcal{L}$ with respect to x, the routes must choose x in order to maximize its *surplus*:

$$x_r : \max_{x_r} U_r(x_r) - q_r x_r.$$

Under Assumption 2.1, this leads to the choice $x_r = U'^{-1}(q_r)$. The corresponding surplus is then:

$$S_r(q_r) = U_r(U_r'^{-1}(q_r)) - q_r U_r'^{-1}(q_r),$$

and the dual function is:

$$\mathcal{D}(p) = \sum_r S_r(q_r) + \sum_l p_l c_l. \tag{2.9}$$

The optimum of Problem 2.2 can be calculated by minimizing $\mathcal{D}(p)$ over $p \geqslant 0$ and choosing x_r accordingly. Since the problem is convex, there will be no duality gap (c.f. Appendix A.1). To find a suitable algorithm, we can make a gradient descent in the dual function. Consider the derivative:

$$\frac{\partial \mathcal{D}}{\partial p_l} = \sum_r \frac{\partial S(q_r)}{\partial p_l} + c_l. \tag{2.10}$$

By using the inverse function theorem we can compute:

$$\frac{\partial S_r(q_r)}{\partial q_r} = -x_r,$$

and thus:

$$\frac{\partial S(q_r)}{\partial p_l} = -x_r \frac{\partial q_r}{\partial p_l} = -x_r R_{lr}.$$

Substituting in (2.10) we get:

$$\frac{\partial \mathscr{D}}{\partial p_l} = c_l - \sum_r R_{lr} x_r = c_l - y_l.$$

Therefore, minus the gradient of the dual function measures the amount each capacity constraint is violated. A gradient descent algorithm will follow the opposite direction of the gradient. The complete dynamics are as follows:

$$x_r = U_r'^{-1}(q_r) \quad \forall r, \tag{2.11a}$$

$$y = Rx, \tag{2.11b}$$

$$\dot{p}_l = \gamma_r (y_l - c_l)^+_{p_l}, \tag{2.11c}$$

$$q = R^T p, \tag{2.11d}$$

where $\gamma_r > 0$ is a constant (step size). The function $(\cdot)^+_p$ is called *positive projection* and is simply a way to prevent prices from becoming negative. It has the following definition:

$$(x)^+_p = \begin{cases} x & \text{if } p > 0 \text{ or } p = 0 \text{ and } x > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{2.12}$$

The algorithm in (2.11) is appropriately called the *dual algorithm* and its convergence was established in the following result:

Proposition 2.4 ([Paganini, 2002])**.** *Assume that the matrix R has full row rank, i.e., given q there exists a unique p such that $q = R^T p$. Under this assumption, the dual algorithm* (2.11) *is globally asymptotically stable, and its equilibrium is the unique optimum of Problem 2.2.*

The dual algorithm is also decentralized in the sense discussed at the end of the previous Section, and moreover solves Problem 2.2 exactly. In Section 2.4 we will show that this class of algorithms model the behavior of *delay-based* congestion control protocols in the Internet.

2.3.3 The primal-dual algorithm

We can combine the previous ideas into a single algorithm, where the connections adapt their rate to a current route congestion price and the links generate their prices dynamically.

The algorithm is then called *primal-dual* and it was introduced in the context of networks by [Alpcan and Basar, 2003, Wen and Arcak, 2003]. The closed loop dynamics in this case are:

$$\dot{x}_r = k_r(U_r'(x_r) - q_r) \quad \forall r, \tag{2.13a}$$

$$y = Rx, \tag{2.13b}$$

$$\dot{p}_l = \gamma_l (y_l - c_l)^+_{p_l} \quad \forall l, \tag{2.13c}$$

$$q = R^T p. \tag{2.13d}$$

We have the analogous convergence result:

Proposition 2.5 ([Wen and Arcak, 2003]). *Assume that the matrix R has full row rank. Under this assumption, the dynamics of* (2.13) *are globally asymptotically stable, and its equilibrium is the unique optimum of Problem 2.2.*

Remark 2.1. In the algorithms presented above, we assumed that $k_r > 0$ and $\gamma_l > 0$ are constants. The same results are valid if we take $k_r(x_r) > 0$ where $k_r(\cdot)$ is a continuous function. Analogously, we can take $\gamma_l(p_l) > 0$.

All the previous algorithms can be summarized in the following diagram, which shows the feedback structure of the control loop.

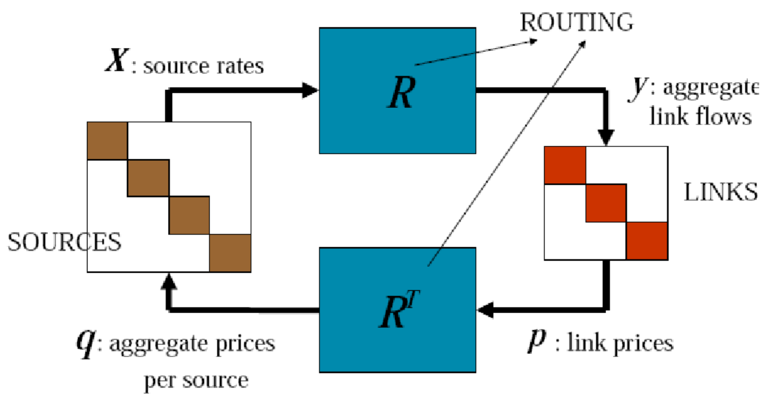

FIGURE 2.3: CONTROL LOOP FOR THE DISTRIBUTED CONGESTION CONTROL ALGORITHMS.

The diagonal blocks in the diagram represent the decentralized control laws, followed by every source and every link in the network. The different sources and links are interconnected via the routing matrix R, which allows us to map connection rates into link rates, and link prices to route prices.

So far, we have discussed the methods in a purely theoretical approach. In the next Section we describe the relationship between the control laws so far presented and the congestion control algorithms deployed in the Internet.

2.4 Congestion control algorithms in the Internet

The transmission of data flows over the Internet is done mainly using the Transmission Control Protocol (TCP) [RFC 793, 1981]. This protocol provides a reliable transport over an unreliable datagram network. The protocol takes the input flow of data from the application layer and segments it into *packets*, which will be delivered to the network layer below, in charge of relaying these packets to the destination. In order to recover from possible losses in the communication, TCP adds a sequence number to each packet and the destination sends an acknowledgment (ACK) packet in the reverse path to tell the source the data has correctly arrived. When TCP detects a missing acknowledgment, it will interpret it as a lost packet and retransmit the data. When this happens, the receiver must buffer the packets until it fills all the gaps, so it can deliver an ordered stream to the application layer above at the destination.

In order to control the number of *in-flight* packets (i.e. packets sent but whose ACK has not yet been received), the sender side maintains a *transmission window* W, which is the maximum allowed number of in-flight packets. The simplest case is $W = 1$ which is the *stop-and-wait* algorithm. In that case, the sender waits for the ACK of the previous packet before proceeding with the next. This choice of W is clearly inefficient when the Round Trip Time (RTT) between source and destination is large, since it can only send one packet per RTT.

A more efficient situation can be achieved by enlarging the number of in flight packets as depicted in Figure 2.4. In that case, the average rate obtained by the source will be given by:

$$x = \frac{W}{RTT}.$$

Suppose now that the network is able to provide a maximum amount of bandwidth c (in packets per second). In that case, an optimal choice of W will be equal to the bandwidth delay

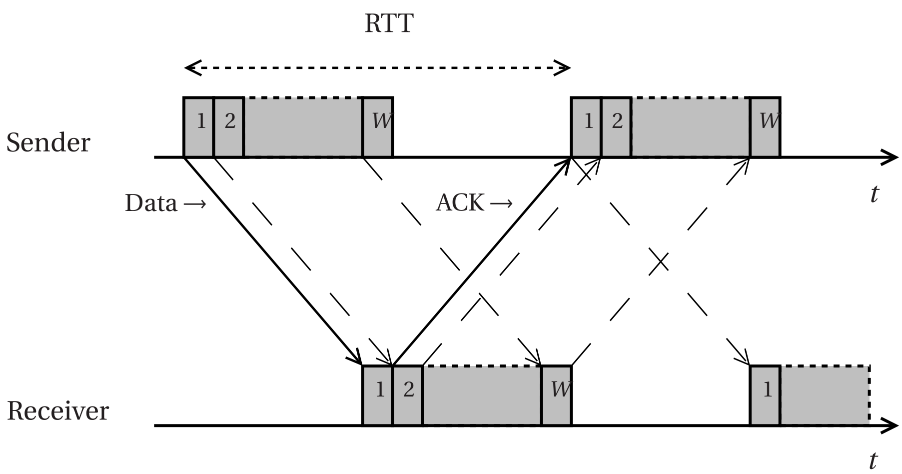

FIGURE 2.4: SLIDING WINDOW PROTOCOL AND RATE CALCULATION.

product $c \times d$, with d the round-trip propagation delay of the network. Setting W greater than this value will only increase the total RTT due to buffered packets that cannot be immediately delivered, and the average rate will still be equal to c. Setting W lower than this value will not make use of the full capacity of the network.

The problem is that, even if the source can measure its RTT, it does not know c a priori, and most importantly, the value of this allocated capacity may be time varying due to other connections competing for the same resources. In [Jacobson, 1988], the author described a simple algorithm to adapt the value of W. We now describe this algorithm and show how it can be interpreted as a primal adaptation in the language of Network Utility Maximization.

2.4.1 Loss based congestion control: TCP Reno and its variants

The congestion control mechanism designed by Jacobson, proceeds as follows. At the start of the connection, the window is set to $W = 1$ and the *slow start phase* begins.

In this phase, with each ACK received, the value of W is increased by 1. This has the effect of effectively duplicating the transmission window each RTT, since for each W packets sent in a cycle, W ACKs will be received (assuming no packet is lost) and in the next cycle, the window will be $2W$. This procedure continues until a certain threshold named *ssthresh* (for slow start threshold) is attained. A typical initial value of *ssthresh* is 32 Kbytes, which in typical networks translate to $W \approx 22$ packets. So, the *ssthresh* is attained early in the data transfer, provided there are no losses, with the slow start phase lasting typically 5 $RTTs$. At this point the *congestion avoidance phase* begins.

In this new phase, with each ACK received the window is increased by $1/W$. This has the net effect of increasing almost linearly the congestion window by 1 packet on each cycle, which is a slower growth than the slow start phase. For a moderately large connection, most of the data transfer will be achieved in this phase.

The increase in W continues until the sender detects a loss. In the original proposal, losses were detected when a certain amount of time (based on the measured RTT) has passed without receiving the ACK of a certain packet, an event called a *timeout*. At this point, the sender:

- Returns the transmission window to $W = 1$.
- Sets the *ssthresh* to half the value of the window prior to the loss.
- Restarts transmission from the lost packet onwards performing again the slow start phase.

Soon after its introduction, it was clear that the timeout mechanism to detect losses led to a drastic reduction in the rate, due to the combination of the idle times and the return of W

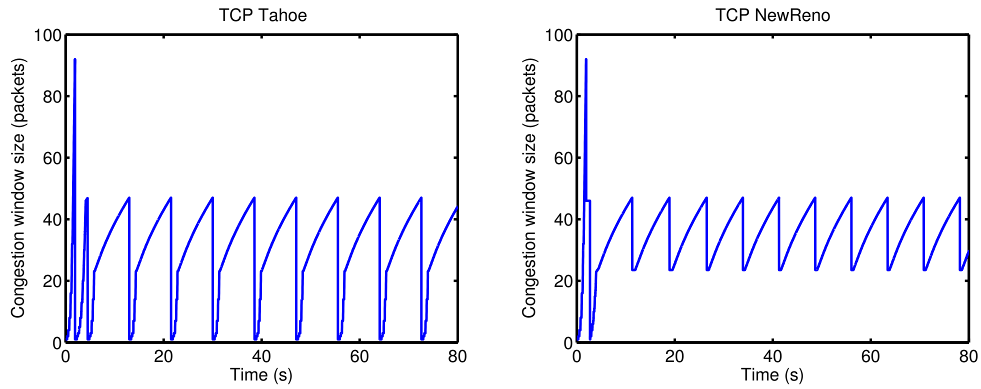

FIGURE 2.5: TYPICAL WINDOW EVOLUTION FOR TCP-TAHOE (LEFT) AND RENO (RIGHT)

to 1. To solve this issue, the *Fast Retransmit* mechanism was introduced. The main difference with the original proposal is the detection of losses via duplicate ACKs. If a single packet is lost, the stream of packets continue to arrive to the destination, but there is a gap in the sequence number. This produces a number of ACKs indicating that a packet is missing and requesting it. Once three of these "duplicate ACKs" are received, the sender assumes that a loss has occurred. It retransmits the requested packet and resumes from the slow start phase.

The algorithm so far described became known as TCP Tahoe, since it was first implemented in the Tahoe version of the BSD operating system. A typical evolution of W under the TCP-Tahoe algorithm is shown on the left in Figure 2.5.

The variants of the algorithm introduced later, known as TCP-Reno and TCP-NewReno (see [Floyd and Henderson, 1999] and references therein) tried to further improve the average rate by eliminating the slow start phase in steady state. These congestion control algorithms proceed as follows:

- Retransmit immediately the requested packet.
- Reduce the *ssthresh* to half the value of the window prior to the loss.
- Set the window to *ssthresh* and resume from the congestion avoidance phase (they skip the slow start altogether).

The result is a more regular behavior of the congestion window, as depicted on the right in Figure 2.5. Timeouts are still observed, and in this case the algorithm defaults to TCP-Tahoe behavior, but in the typical scenarios, the connection remains most of the time in the congestion avoidance phase.

A first analysis on the average rate obtained by the TCP-Reno mechanism was presented in [Mathis et al., 1997]. The authors establish the following relationship between the average window $\bar{W}$ and the packet loss probability q:

$$\bar{W} = \sqrt{\frac{3}{2q}}.$$

Taking into account the relationship between W and the average rate we have the following equation, called *Mathis formula*:

$$\bar{x} = \frac{1}{RTT}\sqrt{\frac{3}{2q}}, \tag{2.14}$$

where x is given packets per second. Note that equation (2.14) is in the form of a demand curve of equation (2.5a), where the price is the packet loss probability and the connection "purchases" less rate whenever this price is high.

A more detailed analysis of the TCP-Reno was given in [Kelly et al., 1998]. Assume that the network drops packets from connection r with probability q_r. Moreover, let RTT_r be the round trip time of connection r, and assume that it is fixed (this is the case when propagation delays are dominant). Then the rate at which the network drops packets from connection r is $x_r q_r$. If we assume that the window W_r increases by $1/W_r$ when a packet is correctly received, and decreases by a factor $0 < \beta < 1$ when a packet is lost, a suitable fluid model for the window evolution will be:

$$\dot{W}_r = \frac{1}{W_r} x_r (1 - q_r) - \beta W_r x_r q_r,$$

which can be rewritten in terms of the rate x_r as:

$$\dot{x}_r = \frac{1}{RTT_r^2}(1 - q_r) - \beta x_r^2 q_r. \tag{2.15}$$

As for the loss probabilities, a suitable model for networks with small buffers is to consider that each link drops packets proportional to its excess input rate, namely:

$$p_l = \left(\frac{y_l - c_l}{y_l}\right)^+,$$

where as usual $(x)^+ = \max\{x, 0\}$. Moreover, if link loss probabilities are small (which is the typical case in high bandwidth scenarios), and assuming that each link behaves independently, we can approximate to first order the route loss probability as:

$$q_r = \sum_{l:l \in r} p_l.$$

If route loss probabilities are small, we can also simplify equation (2.15) by making the approximation $(1-q_r)\approx 1$ in the first term. The complete dynamics follow:

$$\dot{x}_r = \beta x_r^2 \left(\frac{1}{\beta RTT_r^2 x_r^2} - q_r \right) \quad \forall r, \tag{2.16a}$$

$$y = Rx, \tag{2.16b}$$

$$p_l = \left(\frac{y_l - c_l}{y_l} \right)^+ \quad \forall l, \tag{2.16c}$$

$$q = R^T p. \tag{2.16d}$$

The previous dynamics are in the form of a primal controller of Section 2.3.1 with penalty function $f_l(y) = (1 - c_l/y)^+$. Therefore, TCP-Reno can be seen as a primal algorithm that decentralizes the resource allocation problem. The corresponding utility function is:

$$U_r(x_r) = -\frac{1}{\beta RTT_r^2 x_r},$$

which belongs to the $\alpha-$family with $\alpha = 2$ and $w_r = 1/(\beta RTT_r^2)$. Note also that the equilibrium of (2.16) satisfies:

$$\frac{1}{\beta RTT_r^2 \hat{x}_r^2} - \hat{q}_r = 0,$$

or equivalently:

$$\hat{x}_r = \frac{1}{RTT_r}\sqrt{\frac{1}{\beta \hat{q}_r}},$$

which is similar to equation (2.14) from the static analysis.

With minor changes, the TCP-Reno algorithm evolved into TCP-NewReno, which fixes some issues in the Fast Recovery/Fast Retransmit algorithm and provides a more stable behavior. This has been the mainstream protocol for data transfer in the Internet, and its success has allowed the Internet to scale. Only very recently new proposals like TCP-Cubic or TCP-Redmond are in consideration, with neither one being a clear replacement for the previous algorithms. The main breakthrough of [Kelly et al., 1998] was to relate the behavior of these congestion control algorithms to economic models.

2.4.2 Delay based congestion control

Besides the loss based protocols studied in the previous section, some TCP variants proposed the use of *queueing delay* as a signal of congestion. In such a protocol, the source measures the minimum observed RTT. This measure is is an estimation of the round trip latency in the communication, i.e. when all the buffers are empty. In this context, this is called the *base RTT* in this context), and the source reacts to an increase of the RTT above this estimated value. Variants such as TCP-Vegas [Brakmo and Peterson, 1995] and TCP-Fast [Cheng et al., 2004] are based on this idea. We now show that using queueing delay as the price can be assimilated to performing the dual algorithm described in Section 2.3.2.

To see this, recall the dynamics of the price variable in the dual algorithm (2.11):

$$\dot{p}_l = \gamma_l (y_l - c_l)^+_{p_l}.$$

Note that the term $y_l - c_l$ tracks the excess input rate in the link. Integrating $\dot{p}_l$ we have that p_l is proportional to the amount of data accumulated in the buffer of link l. By choosing $\gamma_l = 1/c_l$ we have that p_l represents the instantaneous queueing delay in the link.

Since the route price is $q_r = \sum_{l:l\in r} p_l$, this price tracks the total queueing delay suffered by route r packets along its path. Therefore, if the connection is able to measure this queueing delay and react to this price with some demand curve $f_r(q_r)$, we have a dual algorithm. If the route instead of instantaneously adapting to the current queueing delay, chooses to adapt its rate in a primal fashion, we have a primal-dual algorithm.

In the case of TCP-Vegas and TCP-Fast, the demand curve is chosen in a way compatible with the utility function $U_r(x_r) = w_r \log(x_r)$, therefore providing weighted proportional fairness among the ongoing connections.

2.4.3 Random Early Detection

As we discussed above, typical TCP congestion control algorithms react to lost packets. However, this leads to an undesirable behavior in the network buffers, which are working almost full most of the time. This leads to increased delays, which can affect for instance non TCP traffic used in real time network applications.

To solve this issue, Floyd et. al. proposed in [Floyd and Jacobson, 1993] to use a *Random Early Detection* (RED) algorithm. The main idea is to discard packets randomly with increasing probability as the buffer starts to fill. This leads to an early detection of congestion by the sources, which cooperate lowering their rates. When appropriately designed, this also helps to reduce the synchronization between packet losses of different sources, enabling a better use of network resources.

A simple model of the RED algorithm is the following: consider that the packet loss probability is proportional to the buffer occupancy, namely:

$$p_l = \kappa_l b_l.$$

From the discussion in the previous section, we know that the buffer occupancy obeys the following equation:

$$\dot{b}_l = (y_l - c_l)^+_{b_l}.$$

Putting together both equations we have:

$$\dot{p}_l = \kappa_l (y_l - c_l)^+_{p_l},$$

so the packet loss probability now follows the dual price adaptation, with $\gamma_l = \kappa_l$. If we assume that the loss probabilities are low, we have that the route loss probability is again $q_r = \sum_{l:l\in r} p_l$. Combining all these equations with the model for TCP-Reno (2.15) we have that TCP-Reno with RED follows a primal-dual dynamics. In the case of TCP-Reno, this will solve exactly Problem 2.2 with utilities of the $\alpha-$family with $\alpha = 2$.

2.5 Network utility maximization beyond congestion control

So far, we have describe how the NUM framework introduced in [Kelly et al., 1998] provided the mathematical tools to analyze the behavior of typical congestion control algorithms. This success led to a paradigm shift in network protocol design, based on convex optimization methods.

In particular, the NUM framework has been applied extensively in the literature to provide solutions to several resource allocation problems, beyond traditional congestion control. For instance, several generalizations of the above methods were developed to tackle the multi-path resource allocation problem. There, connections may be split up along several paths across the network. Two main formulations have been given: one in which the paths are pre-defined (c.f. [Han et al., 2006, Voice, 2007]) and the transport layer is in charge of managing the traffic on each route. Another formulation decomposes the multi-path problem in a congestion control layer, similar to the one discussed above, and a routing problem that can be solved by the intermediate nodes in a decentralized way, by using simultaneously all of the resources in the network [Paganini and Mallada, 2009]. This is achieved by applying a dual decomposition of the NUM problem, leading to separate optimization problems to be solved in the different network layers, and communicating through suitably generated network prices.

Another problem where the NUM framework proved useful is in the wireless setting, where several links may interfere with each other. Here, besides the typical resource allocation to each entity in the links, the additional *scheduling* problem arises, i.e. which links may be active at a given time, in order to communicate efficiently and without interference. By means of the same dual decomposition techniques, a suitable solution is achieved, where the problem decomposes in the congestion control and scheduling problems, though the last can be difficult to decentralize. A good reference here is [Lin et al., 2006].

These examples motivate the study of resource allocation in networks following the NUM framework. The general technique can be summarized as follows: state the NUM problem, that is, to maximize a certain utility notion which involves all the entities in the network, and let the properties of the network act as constraints in the given optimization problem. If these constraints are convex, apply a dual decomposition analysis to translate the above problem to suitable subproblems that must be solved by the different layers of the network, possibly communicating between them through well defined prices. If this decomposition is possible, several algorithms analogous to the primal, dual, and primal-dual algorithms presented in this chapter can be applied in order to drive the system to the optimal allocation. A good reference for this cross-layer analysis of networks is [Chiang et al., 2007]. In the following, we shall use this procedure to tackle several problems arising in network resource allocation.

Part I

Connection level resource allocation and user-centric fairness

3

Connection level models and user incentives

The models and discussion of Chapter 2 analyze the resource allocation in networks treating each flow or connection as an individual entity, which is permanently present in the network. However, in real networks there are two important phenomena that must be considered. In the first place, network users may open multiple simultaneous connections, maybe with different destinations and along multiple routes. On the other hand, the number of established connections or *flows* may be time varying, subject to the user demands and the fact that they may carry different amounts of work.

In this chapter, we review the models available to describe such situations. In Section 3.1 we review the relevant NUM framework for the connection level perspective as presented by [Srikant, 2004]. A first contribution of this thesis is presented in Section 3.2, where it is shown that under suitable assumptions a user has incentives to open more connections, in order to get more bandwidth. This is an intuitive result, it has been used before [Briscoe, 2006] as an argument against the flow-level notion of fairness enforced by current protocols, and we provide the first rigorous proof. Finally, in Section 3.3 we present the standard model for dealing with the time varying aspect of connections, and discuss its implications.

3.1 Introduction

The models and algorithms described in Chapter 2 assume that each connection sharing resources in the network has an associated increasing and concave utility function that values its desire for resources.

This economic interpretation, summarized in Problem 2.2 and the underlying algorithms is, however, only a metaphorical description. TCP-Reno and its variants, for instance, have been designed in a heuristic manner, and only later these algorithms were interpreted in terms of economic ideas.

In particular, the utility function that describes TCP-Reno is determined by the protocol, as we saw in Chapter 2. The end-user has little control over this, since it is implemented in the kernel of the operating system, and thus changing it may be difficult. As a consequence, the resource allocation provided by the currently predominating Internet protocols does not really reflect the user valuation of bandwidth.

However, as we shall see below, users can cheat on the resource allocation in a simpler way: they can open more connections, and this is indeed a mechanism that has been used in the Internet for a long time. The emergence of peer-to-peer file exchange software, which opens several parallel connections, has deepened its implications nowadays.

In order to model this situation, we shall focus on the resource allocation provided by the network in the presence of multiple connections. The model below was introduced by [de Veciana et al., 1999, Bonald and Massoulié, 2001] and is summarized in [Srikant, 2004].

We consider, as in Chapter 2 a network composed of links, indexed by l, with capacity c_l, and a set of paths or routes, indexed by r. End-to-end connections (flows) travel through a single path, specified by the routing matrix R. x_r denotes the rate of a single connection on route r, and n_r will denote the number of such connections. We define:

$$\varphi_r = n_r x_r,$$

which is the aggregate rate on route r, obtained by the whole set of ongoing connections on this route. The aggregate rate in link l from all the routes is:

$$y_l = \sum_r R_{lr}\varphi_r = \sum_r R_{lr} n_r x_r. \tag{3.1}$$

Connections present in the network regulate their rate through some TCP congestion control algorithm. Assume each connection on route r can be modelled by some utility function U_{TCPr}, and connections along the same path have the same utility. Then, the resource allocation of the Network Problem 2.2 reduces to:

Problem 3.1 (TCP Congestion control)**.** *For fixed* $\{n_r\}_{r\in\mathscr{R}}$, $n_r > 0$,

$$\max_{\varphi_r} \sum_r n_r U_{TCPr}\left(\frac{\varphi_r}{n_r}\right),$$

subject to capacity constraints:

$$\sum_l R_{lr}\varphi_r \leqslant c_l \quad \textit{for each } l.$$

The above optimization problem gives a notion of "flow-rate fairness", where the utility U_{TCPr} is assigned to the rate $x_r = \frac{\varphi_r}{n_r}$ of each TCP flow, modelling protocol behavior. For future convenience, we have chosen to express the problem in terms of the φ_r variables. TCP utilities are assumed increasing and strictly concave, and we will mainly focus on the $\alpha-$fair family of [Mo and Walrand, 2000].

The Karush-Kuhn-Tucker (KKT) conditions for Problem 3.1 include

$$U'_{TCPr}\left(\frac{\varphi_r}{n_r}\right) = q_r \quad \forall r,$$

which are equivalent to the *demand curve*:

$$x_r = \frac{\varphi_r}{n_r} = f_{TCPr}(q_r), \tag{3.2}$$

with $f_{TCPr} = [U'_{TCPr}]^{-1}$. In particular, for $\alpha-$fair utilities we have $f_{TCPr}(q_r) = (q_r/w_r)^{-1/\alpha}$.

Therefore, TCP congestion control algorithms behave as decentralized ways of achieving the optimum of Problem 3.1, where the utility function U_{TCPr} models the *protocol behavior*. This defines a mapping $\Phi : n \mapsto \varphi$; that is, given the number of connections $n = (n_r)$ in each route, the resource allocation $\varphi = (\varphi_r)$ is given as the solution of Problem 3.1. In the next Section we derive an important property of this mapping.

3.2 User incentives

From the user-level perspective, for a given number of connections in each route, the resource allocation is determined by the flow-rate fairness imposed by the network through the solution of Problem 3.1. However, a user vying for more resources may challenge this by opening more connections along the same route. Our first result states that this indeed allows the user to get a larger share of bandwidth.

Theorem 3.1. *Assume R has full row rank. Then the map $\varphi = \Phi(n)$ is such that:*

$$\frac{\partial \varphi_r}{\partial n_r} \geqslant 0 \quad \forall r, n_r > 0.$$

Proof. Consider the map $\Phi : n \mapsto \varphi$ defined by Problem 3.1. This map is continuous when $n > 0$ [Kelly and Williams, 2004]. We will also assume that in a neighborhood of the solution of Problem 3.1, all links are saturated (if there are locally non-saturated links, they can be easily removed from the analysis).

In this case, the KKT conditions of Problem 3.1 imply:

$$U'_{TCPr}\left(\frac{\varphi_r}{n_r}\right) = q_r, \quad \forall r,$$
$$\sum_r R_{lr}\varphi_r = c_l \quad \forall l.$$

From the first group of equations we have that:

$$\varphi_r = n_r f_{TCPr}(q_r) \quad \forall r, \tag{3.3}$$

and substituting in the link constraints we have that the optimal link prices must satisfy:

$$F(n,p) = R\text{diag}(n) f_{TCP}(R^T p) - c = 0.$$

Here, $\text{diag}(n)$ denotes a diagonal matrix with the entries of n, $c = (c_l)$ is the vector of link rates and $f_{TCP}(R^T p)$ is the vector of flow rates determined by the demand curve in each route.

By using the Implicit Function Theorem we have that:

$$\frac{\partial p}{\partial n} = -\left(\frac{\partial F}{\partial p}\right)^{-1}\left(\frac{\partial F}{\partial n}\right).$$

Define now the following matrices:

$$N = \text{diag}(n),$$
$$F = \text{diag}(f_{TCP}(R^T p)),$$
$$F' = \text{diag}(f'_{TCP}(R^T p)).$$

Note that the diagonal entries of N and F are strictly positive, and the diagonal entries of F' are negative, since we assume the links are saturated and therefore the prices involved are positive. After some calculations we arrive to:

$$\frac{\partial F}{\partial p} = RNF'R^T,$$
$$\frac{\partial F}{\partial n} = RF,$$

and thus:

$$\frac{\partial p}{\partial n} = -(RNF'R^T)^{-1}(RF).$$

Note that the first matrix is invertible since R has full row rank and the diagonal matrix has definite sign.

We now turn to calculating $\frac{\partial \varphi}{\partial n}$. From equation (3.3) we have:

$$\frac{\partial \varphi}{\partial n} = F + NF'R^T \frac{\partial p}{\partial n},$$

and therefore:

$$\frac{\partial \varphi}{\partial n} = \left(I - NF'R^T(RNF'R^T)^{-1}R\right) F,$$

where I is the identity matrix.

We would like to prove that the diagonal terms of this matrix are non-negative. Since F is a diagonal matrix with positive entries, we can reduce the problem to proving that the following matrix has positive diagonal entries:

$$M = I - DR^T(RDR^T)^{-1}R,$$

where $D = -NF'$ is also a diagonal matrix with strictly positive entries. We can thus define $\sqrt{D} = \text{diag}(\sqrt{d_{ii}})$ and write:

$$\sqrt{D}^{-1} M \sqrt{D} = I - \sqrt{D}R^T(RDR^T)^{-1}R\sqrt{D},$$

which is of the form $I - A^T(AA^T)^{-1}A$. Note that this last matrix is symmetric and verifies $(I - A^T(AA^T)^{-1}A)^2 = I - A^T(AA^T)^{-1}A$. Thus, it is a self-adjoint projection matrix, and therefore is positive semidefinite. From this we conclude that the diagonal entries of $\sqrt{D}^{-1} M \sqrt{D}$ are non-negative, and since the diagonal entries of M are not altered by this transformation, M has non negative diagonal entries, which concludes the proof. □

The above result states that for a general network, $\partial \varphi_R / \partial n_r$ is non-negative, and in fact examples can be constructed where equality holds. However, in network scenarios where there is real competition for the resources, typically the inequality above is strict.

Theorem 3.1 therefore implies that a greedy user has incentives to increase the number of connections along a route in order to garner more bandwidth. To formalize this further, consider that user i has an associated increasing and concave utility function $U_i(\varphi^i)$ where φ^i is the total rate it gets from the network. We can define a game between users in which the strategy space is the number of connections each user opens, and the payoff function is $U_i(\varphi^i(n))$. As a consequence of Theorem 3.1 we have the following corollary:

Corollary 3.1. *The dominant strategy for the connection-level game is to increase the number of connections.*

This formalizes arguments of [Briscoe, 2006] regarding the limitations of flow-rate fairness.

We conclude that, if a subset of users behave in this way, the number of ongoing connections in each route will grow without bounds, which is an undesirable situation since the overheads involved would lead to an important waste of resources. This kind of situations are known as the *tragedy of the commons* [Hardin, 1968], in which each individual acting selfishly has incentives to enlarge its use of the resources, but when the group behaves in this way the common good is depleted, or at least inefficiently used. We shall address this issue further in Chapter 4 where we develop algorithms and admission control mechanisms to control this selfish behavior and drive the system to a suitable social optimum.

3.3 Stochastic demands

Another way of taking users into account is through a stochastic model for demand. Here, users are not strategic, instead they open connections on route r at times governed by a Poisson process of intensity λ_r. Each connection brings an exponentially distributed workload with mean $1/\mu_r$. These kind of models were introduced by [de Veciana et al., 1999] and [Bonald and Massoulié, 2001].

For each network state $n = (n_r)$, the rate at which each connection is served is assigned according to Problem 3.1. A time scale separation assumption is often used to simplify the analysis, namely, we assume that congestion control protocols drive the system to the optimum of Problem 3.1 for a given n_r instantaneously; i.e. the congestion control operates at a faster (packet-level) timescale than the connection arrival/departure process.

We can define the average load on route r as $\rho_r = \lambda_r/\mu_r$. Using the time-scale separation assumption between congestion control and connection-level dynamics, the process $n(t) = (n_r(t))$ becomes a continuous time Markov chain (c.f. Appendix A.4), with the following transition rates:

$$n \mapsto n + e_r \quad \text{with rate} \quad \lambda_r, \tag{3.4a}$$

$$n \mapsto n - e_r \quad \text{with rate} \quad \mu_r \varphi_r(n). \tag{3.4b}$$

Here e_r denotes the vector with a one in coordinate r and 0 elsewhere. $\varphi_r(n)$ is the rate assigned to route r by the map Φ defined earlier. The process defined by equations (3.4) is a multidimensional birth and death process with coupled death rates. This process is not reversible and the global balance equations cannot be solved explicitly except in some special cases. However, its stability (in the sense of positive recurrence) can be analyzed. The main result is:

Theorem 3.2 ([de Veciana et al., 1999, Bonald and Massoulié, 2001]). *For the weighted $\alpha-$fair resource allocation with $\alpha > 0$, the Markov model defined by* (3.4) *is stable (i.e. positive recurrent) if:*

$$\sum_r R_{rl} \rho_r < c_l \quad \forall l. \tag{3.5}$$

Moreover, if $\sum_r R_{rl}\rho_r > c_l$*, for some* l*, the process is transient.*

Note that this theorem states that network stability is guaranteed whenever every link load $\rho^l = \sum_r R_{rl}\rho_r$ does not exceeds link capacity. This result may seem trivial at first sight, since condition (3.5) is the natural condition arising in queueing theory, whenever each link is considered a separate queue.

Note however that in fact, each *route* behaves as a connection level queue, served through a complex policy defined by the resource allocation $\varphi(n)$. The fact that the stability region of the system coincides with the interior of the capacity region, i.e. the interior of the set defined by the capacity constraints, is due to the fact that the $\alpha-$fair policy for $\alpha > 0$ can be seen as a suitable generalization of the processor sharing service discipline to this coupled system. We can also construct simple examples for $\alpha = 0$ where Theorem 3.2 does not hold.

Theorem 3.2 also assumes that flow sizes follow a exponential distribution. This is a restrictive hypothesis, since it is known that typical Internet flows do not follow this distribution. Nevertheless, the result has also been extended in different ways in [Lin and Shroff, 2006], [Massoulié, 2007] and [Paganini et al., 2009], showing that the stability condition is the same under more general hypotheses for the arrival and workload processes, however the theory becomes more involved in that case.

The stochastic stability region is therefore characterized. However, fairness between users is not addressed by this kind of result: the connection level dynamics are such that either the loads can be stabilized and users are satisfied, or the network is unstable. In the latter case, the number of connections grow without bounds, up to a point where user maybe impatience comes into play and the connections are dropped. This undesirable behavior is *independent* of the flow rate fairness imposed by TCP (i.e. the value of α).

Some authors [Massoulié and Roberts, 1999] argue that the above situation requires admission control of connections. While admission control may overcome instability, we believe that it should be carried out in a way that fairness *between users* is taken into account. To tackle this issue, in the following Chapter we shall define a suitable notion of *user-centric fairness*, where a user may represent a bundle of connections and routes over the network. Applying this notion, we will construct different decentralized mechanisms to drive the network to a fair operating point.

4

User-centric fairness

The models presented in Chapter 2 make use of an economic interpretation of the resource allocation problem to formulate a suitable NUM problem. On the other hand, the discussions of Chapter 3 show that this economic language is metaphorical in the sense that user valuation of bandwidth is not taken into account, the underlying packet-level dynamics being governed by some underlying protocol and the flow-rate fairness it imposes.

However, the economic language introduced suggests the possibility of more closely relating network resource allocation with the real incentives of Internet users. A significant gap in the literature has remained between these two points of view. Among the many reasons for this disconnection, we focus here on one: users do not care about TCP connections, but about the aggregate rate of service they receive from the network. Applications can, and often do, open multiple connections to vie for a larger share of the bandwidth "pie", and in Theorem 3.1 we showed that this is indeed a rational strategy.

What we want is to define a suitable notion of *user-centric fairness*, where resource allocation takes place in terms of the *aggregate rate* of connections serving a common entity, which can be identified as a certain "user". These aggregations can be quite general and may involve single-path or multi-path settings: for instance, emerging peer-to-peer (p2p) systems dynamically manage multiple connections with multiple sites, all serving the common goal of a single download. Fairness in this context applies naturally to the overall download rate of flows going through different routes.

In this Chapter we proceed to define the notion of user centric fairness by a suitable NUM problem, which can be solved via the standard techniques of Lagrangian primal-dual decomposition presented in Chapter 2. In Section 4.1 we define this notion in the general case. Then, in Section 4.2 we derive suitable decentralized control algorithms for the aggregate rate that drive the system to the desired resource allocation. Since the aggregate rate is often difficult to control directly by the user, in Section 4.3 we proceed to analyze algorithms based on controlling the number of connections in the single path case. In Section 4.4 we generalize these algorithms to the multi-path setting. All the algorithms of this Chapter are derived assuming some form of *cooperation* between users. We postpone the discussion on how the network may protect itself of non-cooperative users to Chapter 5. Finally, in Section 4.5 we discuss a packet level implementation of the proposed algorithms and show an example of the mechanism in action.

4.1 Definition

Let us assume, as before, that the network is composed by a set of resources or links of capacity c_l. Routes, indexed by r are established along these links, and we have, as before, a routing matrix R_{lr} which stores the incidence of routes on links.

Above this network, we have new entities which we call *users*, and we index them by i. Users open one or more end to end connections through the network, over a set of routes. If a given user opens connections over route r we write $r \in i$. Note that the same route can be in use by more than one user. In that case we duplicate the index r, indexing routes of different users with different labels.

These connections obtain from the network an aggregate rate φ_r on route r, and we define the *total rate* a given user obtains from the network through all of its routes as:

$$\varphi^i = \sum_{r \in i} \varphi_r.$$

Let U_i be an increasing and concave utility function that models *user preferences* or desire for bandwidth, independent of the protocols in place in the network. We propose the following notion of *user-centric fairness*, defined via the following NUM problem:

Problem 4.1 (User Welfare)**.**

$$\max_{\varphi_r} \sum_i U_i(\varphi^i),$$

subject to:

$$\sum_r R_{lr} \varphi_r \leqslant c_l \quad \forall l.$$

The above Problem amounts to finding a suitable resource allocation for every route in the network so that the *total user welfare* is maximized. By using different utility functions (e.g. the $\alpha-$family of Definition 2.4) we can model different notions of fairness between users.

Note also that the framework is very general: a user is defined by a set of routes and a utility function. This can model for instance users downloading data from several locations, multiple parallel paths, users uploading data to several points in the network, and of course, the single path situations discussed earlier.

As compared to the original resource allocation Problem 2.2 from [Kelly et al., 1998], the above definition takes into account not only the behavior of individual connections along a route, controlled by the transport protocol TCP, but also on the number of active connections associated with each user.

Our main goal in this part of the thesis is to find suitable decentralized algorithms that, taking the underlying TCP congestion control as given, drive the system to the resource allocation of Problem 4.1.

A first step in our study is to assume that users cooperate in order to maximize its social welfare. Even in this case, Problem 4.1 is more difficult that Problem 2.2 since the new objective function is not necessarily strictly concave, even when the utilities satisfy Assumption 2.1. To see this note that the objective function can be rewritten as:

$$\sum_i U_i(\varphi^i) = \sum_i U_i\left(\sum_{r\in i} \varphi_r\right).$$

Therefore, switching some resources from a route $r \in i$ to another route $r' \in i$, while maintaining the value of φ^i does not change the value of the objective function. Thus any convex combination $\theta\varphi_r + (1-\theta)\varphi_{r'}$ produces the same total welfare. Of course, in the single-path case where each user is confined to a single route, Problem 4.1 and Problem 2.2 coincide.

In the following Section we will show how to address this issue, assuming that users can directly control the rate φ_r for each $r \in i$. Then we will proceed to find suitable laws for n_r, the number of connections in each route, that emulate the desired behavior for φ_r.

4.2 Controlling the aggregate rates

In this Section, we assume that users have direct control of the total rates per route φ_r, $r \in i$, and we would like to propose a decentralized dynamic control law for the variables φ_r such that the network is driven to the solution of Problem 4.1.

We will resort again to the Lagrangian duality framework. We denote $\varphi = (\varphi_r)$ the rate vector, $p = (p_l)$ the price vector and, as before, $y = (y_l)$ is the vector of link rates and $q = (q_r)$ the vector of route prices.

The Lagrangian of Problem 4.1 is:

$$\begin{aligned}\mathscr{L}(\varphi,p) &= \sum_i U_i(\varphi^i) - \sum_l p_l(y_l - c_l)\\ &= \sum_i U_i\left(\sum_{r\in i}\varphi_r\right) - \sum_r q_r\varphi_r + \sum_l p_l c_l.\end{aligned} \tag{4.1}$$

The KKT conditions that characterize the optimal route rates can be obtained by maximizing $\mathscr{L}$ over φ_r with fixed prices. The following conditions must be satisfied:

$$\text{Either}\quad U_i'\left(\sum_{r\in i}\varphi_r^*\right) = q_r^*, \tag{4.2a}$$

$$\text{or}\quad U_i'\left(\sum_{r\in i}\varphi_r^*\right) < q_r^* \text{ and } \varphi_r^* = 0. \tag{4.2b}$$

In particular, equations (4.2) imply that:

$$U_i'(\varphi^{i,*}) = q_i^* := \min_{r\in i} q_r^*. \tag{4.3}$$

Therefore, in the optimal allocation, user i only sends traffic through the paths with minimum price. The total rate of each user, φ^i, is determined in the optimum by condition (4.3). However, in general the optimal route rates φ_r need not be unique, since it may be possible to have several allocations φ_r with $\sum_{r\in i}\varphi_r = \varphi^i$ for each i, and producing the same prices in the network.

The User Welfare Problem 4.1 is similar to the congestion control problem described in Chapter 2, so a number of distributed approaches are available to drive φ_r to the desired allocation. Some difficulties appear due to the lack of strict concavity in the objective of Problem 4.1. For instance, in one of the very first works on the subject, [Arrow et al., 1958] showed that the primal-dual dynamics may lead to oscillatory behavior of the dynamics. We illustrate this by a simple example:

Example 4.1 (Primal-dual dynamics for the User Welfare problem). We consider a single user which has two routes available, as in Figure 4.1. For simplicity, we choose for the user the utility function $U(\varphi) = \log(\varphi)$, though the behavior will be the same with any utility. Link capacities are taken as $c_1 = c_2 = 1$ and since there is a route through each link, R is the 2×2 identity matrix.

In this setting, Problem 4.1 becomes:

$$\max_{\varphi} \log(\varphi_1 + \varphi_2),$$

subject to:

$$\varphi_1 \leqslant 1, \quad \varphi_2 \leqslant 1.$$

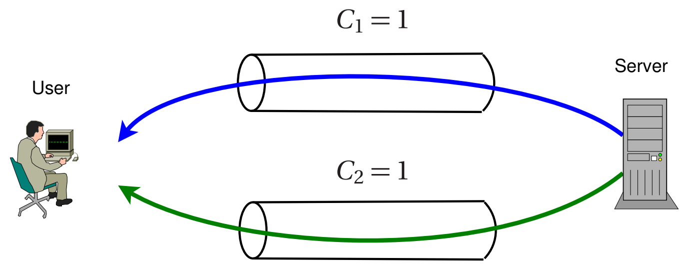

FIGURE 4.1: NETWORK OF TWO PARALLEL LINKS AND ONE USER.

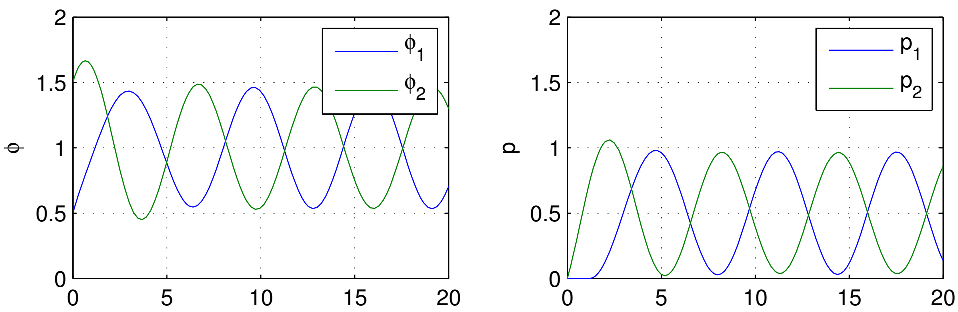

FIGURE 4.2: EVOLUTION OF THE PRIMAL DUAL DYNAMICS FOR EXAMPLE 4.1 WITH INITIAL CONDITION $\varphi_1 = 0.5$, $\varphi_2 = 1.5$ AND PRICES $p_1 = p_2 = 0$.

It is clear that the solution is $\varphi_1^* = \varphi_2^* = 1$ since U is increasing, and the equilibrium prices are $p_1^* = p_2^* = 1/2 = U'(\varphi_1^* + \varphi_2^*)$. Thus in this case the solution is unique.

The behavior of the system for the control law $\dot{\varphi}_r = U'(\varphi) - q_r$ was simulated and it is shown on Figure 4.2. We can see that even if the solution is unique, the system may oscillate around the equilibrium value.

To correct this problem, we will propose now a variant of primal-dual dynamics, with an additional damping term, that enables us to obtain global convergence to the equilibrium. Consider the following control law for φ_r:

$$\dot{\varphi}_r = k_r \left(U_i'\left(\sum_{r\in i} \varphi_r\right) - q_r - \nu \dot{q}_r \right)^+_{\varphi_r}, \tag{4.4a}$$

$$y = Rx, \tag{4.4b}$$

$$\dot{p}_l = \gamma_l (y_l - c_l)^+_{p_l}, \tag{4.4c}$$

$$q = R^T p. \tag{4.4d}$$

where $k_r > 0$, $\gamma_l > 0$ and $\nu > 0$. Recall that $(\cdot)^+_{\varphi_r}$ is the positive projection defined in equation

(2.12), and given by:

$$(x)_y^+ = \begin{cases} x & \text{if } y > 0 \text{ or } y = 0 \text{ and } x > 0 \\ 0 & \text{otherwise.} \end{cases}$$

Note that the damping term $\nu \dot{q}_r$ does not affect the equilibrium of the dynamics. We can see that, due to the projection $(\cdot)_{\varphi_r}^+$ the equilibrium of (4.4) satisfies the KKT conditions (4.2): in particular if $U_i'\left(\varphi^{i,*}\right) < q_r^*$ then φ_r^* must be zero.

The main idea behind the algorithm in (4.4) is that users, instead of reacting to the price q_r, they must react to the predicted route price $q_r + \nu \dot{q}_r$, thus anticipating possible changes. This idea first appeared in [Paganini and Mallada, 2009] in the context of combined multi-path congestion control and routing.

We have the following result concerning the stability of the algorithm:

Theorem 4.1. *Under the control law given in equations* (4.4) *all trajectories converge to a solution of Problem 4.1.*

Before proceeding with the proof, we shall need the following simple result on the positive projection, which is used as a standard argument in many congestion control theorems.

Lemma 4.1. *Given $u \geqslant 0$, $u_0 \geqslant 0$ and $v \in \mathbb{R}$, the following inequality holds:*

$$(u - u_0)(v)_u^+ \leqslant (u - u_0)v. \tag{4.5}$$

Proof. If $u > 0$ the left and right hand sides of (4.5) coincide. If $u = 0$, then the left hand side is $-u_0 \max\{v, 0\}$. If $v > 0$ this term is equal to $-u_0 v$, so both sides also coincide. If instead we have $v \leqslant 0$, then the left hand side it is $0 \leqslant -u_0 v$, so the inequality holds in all cases. □

Proof of Theorem 4.1. Consider the following Lyapunov function for the system:

$$V(\varphi, p) = \sum_r \frac{1}{2k_r}(\varphi_r - \varphi_r^*)^2 + \sum_l \frac{1}{2\gamma_l}(p_l - p_l^*)^2 + \sum_l \nu(c_l - y_l^*)p_l, \tag{4.6}$$

where (φ^*, p^*) is an equilibrium of the system, and $y^* = R\varphi^*$.

First note that V in the expression (4.6) satisfies $V \geqslant 0$ for every $\varphi \geqslant 0$ and $p \geqslant 0$. The first two terms are quadratic terms. The last term is non negative since $y_l^* \leqslant c_l$ due to the problem constraints, and $p_l \geqslant 0$. Note that this last term vanishes at any equilibrium due to equation (4.4c), which is in turn imposing the complementary slackness condition for the optimum.

When differentiating along trajectories we obtain:

$$\begin{aligned} \dot{V} = &\sum_r (\varphi_r - \varphi_r^*) \left[U_i'\left(\sum_{r \in i} \varphi_r\right) - q_r - \nu \dot{q}_r \right]_{\varphi_r}^+ + \sum_l (p_l - p_l^*)(y_l - c_l)_{p_l}^+ + \\ &+ \sum_l \nu(c_l - y_l^*)(y_l - c_l)_{p_l}^+. \end{aligned} \tag{4.7}$$

Noting that φ_r, φ_r^*, p_l and p_l^* are all non negative, we can apply Lemma 4.1 to get rid of the positive projection by inserting a $\leqslant$ sign in the first two terms. We have:

$$\dot{V} \leqslant \sum_r (\varphi_r - \varphi_r^*) \left[U_i' \left(\sum_{r \in i} \varphi_r \right) - q_r - \nu \dot{q}_r \right] + \sum_l (p_l - p_l^*)(y_l - c_l) + \\ + \sum_l \nu (c_l - y_l^*)(y_l - c_l)_{p_l}^+ . \tag{4.8}$$

By inserting the values at equilibrium appropriately we can decompose (4.8) in the following terms:

$$\begin{aligned}
\dot{V} \leqslant & \sum_r (\varphi_r - \varphi_r^*) \left(U_i' \left(\sum_{r \in i} \varphi_r \right) - q_r^* \right) && (I) \\
& + \sum_r (\varphi_r - \varphi_r^*)(q_r^* - q_r) && (II) \\
& - \sum_r \nu (\varphi_r - \varphi_r^*) \dot{q}_r && (III) \\
& + \sum_l (p_l - p_l^*)(y_l - y_l^*) && (IV) \\
& + \sum_l (p_l - p_l^*)(y_l^* - c_l) && (V) \\
& + \sum_l \nu (c_l - y_l^*)(y_l - c_l)_{p_l}^+ && (VI).
\end{aligned}$$

We start by noting that:

$$\begin{aligned}
(II) + (IV) &= -(\varphi - \varphi^*)^T (q - q^*) + (y - y^*)^T (p - p^*) \\
&= -(\varphi - \varphi^*)^T R^T (p - p^*) + (\varphi - \varphi^*)^T R^T (p - p^*) = 0.
\end{aligned}$$

The complementary slackness condition in turn implies that $(V) \leqslant 0$ since either $y_l^* = c_l$ and the term vanishes, or $y_l^* < c_l$ and $p_l^* = 0$, so each term in the sum is $p_l (y_l^* - c_l) \leqslant 0$.

The bound for the first term derives from the equilibrium condition (4.2). We associate the terms in (I) on each user i:

$$\sum_{r \in i} (\varphi_r - \varphi_r^*) \left(U_i' \left(\sum_{r \in i} \varphi_r \right) - q_r^* \right) = (\varphi^i - \varphi^{i,*}) \left(U_i' \left(\varphi^i \right) - U_i'(\varphi^{i,*}) \right) \\ + \sum_{r \in i} (\varphi_r - \varphi_r^*)(U_i'(\varphi^{i,*}) - q_r^*).$$

The first term of the right hand side is $\leqslant 0$ due to the fact that U_i' is a decreasing function. The second term is $\leqslant 0$ since either $U_i'(\varphi^{i,*}) = q_r^*$ if r is a route with minimum price, or $U_i'(\varphi^{i,*}) < q_r^*$ and $\varphi_r^* = 0$. Summing over i we conclude that $(I) \leqslant 0$.

The remaining terms can be grouped and verify:

$$\begin{aligned}(III)+(VI) &= -\nu(\varphi-\varphi^*)^T\dot{q}+\nu(c-y^*)^T\dot{p}\\ &= -\nu(\varphi-\varphi^*)^T R^T\dot{p}+\nu(c-y^*)^T\dot{p}\\ &= -\nu(y-y^*)^T\dot{p}+\nu(c-y^*)^T\dot{p}\\ &= \nu(c-y)\dot{p}\\ &= \nu\sum_l (c_l-y_l)(y_l-c_l)^+_{p_l}\leqslant 0,\end{aligned}$$

where the last inequality follows from observing that, due to the projection, each summand of the last term is 0 if $p_l=0$ and $y_l<c_l$ or $-(y_l-c_l)^2\leqslant 0$ otherwise.

We conclude that the function V is decreasing along the trajectories. Stability now follows from the LaSalle Invariance Principle (c.f. Appendix A.2). Assume that $\dot{V}\equiv 0$. In particular, the terms (I) and $(III)+(VI)$ must be identically 0 since they are negative semidefinite.

Imposing $(I)\equiv 0$ we conclude that $\varphi^i=\varphi^{i,*}$ for all i and $\varphi_r=0$ for all routes which do not have minimum price. Moreover, $\dot{\varphi}^i=\sum_{r\in i}\dot{\varphi}_r=0$ since φ^i must be in equilibrium.

We also have that:

$$(III)+(VI)=\nu\sum_l (c_l-y_l)(y_l-c_l)^+_{p_l},$$

and since each term has definite sign, imposing $(III)+(VI)\equiv 0$ requires that either $p_l=0$ or $y_l=c_l$ at all times. Therefore, $\dot{p}=0$ and p must be in equilibrium. It follows that q_r is in equilibrium, and therefore returning to the dynamics we must have $\dot{\varphi}_r=K_r$ a constant. If $\dot{\varphi}_r=K_r>0$, it would mean that $\varphi_r\to\infty$ implying that $y_l\leqslant c_l$ is violated at some link. Therefore, $K_r\leqslant 0$ and since $\sum_r\dot{\varphi}_r=0$, we must have $K_r=0$. Therefore φ_r is in equilibrium.

We conclude that in order to have $\dot{V}\equiv 0$ the system must be in a point that satisfies the KKT conditions (4.2), and therefore the system will converge to an optimal allocation for Problem 4.1. Moreover, since V is radially unbounded, this convergence will hold globally. □

Before proceeding with the analysis, let us compare the behavior of the dynamics (4.4) in the parallel links case given in Example 4.1.

Example 4.2 (Primal dual dynamics with damping). Consider again the setting of Example 4.1 depicted in Figure 4.1. Recall that the optimal allocation for Problem 4.1 is given by $\varphi_1^*=\varphi_2^*=1$, and the equilibrium prices are $p_1^*=p_2^*=1/2=U'(\varphi_1^*+\varphi_2^*)$.

The behavior of the system for the control law (4.4) was simulated and it is shown on Figure 4.3. We can see that in this case the system converges to the equilibrium allocation, and the prices find the right equilibrium price, thus correcting the oscillating behavior of Example 4.1.

As a result of Theorem 4.1, if users could control the aggregate input rate over each route following the dynamics (4.4), the system will converge to the optimal allocation of Problem

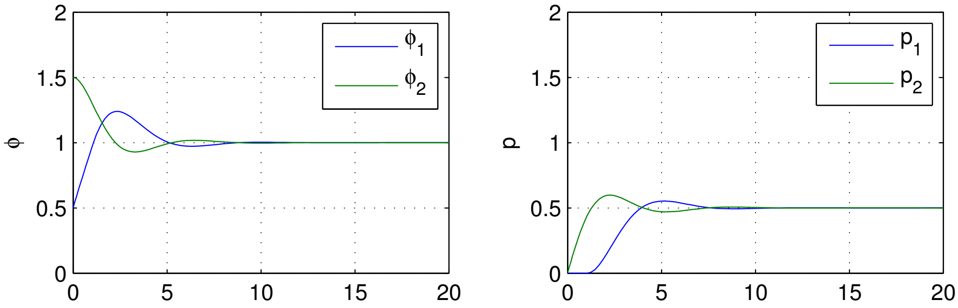

FIGURE 4.3: EVOLUTION OF THE PRIMAL DUAL DYNAMICS FOR EXAMPLE 4.1 WITH INITIAL CONDITION $\varphi_1 = 0.5$, $\varphi_2 = 1.5$ AND PRICES $p_1 = p_2 = 0$ AND DAMPING FACTOR $\nu = 1$.

4.1. Note that the proposed algorithm assumes that user i can control the rate on each of its routes, and the only knowledge it needs is the route price q_r and the predictive term $\dot{q}_r$. Therefore the algorithm is decentralized, since users only need information about the routes they are using to perform the control.

However, in real networks it is not reasonable to assume that the user has fine tuned control over the aggregate rate. As we discussed in Section 2.4, the transport layer gets in the way. Users may open one or several connections on each route, but the rate at which connection is served is determined by the congestion control algorithms in place, i.e. the TCP protocol. As we discussed in Section 3.1, the resource allocation performed by TCP can be modelled by Problem 3.1. In conclusion, the end user may control φ_r but only indirectly, through the number of ongoing connections on each route. The remaining sections in this chapter address this issue.

4.3 Controlling the number of flows: the single-path case

Based on the preceding discussions, we would like to find a suitable control law for the number of ongoing flows in each route such that the system converges to the resource allocation defined by Problem 4.1. In this section, we will focus on the simpler situation in which each user opens connections over only one route in the network. Therefore we can identify a user i and its route r, and the aggregate rate obtained by the user is $\varphi^i = \varphi_r$.

We will assume also that the network operates with some suitable congestion control algorithm such that, for a fixed number of connections $n = (n_r)$, the network seeks to optimize the TCP Congestion Control Problem 3.1. For ease of exposition we will assume that this congestion control is a dual algorithm, but we discuss the alternatives below.

Under these assumptions, the network part of the dynamics will be given by:

$$\begin{aligned}
\dot{p}_l &= \gamma_l (y_l - c_l)^+_{p_l} \quad \forall l, \\
q &= R^T p, \\
x_r &= f_{TCPr}(q_r) \quad \forall r, \\
\varphi_r &= n_r x_r \quad \forall r, \\
y &= R\varphi.
\end{aligned}$$

Here x_r represents the individual rate of each connection on route r. Given a current route price q_r, the rate is determined by the condition:

$$U'_{TCPr}(x_r) = q_r,$$

where $U_{TCP,r}$ models the TCP behavior. Therefore, $f_{TCP,r}$ is the demand curve for the utility function U_{TCPr}, or $U'^{-1}(q_r)$. In the case of $\alpha-$fairness, where $U'_{TCPr}(x_r) = w_r x_r^{-\alpha}$ this demand curve is given by:

$$f_{TCPr}(q_r) = \left(\frac{q_r}{w_r}\right)^{-1/\alpha},$$

and it will be a decreasing function of the price.

We propose to add, on top of this layer, a control law for n_r such that in equilibrium the rate $\varphi_r^* = n_r^* x_r^*$ is determined by the user level utility function U_r instead of the network layer utility U_{TCPr}. The main intuition is to use the current lower layer congestion price as a feedback signal from the network. If under the current conditions the resource allocation is such that user i with route r gets a rate φ_r which does not fulfill the user demand, the rate φ_r must be increased. In the light of Theorem 3.1, we can do so by increasing n_r.

The control law is as follows:

$$\dot{n}_r = k_r (U_r'^{-1}(q_r) - \varphi_r). \tag{4.9}$$

Combining it with the previous equations we arrive at the following dynamics for the system:

$$\dot{n}_r = k_r (U_r'^{-1}(q_r) - \varphi_r) \quad \forall r, \tag{4.10a}$$

$$x_r = f_{TCPr}(q_r) \quad \forall r, \tag{4.10b}$$

$$\varphi_r = n_r x_r \quad \forall r, \tag{4.10c}$$

$$y = R\varphi, \tag{4.10d}$$

$$\dot{p}_l = \gamma_l (y_l - c_l)^+_{p_l} \quad \forall l, \tag{4.10e}$$

$$q = R^T p. \tag{4.10f}$$

We would like to analyze the asymptotic properties of these dynamics. We begin by characterizing its equilibrium:

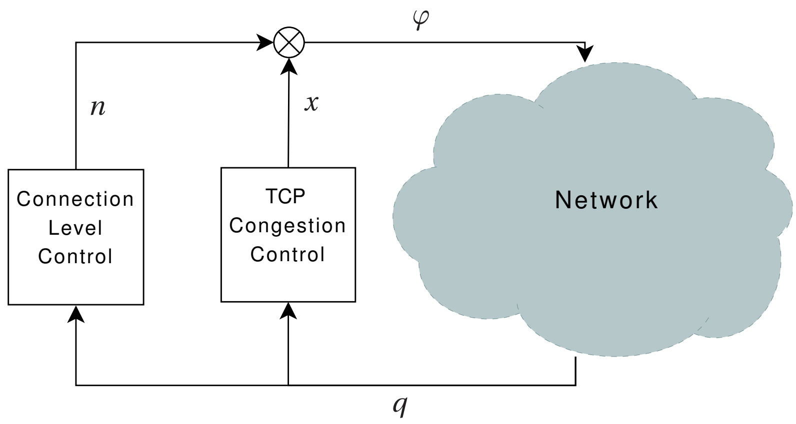

FIGURE 4.4: COMBINED CONNECTION LEVEL AND USER LEVEL CONTROL LOOP

Proposition 4.1. *The equilibrium of* (4.10)*,* (n^*, p^*)*, is such that the resulting* φ^* *is the solution of Problem 4.1 and* x^* *is the solution of Problem 3.1 for the given* n^**.*

Proof. Any equilibrium point (n^*, p^*) of the dynamics (4.10) must verify:

$$U_r'^{-1}(q_r^*) - \varphi_r^* = 0 \quad \forall r,$$
$$p^{*T}(y^* - c) = 0,$$
$$x_r^* = f_{TCPr}(q_r^*) = U_{TCPr}'^{-1}(q_r^*),$$

with $\varphi_r^* = n_r^* x_r^*$, $y^* = R\varphi^*$ and $q^* = R^T p^*$. The first two equations imply that φ^*, p^* satisfy the KKT conditions (4.2) of Problem 4.1. Moreover, the last two equations imply that (x^*, p^*) must satisfy also the KKT conditions of Problem 3.1. Since in this case both problems have unique solutions due to the strict concavity of the utility functions involved, the dynamics (4.10) have a single equilibrium which is the simultaneous solution of both problems. □

The above argument justifies why we can use the same route congestion price q_r to control both the rate of each connection and also the number of connections. The proposed control loop is best explained through Figure 4.4, where we can see that the connection level control is an outer control loop added to the current network congestion control algorithm.

4.3.1 Local stability in the network case

We would like to derive stability results for the system. We begin by local stability for a general network. To prove this result we shall use the *passivity approach*. A system with input u, state s and output v is called *passive* if there exists some storage function of the state $V(s)$ (similar to a Lyapunov function) such that:

$$\dot{V} \leqslant u^T v.$$

If the inequality is strict, the system is called strictly passive. For some background on passivity of dynamical systems we refer the reader to Appendix A.3. We note in particular that the negative feedback interconnection of two passive systems is stable.

The passivity approach was introduced in congestion control by [Wen and Arcak, 2003]. In their paper, the authors prove that the system $(\varphi - \varphi^*) \mapsto (q - q^*)$ with dual dynamics in the links (equations (4.10d), (4.10e) , (4.10f)) is passive with storage function:

$$V_{net}(p) = \sum_l \frac{1}{2\gamma_l}(p_l - p_l^*)^2.$$

Therefore, we have to check passivity of the map $(q - q^*) \mapsto -(\varphi - \varphi^*)$ with the new control law (4.9). Due to the decentralized nature of dynamics (4.10) we can check that the user system $(q_r - q_r^*) \mapsto -(\varphi_r - \varphi_r^*)$ is itself passive for each r since in that case there would be a storage function $V_r(n_r)$ such that:

$$\dot{V}_r(n_r) \leqslant -(q_r - q_r^*)(\varphi_r - \varphi_r^*),$$

and therefore:

$$V(n,p) = V_{net}(p) + \sum_r V_r(n_r)$$

will be a Lyapunov function for the system. Moreover, if the user system is strictly passive, we get asymptotic stability.

To obtain a local stability result, we shall check that the linearized dynamics around the equilibrium of the user part of the system is passive, by using the frequency domain characterization of strict passivity presented in Theorem A.5. We have to check that the input-output transfer function $H(s)$ of a linear system satisfies $\mathrm{Re}(H(j\omega)) \geqslant 0 \ \forall \omega \in \mathbb{R}$. If moreover the inequality is strict, the system is strictly passive.

We now calculate the transfer function of the linearized user part of the system and establish its strict passivity.

Lemma 4.2. *The linearization of the system* $(q_r - q_r^*) \mapsto -(\varphi_r - \varphi_r^*)$ *given by equations* (4.10a), (4.10b), (4.10c) *has a real positive transfer function and therefore is strictly passive.*

Proof. We drop the subscript r for simplicity. Let us denote by δ the deviation of the variables of the equilibrium values. From equations (4.10b) and (4.10c) we have:

$$\delta x = f'_{TCP}(q^*)\delta q = -a\delta q,$$
$$\delta\varphi = \delta(nx) = n^*\delta x + x^*\delta n,$$

where $a > 0$ since f_{TCP} is decreasing in q. Combining the above equations we have:

$$x^*\delta n = \delta\varphi + an^*\delta q. \tag{4.11}$$

Passing equation (4.10a) into the Laplace domain and linearizing we have:

$$s\delta n = k[(U'^{-1})'(q^*)\delta q - \delta\varphi] = -kb\delta q - k\delta\varphi,$$

where $b > 0$ since U'^{-1} is also decreasing in q. Multiplying both sides by x^* and subsituting (4.11) we get:

$$s(\delta\varphi + an^*\delta q) = -kbx^*\delta q - kx^*\delta\varphi,$$

which can be rewritten as:

$$\frac{\delta\varphi}{\delta q} = -H(s) = -\frac{an^*s + kbx^*}{s + kx^*}.$$

So the linearized dynamics from $\delta q \mapsto -\delta\varphi$ behaves as a lead-lag system $H(s)$. It is easy to see that the curve $\mathrm{Re}(H(j\omega))$ is a circle in the right half plane through the points $H(0) = b > 0$ and $H(\infty) = an^* > 0$ and thus the transfer function is positive real. We conclude that the user system is strictly passive. □

Combining Theorem A.5 with Lemma 4.2 and the results from [Wen and Arcak, 2003] we arrive at the main result of this section:

Theorem 4.2. *The equilibrium of the system* (4.10) *is locally asymptotically stable.*

Remark 4.1. With the same ideas, if we replace the protocol demand curve f_{TCP} in equation (4.10b) by a primal adaptation algorithm given by:

$$\dot{x}_r = \beta_r(U'_{TCPr}(x_r) - q_r),$$

then it can be shown that the analog of Lemma 4.2 holds with $H(s)$ given by:

$$H(s) = \frac{(\beta n^* + kx^*a)s + k\beta x^*ab}{(s + \beta a)(s + kx^*)}.$$

This second order system has one zero and two poles. It can be seen that the zero is always to the left of the pole kx^* and thus the phase of the system will always be in $(-\pi, \pi)$, therefore it is also positive real. We conclude that the system (4.10) is also locally asymptotcally stable if a primal-type congestion control is in use.

4.3.2 Global stability in the single bottleneck case

Although the result derived in the previous subsection is only local, as we shall see, simulation results of the system dynamics indicate that stability holds globally as well. However, it is not easy to derive a global result since the local passivity approach does not give an appropriate Lyapunov function for the non-linear system.

To derive a global stability result we first perform a time scale separation analysis of the system. Typically, we can assume that the congestion control protocol operates on a faster timescale than the connection level control of n_r. Since congestion control is known to be stable, it can be assumed that users will control n_r by reacting to the equilibrium price $\hat{q}_r(n_r)$ from Problem 3.1 reached by congestion control. Below we formalize this approach in terms of the *singular perturbation theory* (c.f. [Khalil, 1996]) and derive a global stability result for the single bottleneck case, where the map $n_r \mapsto \hat{q}_r$ can be calculated explicitly.

Let us rewrite the dynamics (4.10) as a singularly perturbed system:

$$\dot{n}_r = k_r(U_r'^{-1}(q_r) - \varphi_r) \quad \forall r, \tag{4.12a}$$

$$\epsilon \dot{p}_l = \gamma_l (y_l - c_l)^+_{p_l} \quad \forall l, \tag{4.12b}$$

where as usual $x_r = f_{TCPr}(q_r)$, $\varphi_r = n_r x_r$, $y = R\varphi$ and $q = R^T p$.

As $\epsilon \to 0$ we can decompose the system in a *boundary layer* model representing the fast dynamics for p and a *reduced system* where n evolves as if p has reached equilibrium. The boundary layer system is simply, for fixed n:

$$\begin{aligned} \epsilon \dot{p}_l &= \gamma_l (y_l - c_l)^+_{p_l} \quad \forall l, \\ x_r &= f_{TCPr}(q_r), \quad \varphi_r = n_r x_r, \\ y &= R\varphi, \quad q = R^T p, \end{aligned}$$

which is a dual congestion control algorithm similar to (2.11) studied in Chapter 2. It is known to be asymptotically stable by Proposition 2.4. Its equilibrium $\hat{x}(n)$, $\hat{\varphi}(n)$, $\hat{p}(n)$ and $\hat{q}(n)$ is the solution of Problem 3.1. In particular the map $n \mapsto \hat{\varphi}(n)$ is the map Φ we studied in Section 3.2.

The reduced system is autonomous in the state n and its dynamics is:

$$\dot{n}_r = k(U_r'^{-1}(\hat{q}_r(n)) - \hat{\varphi}(n)). \tag{4.13}$$

The main advantage of the singular perturbation analysis is the fact that, if both the boundary layer system and the reduced model are asymptotically stable, we can conclude that the complete system (4.12) is asymptotically stable for small values of the perturbation parameter ϵ. We now present an example showing both the complete model and the reduced dynamics for a network situation, illustrating the stability of the system.

Example 4.3 (Controlling the number of connections in a parking-lot network). We consider a parking lot topology, with capacities and round trip delays as depicted in Figure 4.5. Each user opens TCP connections, which we model as having the demand curve:

$$f_{TCPr}(q) = \frac{K_r}{\sqrt{q_r}},$$

where K_r is chosen as in Mathis formula (2.14). In particular, connections in the the long route will have a larger RTT than the shorter ones, and thus K_r will be smaller.

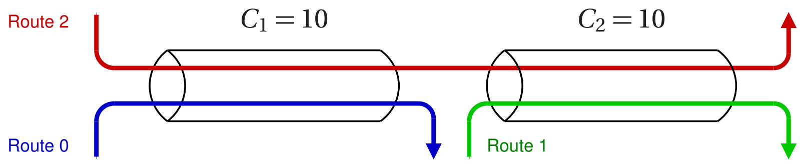

FIGURE 4.5: PARKING LOT TOPOLOGY.

If each user opens only one connection, the resulting allocation would be the solution of Problem 3.1 with $n_r = 1$ in each route. For this problem this allocation turns out to be $x_1^* = 2.6$, $x_2^* = x_3^* = 7.4$.

We would like to achieve proportional fairness, so we choose $U_r(\varphi_r) = \log\varphi_r$ for each user/route in this topology. The solution of the user welfare Problem 4.1 in this case is:

$$\varphi_1^* = 3.33, \quad \varphi_2^* = \varphi_3^* = 6.66.$$

In Figure 4.6 we show the results, starting from an initial condition of $n(0) = (1,2,3)$. We plot the time evolution of the system in n, p and φ. In solid lines, we plot the complete dynamics (4.10). In dashed lines we plot the reduced dynamics (4.13) with time scale separation. Note that both dynamics converge to the desired equilibrium, and show similar behavior. Extensive testing with several initial conditions show that the system indeed behaves in a stable manner in all cases.

In a network situation, however, the reduced system (4.13) is difficult to analyze because $\hat{q}$ and $\hat{\varphi}$ are difficult to express in terms of n, since they are only characterized by Problem 3.1. We now present a stability result for a single-bottleneck case, where the allocation can be calculated explicitly.

We analyze this system under the following assumptions:

- We consider a network with a single bottleneck of capacity c which is shared by N users, each with user utility function U_r, $r = 1,\ldots,N$.
- We assume that the congestion control algorithm of all users follows the same demand curve f_{TCP}. For instance, this could correspond to everyone using TCP-Reno at the transport layer, with the same round trip times.

Under these hypotheses, there will be a single price and, since there is a common TCP demand curve, the individual flow rates will satisfy:

$$\hat{x}_r(n) = \frac{c}{\sum_{i=1}^N n_i} := \hat{x}(n) \quad \forall r \tag{4.14}$$

Therefore, $\hat{q}(n) = U'_{TCP}(\hat{x}(n))$ and $\hat{\varphi}_r(n) = n_r\hat{x}(n)$, which substituted in (4.13) give an autonomous system in the state n. We have the following:

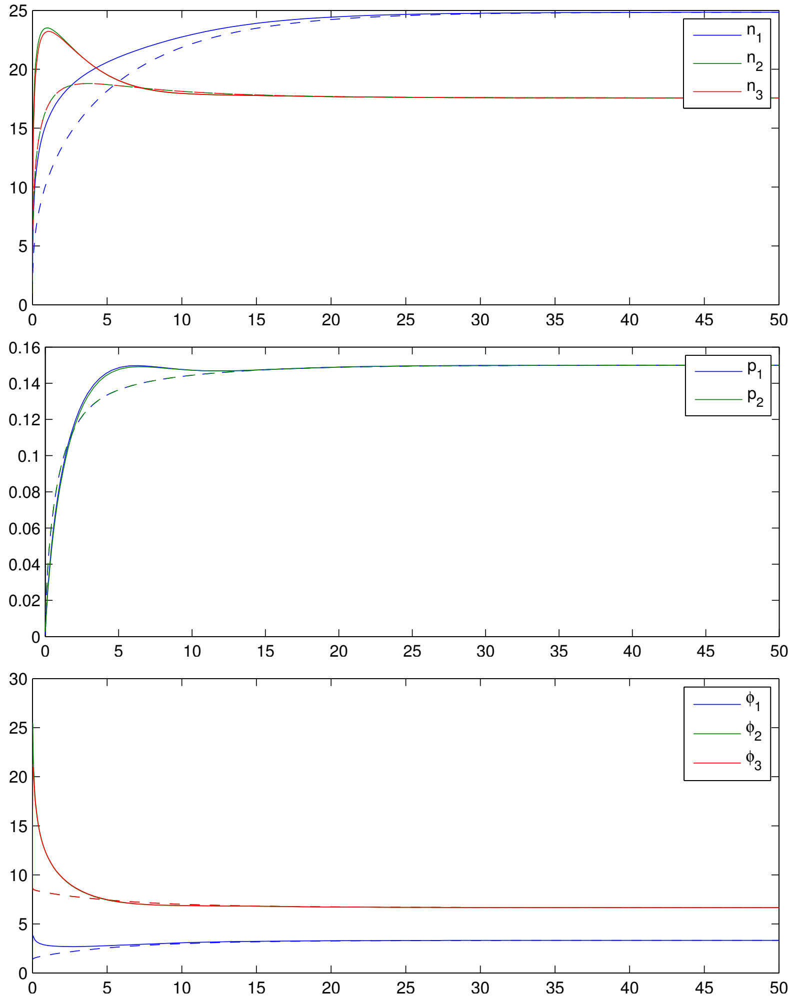

FIGURE 4.6: RESULTS FOR THE PARKING LOT TOPOLOGY. IN SOLID LINES, THE COMPLETE DYNAMICS, IN DOTTED LINES, THE REDUCED MODEL.

Theorem 4.3. *The solutions of the system* (4.13) *complemented with* (4.14) *converge to an equilibrium point* n^* *such that* $\varphi^* = \hat{\varphi}(n^*)$ *is a solution of Problem 4.1, i.e. maximizes the user welfare.*

Proof. First note that an equilibrium of (4.13) together with (4.14) and the lower layer equilibrium automatically ensures the KKT conditions (4.2). We thus focus on the stability.

Denote by $|n| = \sum_{r=1}^{N} n_r$ the total number of connections. This way $\hat{x}(n) = c/|n|$. Adding equations (4.13) in r we obtain:

$$\dot{|n|} = k\left[\sum_{r=1}^{N} U_r'^{-1}(\hat{q}(n)) - \sum_{r=1}^{N} \hat{\varphi}(n)\right].$$

Note that, whenever $n \neq 0$ we have $\sum_{r=1}^{N} \hat{\varphi}(n) = c$, and that:

$$\hat{q}(n) = U'_{TCP}(\hat{x}(n)) = U'_{TCP}(C/|n|).$$

So the dynamics of $|n|$ verify:

$$\dot{|n|} = k \left[\sum_{r=1}^{N} g_r \left(\frac{c}{|n|} \right) - c \right], \tag{4.15}$$

where $g_r(x) = U_r'^{-1}(U'_{TCP}(x))$. The function g_r is increasing since it is the composition of two decreasing functions. Note that (4.15) is an autonomous differential equation in the scalar variable $|n|$.

Denoting by $|n^*|$ the equilibrium of (4.15), define the following candidate Lyapunov function:

$$V_1(|n|) = \frac{1}{2k}(|n| - |n^*|)^2. \tag{4.16}$$

By differentiating along the trajectories we obtain:

$$\begin{aligned}
\dot{V}_1 &= (|n| - |n^*|) \left[\sum_{r=1}^{N} g_r \left(\frac{c}{|n|} \right) - c \right] \\
&= (|n| - |n^*|) \left[\sum_{r=1}^{N} \left(g_r \left(\frac{c}{|n|} \right) - g_r \left(\frac{c}{|n^*|} \right) \right) \right],
\end{aligned}$$

where the last step uses the equilibrium condition. Noting that each g_r is decreasing in $|n|$, we have $\dot{V}_1 \leqslant 0$ with the inequality being strict if $|n| \neq |n^*|$. This shows that $|n|$ converges to equilibrium, and in particular the rate of each connection $\hat{x}(n) \to x^* = c/|n^*|$.

We return now to equation (4.13). By defining $\delta n_r = n_r - n_r^*$ we can rewrite it as:

$$\begin{aligned}
\dot{\delta n_r} &= k[g_r(\hat{x}(n)) - n_r \hat{x}(n)] \\
&= -k\hat{x}(n)\delta n_r + k[g_r(\hat{x}(n)) - n_r^* \hat{x}(n)].
\end{aligned}$$

The term $b_r(t) := g_r(\hat{x}(n)) - n_r^* \hat{x}(n)$ vanishes as $t \to \infty$ since $\hat{x}(n) \to x^*$ and $g_r(x^*) = \varphi_r^* = n_r^* x^*$.

Now for each r take the Lyapunov function $V_2(\delta n_r) = \frac{1}{2k}(\delta n_r)^2$, its derivative is:

$$\dot{V}_2 = -\hat{x}(n)(\delta n_r)^2 + b_r(t)\delta n_r \tag{4.17}$$

Consider now $\varepsilon > 0$ arbitrary, and choose t_0 such that $\hat{x}(n(t)) > \frac{x^*}{2}$ and $|b_r(t)| < \frac{x^*}{4}\varepsilon$ whenever $t > t_0$. Then, in the region where $|\delta n_r| \geqslant \varepsilon$ and for $t > t_0$ we have that:

$$|b_r(t)\delta n_r| < \frac{x^*}{4}\varepsilon|\delta n_r| \leqslant \frac{x^*}{4}(\delta n_r)^2.$$

We can now bound $\dot{V}_2$ as follows:

$$\begin{aligned}
\dot{V}_2 &= -\hat{x}(n)(\delta n_r)^2 + b_r(t)\delta n_r \\
&< -\frac{x^*}{2}(\delta n_r)^2 + \frac{x^*}{4}(\delta n_r)^2 \\
&= -\frac{x^*}{4}(\delta n_r)^2 \\
&\leqslant -\frac{x^*}{4}\varepsilon^2.
\end{aligned}$$

The above inequality proves that V_2 is strictly decreasing along the trajectories, and thus eventually, δn_r will reach the set $|\delta n_r| < \varepsilon$ in finite time. Since the above is true for any $\varepsilon > 0$. we have that $\delta n_r \to 0$ asymptotically. $\square$

Extending the global stability results for more general topologies proves to be a difficult task. In the next section, we shall follow a different approach to derive a suitable dynamics for n, which would enable us to prove global stability results.

4.4 Controlling the number of flows: the multi-path case

So far, we have studied a mechanism to control the aggregate rates φ_r directly in order to reach the optimum of the User Welfare Problem 4.1 in a multi-path setting, which led us to the dynamics (4.4). We also analyzed the possibility of controlling n_r as a means to achieve the optimum of Problem 4.1 by indirectly controlling φ_r. We would like to combine both ideas in a single control algorithm for n_r which can be used in a multi-path setting.

We begin by noting the following simple relationship between n_r and φ_r:

$$\dot{\varphi}_r = (\dot{n_r x_r}) = x_r \dot{n}_r + n_r \dot{x}_r. \tag{4.18}$$

We would like to control n_r such that φ_r follows a control law of the type discussed in Section 4.2. Consider the following law for n_r:

$$\dot{n}_r = n_r \left[k_r(U_i'(\varphi^i) - q_r - \nu \dot{q}_r) - \frac{\dot{x}_r}{x_r} \right]. \tag{4.19}$$

With this choice of $\dot{n}_r$, valid whenever $x_r > 0$, we can substitute in (4.18) to get:

$$\begin{aligned}
\dot{\varphi}_r &= x_r \dot{n}_r + n_r \dot{x}_r \\
&= n_r x_r \left[k_r(U_i'(\varphi^i) - q_r - \nu \dot{q}_r) - \frac{\dot{x}_r}{x_r} \right] + n_r \dot{x}_r \\
&= k_r \varphi_r \left[U_i'(\varphi^i) - q_r - \nu \dot{q}_r \right],
\end{aligned}$$

which is on the form of the control for φ_r analyzed in (4.4), with k_r replaced by $k_r \varphi_r$.

For practical purposes, it is better to express the dynamics above in terms of the route price q_r. We will do so in the case of U_{TCP} being in the $\alpha-$family. We have:

$$x_r = f_{TCPr}(q_r) = \left(\frac{q_r}{w_r}\right)^{-1/\alpha},$$
$$\dot{x}_r = f'_{TCPr}(q_r)\dot{q}_r = -\frac{1}{\alpha}\left(\frac{q_r}{w_r}\right)^{-1/\alpha-1}\dot{q}_r,$$

and therefore:

$$\frac{\dot{x}_r}{x_r} = -\frac{w_r}{\alpha}\frac{\dot{q}_r}{q_r}.$$

The complete dynamics for the system now follows:

$$\dot{n}_r = n_r\left[k_r(U'_i(\varphi^i) - q_r - \nu\dot{q}_r) + \frac{w_r\dot{q}_r}{\alpha q_r}\right] \quad \forall r, \tag{4.20a}$$
$$\varphi_r = n_r f_{TCPr}(q_r) \quad \forall r, \tag{4.20b}$$
$$y = R\varphi, \tag{4.20c}$$
$$\dot{p}_l = (y_l - c_l)^+_{p_l} \quad \forall l, \tag{4.20d}$$
$$q = R^T p. \tag{4.20e}$$

We want to derive a global stability result for these dynamics. We begin by extending the result of Theorem 4.1 to the case where the controller gain is $k_r\varphi_r$. The result is the following:

Theorem 4.4. *Consider the dynamics* (4.4) *with the first equation replaced by:*

$$\dot{\varphi}_r = k_r\varphi_r\left[U'_i(\varphi^i) - q_r - \nu\dot{q}_r\right]. \tag{4.21}$$

Then, for any initial condition $\varphi(0)$ such that $\varphi_r(0) > 0\ \forall r$, the system converges to a solution of the User Welfare Problem 4.1.

Proof. The main issue with equation (4.21) is that if $\varphi_r = 0$ at some time t, then φ_r remains 0 in the future, even if the desired allocation is not 0 for that route. We show that this indeed cannot happen if we start at an initial condition with $\varphi_r > 0$.

Consider the following modification of the Lyapunov function $V(\varphi, p)$ defined in (4.6):

$$V(\varphi,p) = \sum_r \int_{\varphi_r^*}^{\varphi_r} \frac{(\sigma - \varphi_r^*)}{k_r\sigma}d\sigma + \sum_l \frac{1}{2\gamma_l}(p_l - p_l^*)^2 + \sum_l \nu(c_l - y_l^*)p_l, \tag{4.22}$$

which is well defined at least when $\varphi_r > 0\ \forall r$. Note that, when differentiating along the trajectories, $\dot{V}$ satisfies equation (4.7) and so it will be decreasing along trajectories by the same arguments of Theorem 4.1.

Assume that φ_r^* is such that $\varphi_r^* > 0$, then the corresponding term in the Lyapunov function is:

$$V_r(\varphi_r) = \int_{\varphi_r^*}^{\varphi_r} \frac{\sigma - \varphi_r^*}{k_r\sigma}d\sigma = \frac{1}{k_r}(\varphi_r - \varphi_r^*) + \frac{\varphi_r^*}{k_r}\log\left(\frac{\varphi_r^*}{\varphi_r}\right).$$

Therefore, as $\varphi_r \to 0$, $V_r(\varphi_r) \to \infty$. Also if $\varphi_r \to \infty$, $V_r(\varphi_r) \to \infty$.

If now $\varphi_r^* = 0$ we have:

$$V_r(\varphi_r) = \int_{\varphi_r^*}^{\varphi_r} \frac{\sigma - \varphi_r^*}{k_r \sigma} d\sigma = \frac{1}{k_r} \varphi_r,$$

so this term is well defined when $\varphi_r \to 0$ and we can extend $V(\varphi, p)$ continuously in this case. Note also that $V_r(\varphi_r) \to \infty$ when $\varphi_r \to \infty$.

We conclude that the set $\{V \leqslant K\}$ is compact for any $K > 0$. Moreover, these sets do not contain the region $\varphi_r = 0$ for those routes such that $\varphi_r^* > 0$. Therefore, for any initial condition (φ_0, p_0) with φ_0 in the positive orthant, the set $\{V \leqslant V(\varphi_0, p_0)\}$ is compact, and forward invariant due to V decreasing along trajectories.

The result now follows by LaSalle invariance principle: the system will converge to the set where $\dot{V} = 0$ contained in $\{V \leqslant V(\varphi_0, p_0)\}$. Since we analyzed in the proof of Theorem 4.1 that this can only happen with $\varphi = \varphi^*$, a solution of the Problem 4.1, this concludes the proof. □

We now state the main result of this section:

Theorem 4.5. *The dynamics* (4.20)*, starting from any initial condition with $n_r > 0$ and $q_r > 0$ $\forall r$, will converge to an equilibrium point which achieves the optimum of Problem 4.1.*

Proof. It is easy to see that, if $n_r > 0$ and $q_r > 0$ then $\varphi_r > 0$, so the initial condition for φ_r verifies the hypotheses of Theorem 4.4.

As discussed before, (4.20) imply that $\dot{\varphi}_r$ must verify (4.21) and due to Theorem 4.4, φ_r will converge to a solution of Problem 4.1. Therefore, n_r, p_l will converge to a point such that user welfare is maximized. □

Observe that the control law for n_r in equation (4.20a) is similar to to the proposed control law in the single path situation, but with a derivative action. However, the derivative terms play opposing roles, and this suggests that the simpler control law:

$$\dot{n}_r = k_r n_r (U_i'(\varphi^i) - q_r) \tag{4.23}$$

should be also a suitable way to drive the system to the user welfare equilibrium. In fact, when translated to φ_r, the control law (4.23) yields very similar dynamics to (4.4) with a damping parameter $\nu_r = \frac{w_r}{\alpha k_r q_r}$. However, this damping parameter is time varying and route dependent, which is not compatible with our earlier stability argument.

One can also try to perform a time-scale separation analysis of (4.20) based on singular perturbation, as in Section 4.3. However, the derivative action here does not allow to treat the n_r variable as slowly varying. The time scale separation analysis can be performed on (4.23),

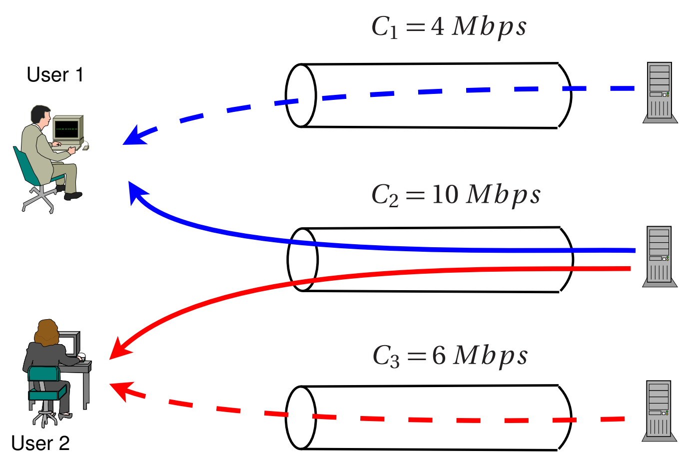

FIGURE 4.7: DOWNLINK PEER-TO-PEER EXAMPLE NETWORK

but in this case global stability results are harder to obtain, due to the fact that the map $n \mapsto \varphi$ is difficult to analyze.

So we have a choice between a more complicated law (4.20a) with guaranteed stability properties, and the simpler one (4.23). Below we show a simulation example illustrating that the simpler dynamics (4.23) is sufficient in some cases:

Example 4.4 (Peer-to-peer downloading example). The algorithms derived in this chapter can be applied to a peer-to-peer file exchange environment. In such a system, users may open multiple connections to several peers to download shared data files. It is natural to consider the utility of a user based on its total download rate, rather on the rate of individual connections. We thus have a multi-path instance of the User Welfare Problem 4.1, where the set of flows per user share a common destination but with different sources.

It is arguable that in a peer-to-peer environment, a certain level of cooperation can be obtained, at least among users that access the system with a common application. This application could be pre-programmed with a connection-level controller, that opens and shuts connections according to the congestion price it measures on each route. There is also peer pressure to respect this control.

The algorithm in this case would work as follows: user i measures the total download rate φ^i and the congestion price q_r on each of its routes. Then it applies the control law (4.20a) (or the simpler control (4.23)) to regulate the number of active connections per route. Each connection will be individually controlled by standard TCP, requiring no change in current protocols. For instance, if a loss-based TCP is used, the price q_r would be the packet loss fraction of the route, which can be deduced from the number of retransmissions.

As an example, consider the network depicted in Figure 4.7, where two users share a total

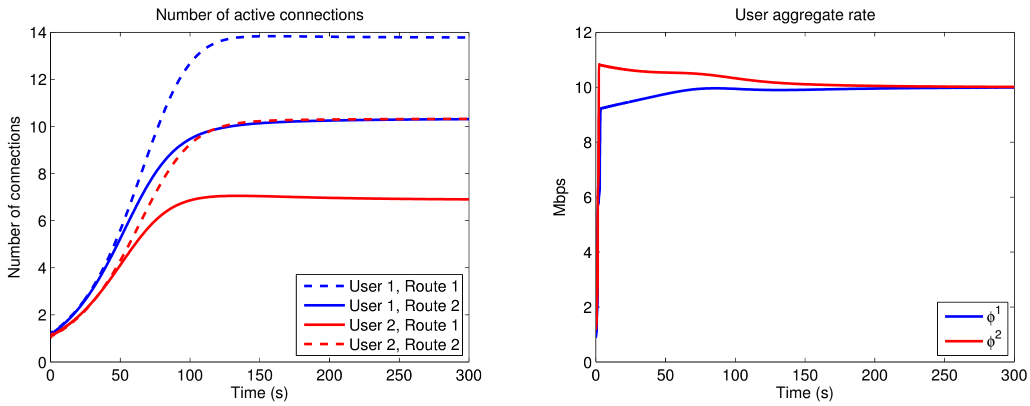

FIGURE 4.8: NUMBER OF CONNECTIONS ON EACH ROUTE AND AGGREGATE RATE (φ^i) FOR THE PEER-TO-PEER DOWNLINK EXAMPLE.

of four links with different capacities, in order to reach their servers. We choose the user utility $U_i(\varphi^i) = \log(\varphi^i)$ to impose proportional fairness and we suppose that different routes have TCP-Reno like demand curves, with $\alpha = 2$ but different weights, in order to represent different round trip times on each route.

It is easy to see that the solution of Problem 4.1 is this case satisfies $\varphi^{1,*} = \varphi^{2,*} = 10Mbps$. In Figure 4.8 we plot the results, starting with an initial condition of 1 flow on each route. As we can see, the algorithm of (4.23) finds the appropriate number of flows per route such that the resource allocation is optimal.

In the following section, a packet level implementation of the control law (4.23) is discussed, and we compare its performance with the above example.

4.5 Packet level simulation

We implemented a packet level simulation of the control law (4.23) using the network simulator ns-2 [McCanne and Floyd, 2000]. With this tool, we can replicate the behavior of congestion control protocols over a network at the packet level, which takes into account the full TCP dynamics.

We simulated the topology of Figure 4.7, with each user beginning with a single TCP connection per route. The congestion control algorithm is TCP-Newreno, which is the latest version of the standard TCP congestion control.

With this choice of TCP, the route price is the packet loss rate on the route. We programmed a user instance, that measures the loss rate on each of its routes by counting packet retransmissions at the transport layer. It also keeps track of the total rate per route φ_r, and therefore is able to calculate φ^i. Each user then maintains a variable `targetN(r)` for each route, which is the target number of connections. This variable is updated periodically by integrating the control law (4.23). As in the previous example, we choose logarithmic utilities to impose proportional fairness in the network.

Each second, the user chooses whether to open or close a connection or route r based on the current value of `targetN(r)`, rounded to he nearest integer. Results are shown in Figure 4.9, which shows that the number of connections approximately track the equilibrium values found in the previous example. The aggregate rate of each user also converges towards the equilibrium allocation of $\varphi^{1,*} = \varphi^{2,*} = 10Mbps$.

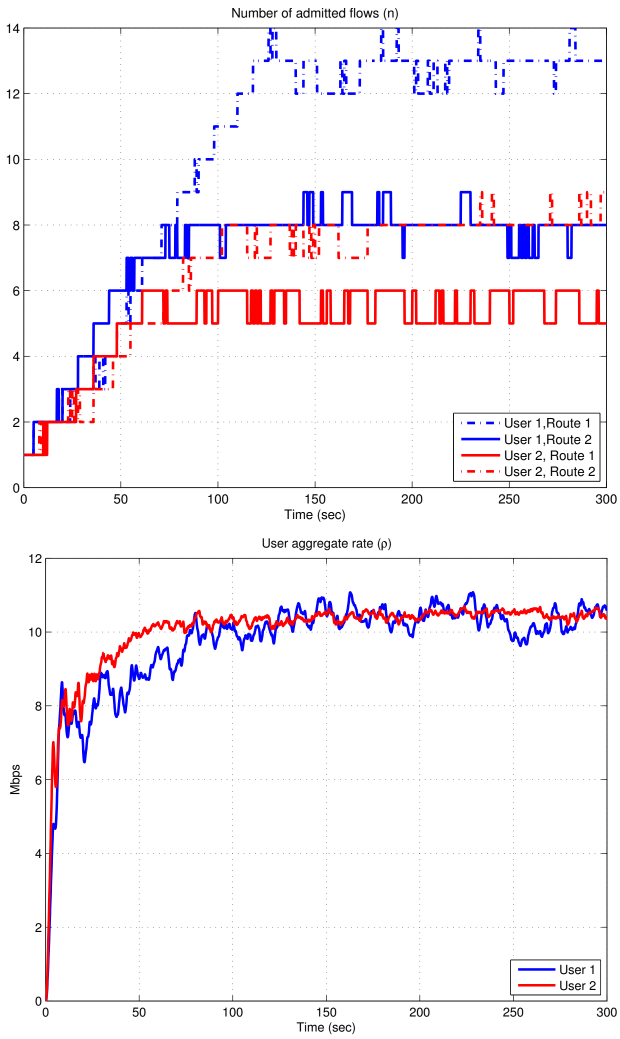

FIGURE 4.9: NUMBER OF CONNECTIONS ON EACH ROUTE AND AGGREGATE RATE (φ^i) IN AN NS-2 SIMULATION FOR THE PEER-TO-PEER DOWNLINK EXAMPLE.

5
Utility based admission control

The analysis of Chapter 4 applies to the case of networks where users cooperate by controlling the number of individual TCP connections in order to maintain some notion of fairness between their total rates.

However, this kind of user control is often not possible. An application may not be capable of tuning the number of connections in such a fine way. In particular, it may not be practical to *terminate* active connections, as may be required by the proposed control algorithms, which may need to reduce n_r. There are also information requirements for the end systems such as being aware of the user aggregate rate φ^i, which may be difficult to monitor. Also, and more categorically, a greedy user may simply refuse to cooperate, and increment its number of ongoing connections in order to grab a larger share of bandwidth, as we discussed in Theorem 3.1.

In these situations, the network should be able to respond and protect itself by imposing some admission control on the incoming connections. Previous work on admission control (see e.g. [Massoulié and Roberts, 1999]) has been mainly motivated by the requirement of maintaining network stability under stochastic models of load, such as those presented on Section 3.3. However, any admission control rule that limits the maximum number of ongoing connections in each route will guarantee stability. The remaining question is whether there is an admission control rule that, in addition to guaranteeing stability, is able to drive an overloaded system into a fair operating point, from a user centric perspective.

In this chapter we focus on this task, based on the notion of user centric fairness introduced in Chapter 4, and using the knowledge we acquired from the proposed dynamics for n_r to drive the system to the optimal user welfare allocation.

The chapter is organized as follows: In Section 5.1 we introduce the proposed admission control rule in the single path scenario. We proceed to analyze this rule under stochastic loads by using the technique of fluid limits, and characterize the equilibrium of these dynamics, as well as showing that this equilibrium is locally asymptotically stable. In Section 5.5 we extend the proposed admission control rule to the multi path setting and also characterize the equilibrium attained. Finally, in Section 5.6 we present several scenarios of packet-level simulation of the algorithms, showing that they can be implemented in real networks.

5.1 Admission control in the single path case

The models in Chapter 4 are suitable for a network where users may open or close connections at will. However, a more realistic setting is one in which users open new connections at random arrival times, and each connection brings a random amount of workload.

This type of models where introduced in [Massoulié and Roberts, 1999], and were reviewed in Section 3.3. Also in [Massoulié and Roberts, 1999], the idea of admission control was advocated. In such a system, the network itself may decide whether each incoming connection is allowed into the network. Admission control allows the system to be stabilized even when the external load on each link is over the link capacity.

What we would like to derive is a decentralized admission control rule that can be enforced at the network edge, such that in case of overload the system allocates resources according to the User Welfare Problem. We concentrate here on the single path case, so each user i can be identified with a route r. Each user is therefore assigned a utility function U_r.

From the analysis of Section 4.3, we see that in order to achieve fairness, each user must increase its number of connections whenever $U_r'^{-1}(q_r) > \varphi_r$, or equivalently, $U_r'(\varphi_r) > q_r$. If on the other hand, the inequality reverses, the number of connections must be decreased. It is therefore natural to propose the following admission control rule: whenever a new connection arrives on route r,

$$\begin{aligned} &\text{if } U_r'(\varphi_r) > q_r \quad \rightarrow \quad \text{admit connection,} \\ &\text{if } U_r'(\varphi_r) \leqslant q_r \quad \rightarrow \quad \text{discard connection.} \end{aligned} \tag{5.1}$$

We call the preceding rule *Utility Based Admission Control*.

This rule imposes a limit on the number of connections a given user is allowed into the network. A strategic user will not get a larger share of bandwidth by simply increasing the number of connections, as discussed in Theorem 3.1, because eventually the admission condition (5.1) will not be met, due to the increasing price on the route. In a scenario where all users are greedy, the network will operate in the region where $U_r'(\varphi_r) = q_r$, which are the KKT conditions of Problem 4.1, and thus the resulting allocation will be fair. Below we formalize these arguments through a stochastic model of the system.

5.2 Stochastic model and its stability

We will make a stochastic model for this system, similar to the one presented in Section 3.3. Assume that each user on route r opens connections through the network. Fresh connections arrive as a Poisson process of intensity λ_r. Each connection brings an exponentially distributed workload of mean $1/\mu_r$. Connection arrival and job sizes are independent, and also independent between users.

We also make the same time scale separation assumption, i.e. congestion control operates on a faster timescale that the connection level processes, and therefore the rate at which each connection is served is determined by the solution of the TCP congestion control Problem 3.1. For a given state $n = (n_r)$ of connections on each route, $x(n) = (x_r(n))$ will denote the solution of Problem 3.1. We denote by $\varphi_r(n) = n_r x_r(n)$ the aggregate rate on route r, and $q_r(n)$ the route price. We assume that in Problem 3.1, the utilities U_{TCPr} come from the $\alpha-$fair family with $\alpha > 0$.

With each user, we associate an increasing and concave utility function U_r (here r denotes the route and the user), that represents the user valuation of bandwidth. The admission control rule (5.1) amounts to comparing $U_r'(\varphi_r(n))$ with $q_r(n)$ in each state, and only admit connections whenever the admission condition is satisfied.

With the above assumptions, the process $n(t)$ is a continuous time Markov chain with state space $\mathbb{N}^N$, with N being the number of users, and transition rates:

$$
\begin{aligned}
n &\mapsto n + e_r \quad \text{with rate } \lambda_r \mathbf{1}_{\{U_r'(\varphi_r) > q_r\}}, \\
n &\mapsto n - e_r \quad \text{with rate } \mu_r \varphi_r,
\end{aligned}
\tag{5.2}
$$

where e_r is the vector with a 1 in coordinate r and 0 elsewhere, and $\mathbf{1}_A$ is the indicator function.

The model (5.2) is similar to the one presented in Section 3.3, but with the admission rule (5.1) incorporated. Recall that, without the admission condition, the Markov chain is stable (positive recurrent) only if the link loads $\rho^l = \sum_r R_{lr}\lambda_r/\mu_r < c_l$. Admission control should stabilize the system in any situation. We now prove that this is indeed the case, for the non-trivial admission rule (5.1).

We need the following Lemmas on the map $n \mapsto \varphi$ for $\alpha-$fair resource allocations, proved in [Kelly and Williams, 2004]:

Lemma 5.1 ([Kelly and Williams, 2004]). *The function $n \mapsto \varphi_r(n)$ is continuous at each n where $n_r > 0$.*

Lemma 5.2 ([Kelly and Williams, 2004]). *For any scale parameter $L > 0$, $\varphi(Ln) = \varphi(n)$.*

We have the following:

Proposition 5.1. *The Markov chain* (5.2) *is stable.*

Proof. We shall prove that there is a finite set that the process $n(t)$ defined by (5.2) cannot leave. It follows that the irreducible set containing $n = 0$ is finite and starting with any initial condition in that set (in particular, an empty network), the process will reach a stable state.

Consider the "box" set $S = \{n : ||n||_\infty \leqslant n_0\}$, with n_0 to be chosen later. We would like to prove that the process cannot leave S. Since (5.2) defines a multidimensional birth and death process, we have to prove that, for n such that $||n||_\infty = n_0$ the process cannot leave the box. To do so, consider the facet:

$$S_r = \{n : ||n||_\infty = n_0,\, n_r = n_0\}.$$

Now for each r we define:

$$\bar{\varphi}_r = \max\{\varphi_r(n) : ||n||_\infty = 1,\, n_r = 1\},$$
$$\underline{\varphi}_r = \min\{\varphi_r(n) : ||n||_\infty = 1,\, n_r = 1\}.$$

Since φ_r is continuous by Lemma 5.1 in the compact set of the right, these values are well defined. Moreover, $\underline{\varphi}_r > 0$ since $n_r > 0$.

If $n \in S_r$, using Lemma 5.2, we have:

$$\varphi_r(n) = \varphi_r\left(\frac{n}{n_0}\right) \geqslant \underline{\varphi}_r \Rightarrow U_r'(\varphi_r(n)) \leqslant U_r'(\underline{\varphi}_r), \tag{5.3}$$

since U_r' is decreasing.

We also have for any $n \in S_r$:

$$\bar{\varphi}_r \geqslant \varphi_r\left(\frac{n}{n_0}\right) = \varphi_r(n) = n_r f_{TCPr}(q_r(n)) = n_0 \left(\frac{q_r(n)}{w_r}\right)^{-1/\alpha},$$

and thus:

$$q_r(n) \geqslant \frac{w_r n_0^\alpha}{\bar{\varphi}_r^\alpha}. \tag{5.4}$$

Now choose n_0 such that for every route r the following is satisfied:

$$\frac{w_r}{\bar{\varphi}_r^\alpha} n_0^\alpha > U_r'(\underline{\varphi}_r). \tag{5.5}$$

Then, for that choice of n_0 and every $n \in S_r$ we can combine (5.3), (5.4) and (5.5) to yield:

$$U_r'(\varphi_r(n)) \leqslant U_r'(\underline{\varphi}_r) < \frac{k_r^\alpha}{\bar{\varphi}_r^\alpha} n_0^\alpha \leqslant q_r(n),$$

and therefore the admission condition is not satisfied. The transition $n \mapsto n + e_r$ is not allowed in S_r and thus the system must remain in the set S. $\square$

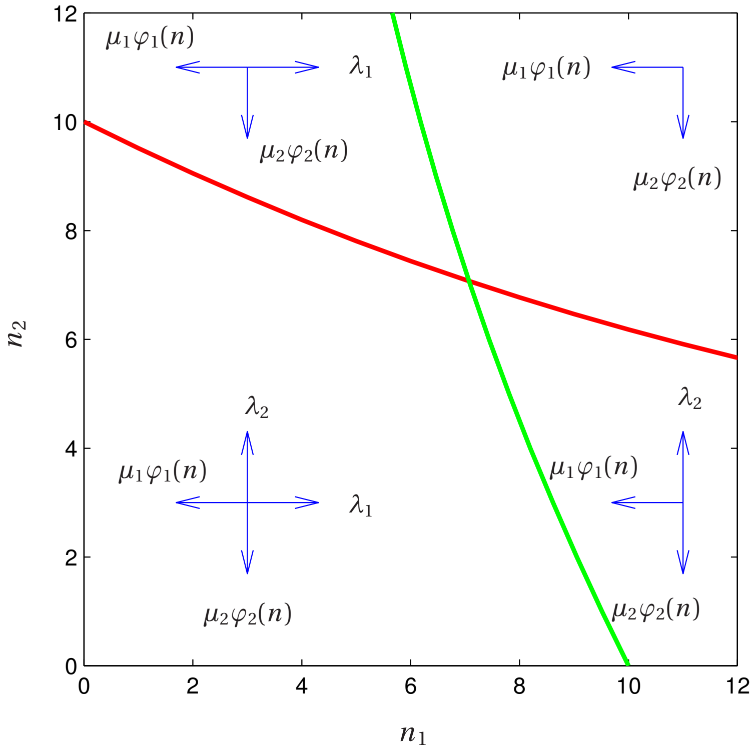

FIGURE 5.1: STATE SPACE, ALLOWED TRANSITIONS AND ADMISSION CONTROL CURVES FOR A SINGLE BOTTLENECK LINK WITH TWO USERS. $\alpha = 2$ AND $U_r = \log(\varphi_r)$.

As an example, in Figure 5.1 we plot the state space and admission control curves of the model defined by (5.2) for the case of a single bottleneck link with two users, $\alpha = 2$ and user utilities $U_r(\varphi_r) = \log(\varphi_r)$.

5.3 Fluid limit derivation for the single path case

Now that stability is assured, we proceed to analyze the fairness of the admission control policy. We will do so by deriving a suitable fluid model for the system (5.2). The model is based in what we call *large network asymptotic*. The main idea is to scale the network size appropriately, by enlarging the capacity of the links and the arrival rate of flows, such that a law of large number scaling occurs. An important remark is that, for the scaling to work appropriately, we also have to scale the user preferences with the size of the network.

More formally, we take a scaling parameter $L > 0$ and consider a network with link capacities scaled by L, i.e. $c_l^L = Lc_l$. We also assign each user a utility $U_r^L(\varphi) = LU_r(\varphi/L)$. Note that with this choice, the utility functions verify the following scaling property:

$$(U_r^L)'(L\varphi) = U_r'(\varphi).$$

That is, the user marginal utility for obtaining L times bandwidth in the scaled network is the same that the marginal utility for the original amount in the original network.

We denote by $\varphi^L(n)$ and $q^L(n)$ the rate allocation and route prices in the scaled network, and as before $\varphi(n)$ and $q(n)$ denote the original values, i.e. $L=1$. The following relationships hold:

Lemma 5.3. *For any $L>0$, the resource allocation and route prices satisfy:*

$$\varphi_r^L(Ln) = L\varphi_r(n) \quad \forall r,$$
$$q_r^L(Ln) = q_r(n) \quad \forall r.$$

Proof. We recall that $\varphi_r(n)$ and $q_r(n)$ satisfy the KKT conditions for Problem 3.1:

$$U'_{TCPr}\left(\frac{\varphi_r(n)}{n_r}\right) = q_r(n) \quad \forall r \text{ with } n_r > 0,$$
$$p_l(n)(y_l(n) - c_l) = 0 \quad \forall l.$$

We now want to verify that the choice $\varphi_r^L(Ln) = L\varphi_r(n)$ and $p_l^L(Ln) = p_l(n)$ satisfy the KKT conditions of the scaled version of Problem 3.1, i.e. with c_l replaced by Lc_l and n by Ln. Noting that with the above choice $y_l^L(Ln) = Ly_l(n)$ and using the above equations we have:

$$U'_{TCPr}\left(\frac{\varphi_r^L(n)}{Ln_r}\right) = U'_{TCPr}\left(\frac{\varphi_r(n)}{n_r}\right) = q_r(n) = q_r^L(Ln) \quad \forall r \text{ with } Ln_r > 0,$$
$$p_l^L(Ln)(y_l^L(Ln) - Lc_l) = p_l(n)(Ly_l(n) - Lc_l) = Lp_l(n)(y_l(n) - c_l) = 0 \quad \forall l.$$

Therefore, φ^L and p^L satisfy the KKT conditions of the scaled version of Problem 3.1, and therefore are the resource allocation and route prices of the scaled network. □

Using the above relationships, we now derive the fluid model of the system. To avoid technicalities, we shall replace the indicator function in (5.2) by a smooth approximation f_ϵ, such that $f_\epsilon(x) = 1$ if $x > \epsilon$, and $f_\epsilon(x) = 0$ if $x < -\epsilon$. The original model can therefore be approximated by:

$$\begin{aligned} n &\mapsto n + e_r \quad \text{with rate } \lambda_r f_\epsilon(U_r'(\varphi_r) - q_r), \\ n &\mapsto n - e_r \quad \text{with rate } \mu_r \varphi_r. \end{aligned} \tag{5.6}$$

Note that, as $\epsilon \to 0$ the above model approximates (5.2).

We have the following result:

Theorem 5.1. *Consider a sequence of processes $n^L(t)$ governed by equations* (5.6) *with $\lambda_r^L = L\lambda_r$, $c_l^L = Lc_l$, $\mu_r^L = \mu_r$, and utility functions that satisfy*

$$U_r^L(\varphi) = LU_r\left(\frac{\varphi}{L}\right),$$

with $L > 0$ a scaling parameter. Consider also a sequence of initial conditions $n^L(0)$ such that $n^L(0)/L$ has a limit $\bar{n}(0)$. Then, the sequence of scaled processes:

$$\bar{n}^L(t) = \frac{n^L(t)}{L}$$

is tight and any weak limit point of the sequence where $n_r > 0$ converges as $L \to \infty$ to a solution of the following differential equation:

$$\dot{n}_r = \lambda_r f_\epsilon(U_r'(\varphi_r) - q_r) - \mu_r \varphi_r, \tag{5.7}$$

where $\varphi(n)$ and $q(n)$ are the allocation maps for $L = 1$.

Proof. The proof is very similar to the fluid limit result from [Kelly and Williams, 2004], but with additional considerations for the admission control term. We shall use the following stochastic representation of the process $n^L(t)$, in terms of standard Poisson processes with a time scaling. Consider $\{E_r(t)\}$ and $\{S_r(t)\}$ to be a family of independent Poisson processes of intensities λ_r and μ_r respectively. Consider also the following processes:

$$\tau_r^L(t) = \int_0^t f_\epsilon(U_r'^L(\varphi^L(n^L(s))) - q_r^L(n^L(s)))\,ds,$$
$$T_r^L(t) = \int_0^t \varphi_r^L(n^L(s))\,ds$$

Here τ_r tracks the amount of time the admission condition is satisfied, and T_r tracks the amount of service provided to route r. The Markov chain evolution (5.6) of $n^L(t)$ starting at $n^L(0)$ can be written as:

$$n_r^L(t) = n_r^L(0) + E_r(L\tau_r^L(t)) - S_r(T_r^L(t)),$$

where the term $L\tau_r^L(t)$ comes from the fact that the arrival rate of n^L is $L\lambda_r$.

Define $\bar{T}_r^L = T_r^L/L$. Now for each r and $t \geqslant 0$:

$$\begin{aligned}\bar{T}_r^L(t) &= \frac{1}{L}\int_0^t \varphi_r^L(n^L(s))\,ds = \frac{1}{L}\int_0^t \varphi_r^L(L\bar{n}^L(s))\,ds = \frac{1}{L}\int_0^t L\varphi_r(\bar{n}^L(s))\,ds \\ &= \int_0^t \varphi_r(\bar{n}^L(s))\,ds,\end{aligned}$$

where we have used Lemma 5.3 for φ_r^L.

We also have:

$$
\begin{aligned}
\tau_r^L(t) &= \int_0^t f_\epsilon(U_r'^L(\varphi_r^L(n^L(s))) - q_r^L(n^L(s)))\,ds \\
&= \int_0^t f_\epsilon(U_r'^L(\varphi_r^L(L\bar{n}^L(s))) - q_r^L(L\bar{n}^L(s)))\,ds \\
&= \int_0^t f_\epsilon(U_r'^L(L\varphi_r(\bar{n}^L(s))) - q_r(\bar{n}^L(s)))\,ds \\
&= \int_0^t f_\epsilon(U_r'(\varphi_r(\bar{n}^L(s))) - q_r(\bar{n}^L(s)))\,ds.
\end{aligned}
$$

Therefore, the process $\bar{n}^L$ satisfies the following:

$$
\begin{aligned}
\bar{n}_r^L(t) &= \frac{n_r^L(0)}{L} + \frac{1}{L}E_r(L\tau_r^L(t)) - \frac{1}{L}S_r(L\bar{T}_r^L(t)), \\
\bar{T}_r^L(t) &= \int_0^t \varphi_r(\bar{n}^L(s))\,ds, \\
\tau_r^L(t) &= \int_0^t f_\epsilon(U_r'(\varphi_r(\bar{n}^L(s))) - q_r(\bar{n}^L(s)))\,ds.
\end{aligned}
$$

The conclusion now follows from the same arguments in [Kelly and Williams, 2004] applying the hypothesis for $n^L(0)$ and the functional law of large numbers for the processes E_r and S_r, namely, $\frac{1}{L}E_r(Lu) \to \lambda_r u$ and $\frac{1}{L}S_r(Lu) \to \mu_r u$. Note that the functions $\bar{T}_r$ are Lipschitz since they are the integral of a bounded function (φ_r is bounded by the maximum capacity in the network). Also, τ_r^L is a Lipschitz function because f_ε is bounded by 1.

So any limit point of $\bar{n}^L$ must satisfy:

$$
\bar{n} = \bar{n}(0) + \lambda_r \int_0^t f_\epsilon(U_r'(\varphi_r(\bar{n}(s))) - q_r(\bar{n}(s)))\,ds - \mu_r \int_0^t \varphi_r(\bar{n}(s))ds.
$$

Differentiating the above equation at any regular point of the limit, we arrive at:

$$
\dot{\bar{n}} = \lambda_r f_\epsilon(U_r'(\varphi_r(\bar{n}(t))) - q_r(\bar{n}(t))) - \mu_r \varphi_r(\bar{n}(s)),
$$

which is the desired result. □

5.4 Equilibrium and stability of the fluid limit

We now analyze the system in the fluid limit. We would like to characterize its equilibrium and prove stability results. From now on we focus on the fluid dynamics:

$$
\dot{n}_r = \lambda_r f_\epsilon(U_r'(\varphi_r) - q_r) - \mu_r \varphi_r, \tag{5.8}
$$

where $\varphi = \varphi(n)$ and $q = q(n)$ are the allocation maps defined through the TCP Congestion Control Problem 3.1, and n lies in the positive orthant.

We will assume that the user average loads $\rho_r = \lambda_r/\mu_r$ do not verify the network capacity constraints, i.e. the network without admission control is unstable. If it is stable, admission control is not needed and the fluid limit will converge to $n = 0$ by the same arguments of [de Veciana et al., 1999, Bonald and Massoulié, 2001].

The equilibrium condition for (5.8) is:

$$\varphi_r^* = \frac{\lambda_r}{\mu_r} f_\epsilon(U_r'(\varphi_r) - q_r) = \rho_r f_\epsilon(U_r'(\varphi_r) - q_r),$$

where ρ_r is user r average load.

Since f_ϵ is bounded above by 1, the only possibilities are:

$$\begin{aligned} &\varphi_r^* < \rho_r \quad \& \quad |U_r'(\varphi_r) - q_r| < \epsilon \quad \text{or} \\ &\varphi_r^* = \rho_r \quad \& \quad U_r'(\varphi_r) > q_r + \epsilon. \end{aligned}$$

As $\epsilon \to 0$ the above translate to:

$$\varphi_r^* < \rho_r \quad \& \quad U_r'(\varphi_r^*) = q_r^* \quad \text{or} \tag{5.9a}$$

$$\varphi_r^* = \rho_r \quad \& \quad U_r'(\varphi_r^*) > q_r^*. \tag{5.9b}$$

The interpretation of the above conditions is the following: either the equilibrium allocation for user r is less than its demand, and the system is in the boundary of the admission condition, or the user is allocated its full average demand and admission control is not applied.

We would like to relate this to the User Welfare Problem 4.1 defined before. Considered the following modification of Problem 4.1:

Problem 5.1 (Saturated User Welfare)**.**

$$\max \sum_r U_r(\varphi_r),$$

subject to:

$$\begin{aligned} &R\varphi \leqslant c, \\ &\varphi_r \leqslant \rho_r \quad \forall r. \end{aligned}$$

Problem 5.1 has the following interpretation: allocate resources to the different users according to their utility functions, but do not allocate a user more than its average demand. It amounts to saturating the users to a maximum possible demand, given by the value ρ_r.

We have the following:

Proposition 5.2. *The equilibrium points of of* (5.8) *as* $\epsilon \to 0$ *converge to the optimum of Problem 5.1.*

Proof. As $\epsilon \to 0$, the equilibrium of (5.7) verifies equations (5.9). We will show that these are equivalent to the KKT conditions of Problem 5.1.

The Lagrangian of Problem 5.1 is:

$$\mathscr{L}(\varphi, p, \eta) = \sum_r U_r(\varphi_r) - p^T(R\varphi - c) - \eta^T(\varphi - \rho),$$

where p as before are the link prices, η is the vector of multipliers for the additional constraint, and $\rho = (\rho_r)$. The KKT conditions are:

$$U_r'(\varphi_r^*) = q_r^* + \eta_r^* \quad \forall r, \tag{5.10a}$$

$$\eta_r^*(\varphi_r^* - \rho_r) = 0 \quad \forall r, \tag{5.10b}$$

$$p_l^*\left(\sum_r R_{lr}\varphi_r^* - c_l\right) = 0 \quad \forall l. \tag{5.10c}$$

Now consider an equilibrium allocation $\hat{\varphi}$ that verifies (5.9). The last condition above is always satisfied by $\hat{\varphi}$ since it is one of the KKT conditions of the TCP congestion control problem that defines $n \mapsto \varphi$. We can take $\hat{\eta}_r = U_r'(\hat{\varphi}_r) - \hat{q}_r$. With that choice, the triplet $(\hat{\varphi}, \hat{q}, \hat{\eta})$ satisfies the KKT conditions (5.10). The first condition is satisfied automatically. Moreover, condition (5.9a) implies $\hat{\eta} = 0$ and then the second KKT condition is valid in that case. Also, (5.9b) implies $\hat{\eta}_r > 0$ but at the same time $\hat{\varphi} = \rho_r$, so the second KKT condition is also valid in this case. □

Therefore, the equilibrium allocation under admission control in an overloaded network is a solution of Problem 5.1. Note that if the traffic demands are very large ($\rho_r \to \infty$), Problem 5.1 becomes the original User Welfare Problem 4.1, and admission control is imposing the desired notion of fairness. Moreover, if some users demand less than their fair share according to Problem 4.1, the resulting allocation *protects* them from the overload by allocating these users their mean demand, and sharing the rest according to the user utilities. In Section 5.6 we shall give a numerical example of this behavior.

We now turn our attention to the stability of the equilibrium. We shall prove a local stability result by using the same passivity arguments analyzed in Section 4.3. To do so, we will lift the time scale separation assumption, but nevertheless assume that the dynamics (5.8) is still a good approximation of network behavior. The result is the following:

Proposition 5.3. *Assume that utility based admission control can be modelled by the dynamics* (5.8)*, and the links perform the dual dynamics price adaptation* (2.11)*. Then the equilibrium of the system is locally asymptotically stable.*

Proof. The idea of the proof is to show that the system $-(q-q^*) \mapsto (\varphi-\varphi^*)$ linearized around the equilibrium is strictly passive, since the remaining half of the control loop $(\varphi-\varphi^*) \mapsto (q-q^*)$ is passive, as shown by [Wen and Arcak, 2003], and discussed in Section 4.3.

To do so, observe that in the single path case, the user part of the system is diagonal. We focus on one user and drop the subscript i accordingly. The Laplace domain version of (5.8) around equilibrium is:

$$\begin{aligned} s\delta n &= \lambda a(-b\delta\varphi - \delta q) - \mu\delta\varphi \\ &= -(ab\lambda+\mu)\delta\varphi - \lambda a\delta q \end{aligned} \tag{5.11}$$

where $a = f'_\epsilon(U'(\varphi^*) - q^*) \geqslant 0$ and $b = -U''(\varphi^*) > 0$ are constants. Note also that:

$$\delta\varphi = n^*\delta x + x^*\delta n = -dn^*\delta q + x^*\delta n \tag{5.12}$$

where we have used that $\delta x = -d\delta q$ with $d = -f'_{TCP}(q^*) > 0$. Combining (5.11) and (5.12) we obtain the transfer function for the system:

$$\frac{\delta\varphi}{\delta q} = -\frac{n^*ds + \lambda a x^*}{s + x^*(ab\lambda+\mu)} = -H(s)$$

For simplicity we focus on the case where $a > 0$, which corresponds to the situation where admission control is operating, $H(s)$ is strictly positive real and thus strictly passive (c.f. Theorem A.5). Since each block in the diagonal is strictly passive we can find a suitable storage function for the complete system by adding the storage function of each user subsystem, and we can construct a strictly decreasing Lyapunov function for the feedback loop. This shows that the equilibrium is locally asymptotically stable.

The case where $a = 0$ (i.e. when the equilibrium fully satisfies user demand) can be treated in a similar way by using more detailed passivity results for the network side (c.f. [Wen and Arcak, 2003, Theorem 2]). □

As in the case of cooperative control studied in Section 4.3, global stability results are harder to obtain.

5.5 Admission control in the multi-path case

We now analyze how to extend the results on admission control to a situation where the user opens connections on several paths, and gets an utility for the aggregate. We assume that connection arrivals on each path are independent, following a Poisson process of intensity λ_r, and with exponentially distributed workloads of mean $1/\mu_r$. Thus, the lower layers of the network allocate resources as in the single path situation, and in particular each route has an average load $\rho_r = \lambda_r/\mu_r$. However, we shall control the number of connections in each path taking into account the *aggregate rate* each user perceives through all its paths.

The logical extension of the admission control rule (5.1) to the case where users may open connections along multiple routes is the following:

$$\begin{aligned} &\text{If } U_i'(\varphi^i) > q_r \rightarrow \text{ admit connection on route } r. \\ &\text{If } U_i'(\varphi^i) \leqslant q_r \rightarrow \text{ discard connection on route } r. \end{aligned} \tag{5.13}$$

where $\varphi^i = \sum_{r \in i} \varphi_r$, as before.

The analysis of the above policy is similar to the single path case. Performing a fluid limit scaling of the underlying Markov process, the fluid limit dynamics of the system become:

$$\dot{n}_r = \lambda_r f_\epsilon(U_i'(\varphi^i) - q_r) - \mu_r \varphi_r. \tag{5.14}$$

And we have the analogue of Proposition 5.2. Consider the following problem:

Problem 5.2 (Saturated User Welfare (multi-path))**.**

$$\max \sum_r U_r(\varphi^i)$$

subject to:

$$\begin{aligned} R\varphi &\leqslant c, \\ \varphi_r &\leqslant \rho_r \quad \forall r \end{aligned}$$

We have the following:

Proposition 5.4. *The equilibrium points of* (5.14) *converge, as* $\epsilon \to 0$*, to the optimum of Problem 5.2.*

Stability results of this equilibrium, however, proved harder to obtain. We shall explore the behavior of this policy through packet level simulation in the following section.

5.6 Simulation examples

In this section we discuss practical implementation issues and explore the performance of the policies developed through ns2 simulations. We present two scenarios of admission control, and show how the desired level of fairness is attained.

We implemented in ns2 several mechanisms: first we implemented a “User” class that feeds the network with a Poisson process of arriving connections. Moreover, each connection size is an independent exponential random variable, as in the model of Section 5.1. Each connection is established in an independent way by using a new instance of TCP-Newreno.

At the network edge, the aggregate rate of each user is measured, as well as the loss probability, which is the route price. The loss probability is estimated from the number of retransmitted packets, and averaged between all the connections sharing the same route. This is done in this case by sniffing the connections, and approach that we understand can be difficult on real networks. However, recent work on explicit congestion notification (c.f. Re-ECN [Briscoe, 2008]) can also be used to communicate congestion prices to the edge router, and can be a possible solution to this problem.

Admission control is performed by the rule (5.13), where the utility function can be chosen from the $\alpha-$family.

5.6.1 Single path scenario

Our first example consists of a linear parking lot network, with three single path users. We choose the capacities as in Figure 5.2, and admission control is performed with $\alpha = 5$, to emulate the max-min fair allocation, which for this network is given by $\varphi^{*,1} = 5Mbps$, $\varphi^{*,2} = \varphi^{*,3} = 3Mbps$.

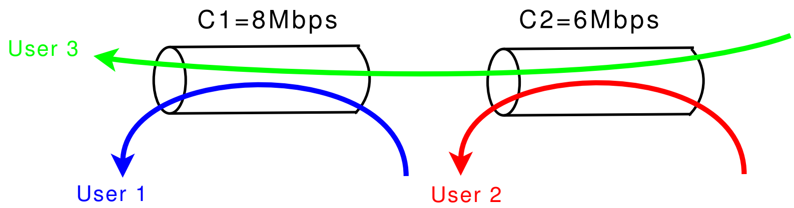

FIGURE 5.2: PARKING LOT TOPOLOGY.

In our first simulation, the user loads are such that the network is overloaded, with each user load greater that its fair share. The results show that the users are admitted $n_1 \approx 5$, $n_2 \approx 8$ and $n_3 \approx 15$ simultaneous connections in equilibrium, with total rates shown in Figure 5.3. There we see that the max-min allocation is approximately achieved. Observe that, despite having the same equilibrium rate as user 2, user 3 is allowed more connections into the network because its RTT is higher, and thus its connections slower.

We simulated a second situation in which we reduced the average load of user 2 to below its fair share according to Problem 4.1. In this case, $\rho^2 = 1Mbps$ and is therefore saturated in the sense of Problem 5.1. The new average rates for this situation are shown in figure 5.4. Note that admission control protects user 2 by allowing its share of $1Mbps$ into the network. The remaining capacity is allocated as in Problem 5.1, whose optimum is $\varphi^{*,1} = \varphi^{*,3} = 4Mbps$, and $\varphi^{*,2} = 1Mbps$.

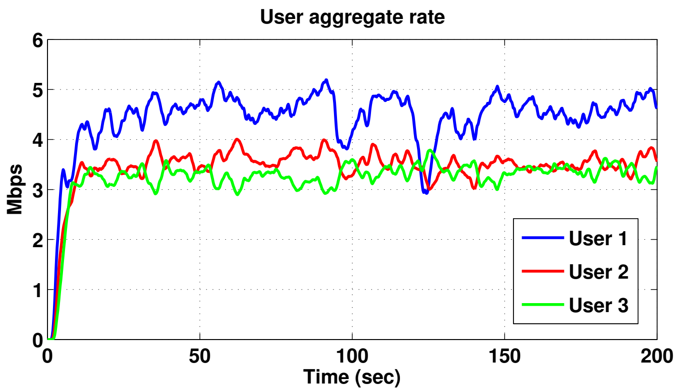

FIGURE 5.3: ADMISSION CONTROL RESULTS FOR THE OVERLOAD CASE.

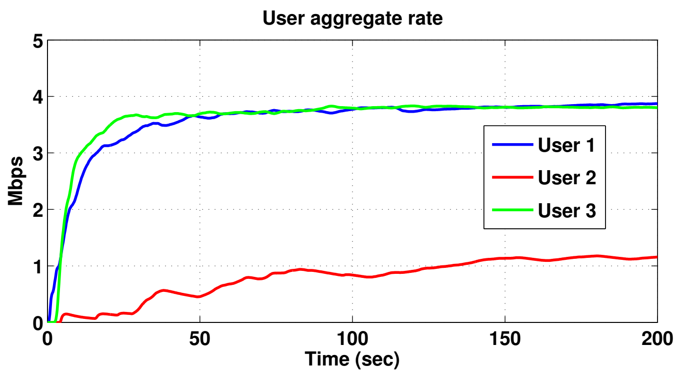

FIGURE 5.4: ADMISSION CONTROL RESULTS WHEN USER 2 IS BELOW ITS FAIR SHARE.

5.6.2 Multi path scenario

In this case, we retake the setting of a p2p downloading system analyzed in Section 4.5. The topology is the same, shown again in Figure 5.5, but now we rely on admission control to impose fairness, instead of user cooperation and control. As before, users generate connections as a Poisson process in each route and the loads are chosen such that the network is in overload. Admission control is performed at the edge of the network, taking into account the total rate of each user, with utilities chosen to obtain proportional fairness.

The results are plotted in Figure 5.6, where we can see the number of ongoing connections and the resulting user aggregate rates. Note that the number of connections is much noisier that in the cooperative example, due to the inherent randomness of connection arrival and departures. However, admission control manages to drive the average user rates in equilibrium around the fair share of the User Welfare Problem 4.1, which is again in this case $\varphi^{1,*} = \varphi^{2,*} = 10Mbps$. Note also that in the absence of admission control, the number of connections would grow without bounds.

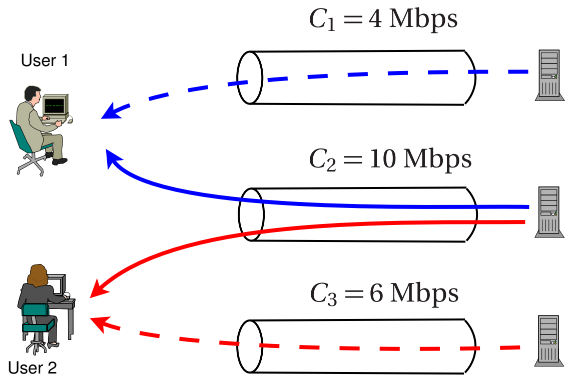

FIGURE 5.5: DOWNLINK P2P EXAMPLE TOPOLOGY.

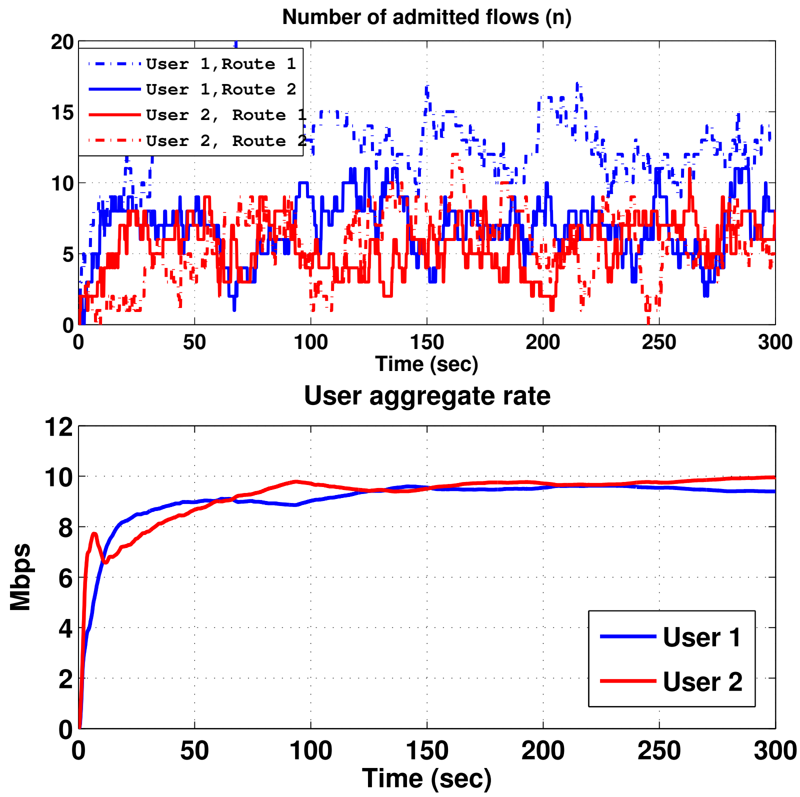

FIGURE 5.6: DOWNLINK P2P EXAMPLE RESULTS: NUMBER OF CONNECTIONS (ABOVE) AND USER AGGREGATE RATES (BELOW.

6
Connection level routing

The analysis of the preceding chapter assumes that each user establishes connections through some set of predefined routes, possibly with multiple destinations. The user manages simultaneously several connections over these routes and derives a utility from the aggregate rate. Moreover, the user has an independent arrival rate λ_r for each route.

In this chapter we focus on a slightly different situation: here, each user has a set of routes available to communicate with a given destination in the network. These routes are indifferent for the user, because all of them serve the same purpose. Each user brings connections into the network, and at each connection arrival, the user or the edge router may decide over which route send the data.

This is a typical instance of the multi-path load balancing problem, but at the connection level timescale. The problem of balancing the load between several possible routes has been analyzed before in several ways. In [Paganini and Mallada, 2009], the combined problem of congestion control and multi-path routing was studied, where the end point entities and the network share responsibility on the rate and price adaptation mechanisms in order to find a suitable optimum. However, in that work, the number of origin-destination pairs is assumed fixed, and with only one connection. From the connection level perspective, in [Han et al., 2006] an algorithm called Multi-path TCP was introduced, which assumes that each connection is served by a congestion control protocol that uses *all* all the possible paths simultaneously. This algorithm enables to make an efficient use of the available capacity, but it has several implementation problems, due to the fact that different packets of the same flow travel through multiple routes, causing reordering problems, so extra control traffic is needed for the system to work. Finally, an interesting class of load balancing techniques was studied in [Jonckheere, 2006], called *balanced routing*, which finds an optimal random splitting of the traffic using the state of the network. However, its applicability is reduced because it assumes that the network resource allocation belongs to the class of *balanced fairness*, which is not in use in real networks since it is difficult to decentralize.

What we propose is to use the route prices as a suitable indicator of congestion over the routes, and use them to derive a simple decentralized algorithm to assign connections to routes. We will rely then on standard TCP congestion control algorithms to control the rate of each connection, maintaining each individual connection along a single path.

This chapter is organized as follows: in Section 6.1 we discuss a stochastic model and its fluid version for routing policies in a network. We then show in Section 6.2 a necessary condition for stability, generalizing the stability condition discussed in Section 3.3 to the multi-path case. In Section 6.3 we propose a decentralized routing policy based on the route prices and analyze its stability in some topologies. Finally, in Section 6.4 we combine the routing policy with the admission control rules introduced in Chapter 5 to give a combined utility based routing and admission control policy for the system.

6.1 Connection level routing policies

We will make a model of the system based on the stochastic model for connections described in Section 3.3. We assume that users, indexed by i, have multiple routes r available to serve their jobs. Users generate incoming connections as a Poisson process of intensity λ_i, and exponential file sizes with mean $1/\mu_i$. Thus $\rho^i = \lambda_i/\mu_i$ represents the *user* average load. Note that here we do not distinguish between the routes, since each user may be served by a set $\mathscr{R}_i$ of possible routes.

As for congestion control, we assume that the TCP layer behaves as described in 3.1, that is, for each number of ongoing connections, the TCP layer allocates resources by optimizing Problem 3.1. The user or the network may now choose, at the start of the connection, to which route send the data from the set $\mathscr{R}_i$, but note that each connection behaves independently after that. In particular each connection follows a single specified path throughout its service. Nevertheless, by appropriately choosing the route, the load may be distributed across multiple paths.

We formalize a routing policy in the following way: given the current state of the network, characterized by the vector $n = (n_r)$ of ongoing connections per route, a routing policy is a selection $r \in \mathscr{R}_i$ for a new connection. We denote by $A_{ir} \subset \mathbb{N}^N$ the set of network states such that connections arriving from user i are routed through route $r \in \mathscr{R}_i$. If the same physical route is possible for many users, we duplicate its index r accordingly, and N is the total number of routes.

The only general requirement for the routing policy is that the sets A_{ir} are a partition of the space for each i, i.e.:

$$\sum_{r \in \mathscr{R}_i} \mathbf{1}_{A_{ir}} \equiv 1 \quad \forall i. \tag{6.1}$$

Let us make a fluid model of this system, under the same ideas of Chapter 5. In a fluid limit, the dynamics of the number of connections is given by:

$$\dot{n}_r = \left(\lambda_i \mathbf{1}_{A_{ir}} - \mu_i \varphi_r\right)^+_{n_r}. \tag{6.2}$$

Here, $\varphi_r = \varphi_r(n)$ as before denotes the total rate assigned to the flows on route r depending on network state. The saturation $(\cdot)^+_{n_r}$ is needed in this case because some routes may not be used, and thus the number of flows must remain at 0.

Remark 6.1. Note that we could have also considered more general routing policies, in which each routing decision is assigned a probability $p_{ir}(n)$ for each network state. The routing policy constraint (6.1) in that case will be the same. However, in the following we will only focus on deterministic routing policies.

Note also that the sets A_{ir} may be quite general. However, for practical implementation, it is necessary that the routing policy is decentralized, i.e. the decision of routing a flow of user i over route r should be performed on local information. We defer this discussion to Section 6.4 and focus now on necessary conditions for stability of the system.

6.2 A necessary condition for stability

Our goal is to characterize the stability region of a routing policy, with dynamics given by (6.2). More formally, we would like to know for which values of ρ^i the fluid model goes to zero in finite time. We recall that finite time convergence of the fluid model is related (c.f. [Robert, 2003, Chapter 9]) to the stochastic stability of the corresponding Markov chain models.

We will first derive a necessary condition for stability, which generalizes the stability condition of (3.5) to the routing case.

For this purpose, introduce for each user i the simplex of possible traffic fractions among available routes:

$$\Delta^i = \left\{ \alpha^i = (\alpha^i_r)_{r \in \mathscr{R}_i} : \alpha^i_r \geqslant 0, \sum_{r \in \mathscr{R}_i} \alpha^i_r = 1 \right\}.$$

The following is the main result of this section:

Theorem 6.1. *A necessary condition for the existence of a policy* $\{A_{ir}\}$ *that stabilizes the dynamics* (6.2) *is the existence for each user of a split* $\alpha^i \in \Delta^i$ *such that:*

$$\sum_l R_{lr}\alpha_r^i \rho^i \leqslant c_l \quad \forall l. \tag{6.3}$$

Remark 6.2. Note that condition (6.3) is the non-strict version of (3.5) for the splitted traffic loads $\rho_r = \alpha_r^i \rho_i$. Thus, equation (6.3) means that for a routing policy to exist, it is necessary that the network is "stabilizable", in the sense that there is a partition of the user loads such that the underlying single path network is stable. Of course, if each user has only one possibility, then $\Delta^i = \{1\}$ and we recover the single path stability condition.

Condition (6.3) was obtained in [Han et al., 2006] for stochastic stability in the case of Multi-path TCP. In that case, however, the TCP layer must be modified to make full simultaneous use of the available routes. Here each route remains single-path, with standard TCP congestion control, and the routing policy is used to achieve the same stability region, whenever possible.

Proof of Theorem 6.1. Consider the convex and compact subset of $\mathbb{R}^L$, with L the total number of links:

$$\mathscr{S} = \left\{ z \in \mathbb{R}^L : z_l = \sum_r R_{lr}\alpha_r^i \rho^i - c_l, \alpha^i \in \Delta^i \right\}$$

The set $\mathscr{S}$ represents the excess of traffic in each link for each possible split. If (6.3) is not feasible for a given load vector ρ^i, then the set $\mathscr{S}$ is disjoint with the closed negative orthant $\mathbb{R}^L_-$. By convexity, there exists a strictly separating hyperplane (c.f. [Boyd and Vanderberghe, 2004, Section 2.5]), i.e. a fixed $\hat{p} \in \mathbb{R}^L$ with $\hat{p} \neq 0$ and such that:

$$\hat{p}^T z \geqslant a + \epsilon \quad \forall z \in \mathscr{S},$$
$$\hat{p}^T z \leqslant a \quad \forall z \in \mathbb{R}^L_-.$$

The second condition implies in particular that $\hat{p}$ has nonnegative entries, since if $\hat{p}_l < 0$, taking $z_l \to -\infty$ we have $\hat{p}_l z_l \to +\infty$ and therefore the inequality is violated for any a. Also, since $z = 0 \in \mathbb{R}^L_-$, we have that $a \geqslant 0$.

Define now $\hat{q} = R^T \hat{p}$ and consider the following state function:

$$V(n) = \sum_i \frac{1}{\mu_i} \sum_{r \in \mathscr{R}_i} \hat{q}_r n_r.$$

Note that $\hat{q}_r \geqslant 0$ and $\hat{q} \neq 0$.

Differentiating V along the trajectories of (6.2) we get:

$$\begin{aligned}\dot{V} &= \sum_i \frac{1}{\mu_i} \sum_{r \in \mathscr{R}_i} \hat{q}_r \dot{n}_r = \sum_i \frac{1}{\mu_i} \sum_{r \in \mathscr{R}_i} \hat{q}_r [\lambda_i \mathbf{1}_{A_{ir}} - \mu_i \varphi_r]^+_{n_r} \\ &\geqslant \sum_i \frac{1}{\mu_i} \sum_{r \in \mathscr{R}_i} \hat{q}_r \left(\lambda_i \mathbf{1}_{A_{ir}} - \mu_i \varphi_r\right),\end{aligned}$$

where in the last step we used the fact that the term inside the projection is negative whenever the projection is active. Reordering the terms we arrive at:

$$\begin{aligned}\dot{V} &\geqslant \sum_i \sum_{r\in\mathscr{R}_i} \hat{q}_r\left(\rho^i \mathbf{1}_{A_{ir}} - \varphi_r\right)\\ &= \sum_i \sum_{r\in\mathscr{R}_i} \sum_l R_{lr}\hat{p}_l \rho^i \mathbf{1}_{A_{ir}} - \sum_i \sum_{r\in\mathscr{R}_i} \sum_l R_{lr}\hat{p}_l \varphi_r\\ &\geqslant \sum_i \sum_{r\in\mathscr{R}_i} \sum_l \hat{p}_l R_{lr}\rho^i \mathbf{1}_{A_{ir}} - \sum_l \hat{p}_l c_l.\end{aligned}$$

In the last step, we used the fact that $\sum_r R_{lr}\varphi_r \leqslant c_l$ due to the resource allocation being feasible.

Regrouping the terms we arrive at:

$$\dot{V} \geqslant \sum_l \hat{p}_l \left(\sum_i \sum_{r\in\mathscr{R}_i} R_{lr}\rho^i \mathbf{1}_{A_{ir}} - c_l \right) \geqslant a + \epsilon > 0,$$

since $z_l = \sum_r R_{lr}\rho^i \mathbf{1}_{A_{ir}} - c_l \in \mathscr{S}$ by the definition of routing policy.

Therefore, V is strictly increasing along the trajectories, and being a linear function of the state, if the condition (6.3) is not met the number of connections grows without bounds. Thus (6.3) is necessary for stability. □

Remark 6.3. Note that the above Theorem remains valid if we change the policy $\mathbf{1}_{A_{ir}}$ by a random splitting policy, possibly dependent on the state $p_r(n)$, since $p_r(n)$ must also be in the set Δ^i for all n.

We therefore have shown that if traffic cannot be split among the routes such that each link is below its capacity on average, then the system cannot be stabilized under any routing policy.

The analogue to the sufficient stability condition (3.5) in this case is:

$$\forall i, \exists \alpha^i \in \Delta^i : \sum_l R_{lr}\alpha^i_r \rho^i < c_l \quad \forall l, \tag{6.4}$$

which is the strict version of (6.3). The following proposition is direct from the single-path stability Theorem 3.2:

Proposition 6.1. *If* (6.4) *holds, then there exists a routing policy that stabilizes the system.*

Proof. If such a choice of α^i_r exists, then the random splitting policy that sends an incoming connection on route r with probability α^i_r stabilizes the system. This is because the system is equivalent to a single path process with arrival rates $\lambda_i \alpha^i_r$ due to the Poisson thinning property, and condition (6.4) is the stability condition of Theorem 3.2 in that case. □

The above shows that the stability region of this system is characterized completely. However, the random splitting policy mentioned in the proof of Proposition 6.1 is not useful in a network environment, since it is not decentralized. Each user must know in advance the average loads of the whole system in order to select the weights α_r^i accordingly, in order to fulfill (6.4).

We would like to introduce a suitable feedback policy that takes information from the network to decide on which route send an incoming connection. We propose such a policy in the next section, and analyze its stability under condition (6.4).

6.3 A decentralized routing policy

In a multi-path setting in which each user may choose among a set of routes, it is natural to try to balance the loads by using the network congestion prices measured on each route. A simple feedback policy for routing is, at each arrival, choose the route with the cheapest price. In our previous notation this amounts to take:

$$A_{ir} = \left\{ n : q_r(n) = \min_{r' \in \mathscr{R}_i} q_{r'}(n) \right\}. \tag{6.5}$$

Implicit in the above equation is some rule for breaking ties when there are multiple routes with minimum price. Since congestion price is inversely related with connection rate, this is equivalent to routing new flows to the path with the best current rate for individual connections. Note also that, in the case of equal parallel links with similar TCP utilities, it can be seen as a generalization of the Joint the Shortest Queue policy for processor sharing queues.

We shall investigate the stability of this policy under condition (6.4). We need the following Proposition:

Proposition 6.2. *Given $n = (n_r) \in \mathbb{R}_+^N$, let $\varphi_r(n)$ and $q_r(n)$ be the corresponding rate allocation and route prices from the TCP Congestion Control Problem 3.1. Choose also α^i that satisfies* (6.4)*. Then there exists $\delta > 0$ such that:*

$$\sum_r q_r(n)\alpha_r^i \rho^i - \sum_{r:n_r>0} q_r(n)\varphi_r(n) \leqslant -\delta \sum_r q_r(n). \tag{6.6}$$

To prove the above Proposition, we must deal with the quantities $q_r(n)\varphi_r(n)$ when $n_r \to 0$. We have the following:

Lemma 6.1. *Assume that n is such that $n_r = 0$. Consider $n^\epsilon = n + \epsilon e_r$ Then $q_r(n^\epsilon)\varphi_r(n^\epsilon) \to 0$ when $\epsilon \downarrow 0$.*

Proof. From the derivation of Theorem 3.1 we know that for $n > 0$:

$$\frac{\partial q}{\partial n} = R^T (RDR)^{-1} RF = BF,$$

with $D, F > 0$ and diagonal. Observe that $B \geqslant 0$ and therefore $b_{rr} \geqslant 0$. We conclude that in the region $n > 0$:

$$\frac{\partial q_r}{\partial n_r} \geqslant 0$$

and thus $q_r(n^\epsilon)$ decreases as $\epsilon \downarrow 0$, therefore it has a limit q_{r0}. If $q_{r0} = 0$ we are done since $\varphi_r(n)$ is bounded by the maximum link capacity. If $q_{r0} > 0$ then $x_r(n^\epsilon) = f_{TCPr}(q_r(n^\epsilon))$ remains bounded and therefore $\varphi_r(n^\epsilon) = n_r^\epsilon x_r(n^\epsilon) \to 0$. $\square$

Proof of Proposition 6.2. Since the inequality in (6.4) is strict, we can choose $\delta > 0$ such that:

$$\psi_r = \alpha_r^i \rho^i + \delta \quad \forall r$$

satisfies the link capacity constraints $R\psi \leqslant c$.

Consider now $n > 0$. Since $\varphi(n)$ is the optimal allocation, and ψ is another feasible allocation for the convex Problem 3.1, we have that the inner product:

$$\nabla_\varphi \left(\sum_r n_r U_{TCPr}(\varphi_r / n_r) \right) (\psi - \varphi) \leqslant 0.$$

In fact, if it were positive, we can improve the solution φ by moving it in the direction of ψ.

Applying the above condition to $\psi_r = \alpha_r^i \rho^i + \delta$ and noting that:

$$\frac{\partial \left(\sum_r n_r U_{TCPr}(\varphi_r / n_r) \right)}{\partial \varphi_r} = U'_{TCPr}\left(\frac{\varphi_r}{n_r} \right) = q_r(n),$$

we conclude that:

$$\sum_r q_r(n) \left(\alpha_r^i \rho^i + \delta - \varphi_r(n) \right) \leqslant 0,$$

which can be rewritten as:

$$\sum_r q_r(n) \alpha_r^i \rho^i - \sum_r q_r(n) \varphi_r(n) \leqslant -\delta \sum_r q_r(n)$$

which proves the result for $n > 0$ componentwise.

If now n is such that $n_r = 0$ for some r, we can take limits and invoke Lemma 6.1 to get the result. $\square$

The previous bound is similar to the one used in [Bonald and Massoulié, 2001] to prove stability in the single path scenario, but with the gradient evaluated at the optimum, instead of another feasible allocation. We now apply this bound to obtain the following characterization of the routing policy:

Theorem 6.2. *Suppose* (6.4) *holds. Then under the dynamics* (6.2) *with the routing policy given by* (6.5) *we have:*

$$\sum_r \frac{q_r \dot{n}_r}{\mu_i} \leqslant -\delta \sum_r q_r \tag{6.7}$$

Proof. We have that:

$$\sum_r \frac{q_r \dot{n}_r}{\mu_i} = \sum_r q_r \left[\rho^i \mathbf{1}_{A_{ir}} - \varphi_r\right]^+_{n_r} \leqslant \sum_r q_r \rho^i \mathbf{1}_{A_{ir}} - \sum_{r:n_r>0} q_r \varphi_r,$$

where the inequality is trivial if the projection is not active. If the projection is active, $n_r = 0$ and thus $q_r \varphi_r = 0$ so the corresponding negative term vanishes.

Regrouping the above in each user we get:

$$\begin{aligned}
\sum_r \frac{q_r \dot{n}_r}{\mu_i} &\leqslant \sum_i \rho^i \sum_{r\in\mathcal{R}_i} q_r \mathbf{1}_{A_{ir}} - \sum_{r:n_r>0} q_r \varphi_r \\
&= \sum_i \rho^i \min_{r\in\mathcal{R}_i} q_r - \sum_{r:n_r>0} q_r \varphi_r \\
&\leqslant \sum_i \rho^i \sum_{r\in\mathcal{R}_i} q_r \alpha^i_r - \sum_{r:n_r>0} q_r \varphi_r,
\end{aligned}$$

where we have used the definition of the routing policy and the fact that the minimum price can be upper bounded by any convex combination of the prices.

Applying Proposition 6.2 we complete the proof. □

It remains to see if we can use the inequality (6.7) to establish asymptotic stability of the fluid dynamics through a Lyapunov argument. Note that, although the Lyapunov function used in the proof of Theorem 6.1 has a similar shape, and now the inequality has been reversed, there is an important difference: the q_r factor of (6.7) is a function of the state. This implies that the left hand side of (6.7) may not be integrated easily to get a Lyapunov function for the state space whose derivative along the trajectories yields the desired result.

We now focus on a special case where we can give an affirmative answer. Assume that the network is composed by a set of parallel bottleneck links as depicted in Figure 6.1. Each user in this network has a set of routes established in any subset of the links. Moreover, assume that all users have identical $\alpha-$fair utilities denoted by $U(\cdot)$ and file sizes are equal for each user, so we can take without loss of generality $\mu_i = 1$.

In such a network, the resource allocation of Problem 3.1 can be explicitly computed as a function of the current number of flows n_r. In particular, all flows through bottleneck l face the same congestion price $q_r = p_l$, and as they have the same utility, they will get the same rate, given by:

$$x_r(n) = \frac{c_l}{\sum_{r'\in l} n_{r'}}.$$

By using that $q_r(n) = U'(x_r(n))$ we have:

$$q_r(n) = x_r(n)^{-\alpha} = \left(\frac{\sum_{r'\in l} n_{r'}}{c_l}\right)^\alpha,$$

with l such that $r \in l$.

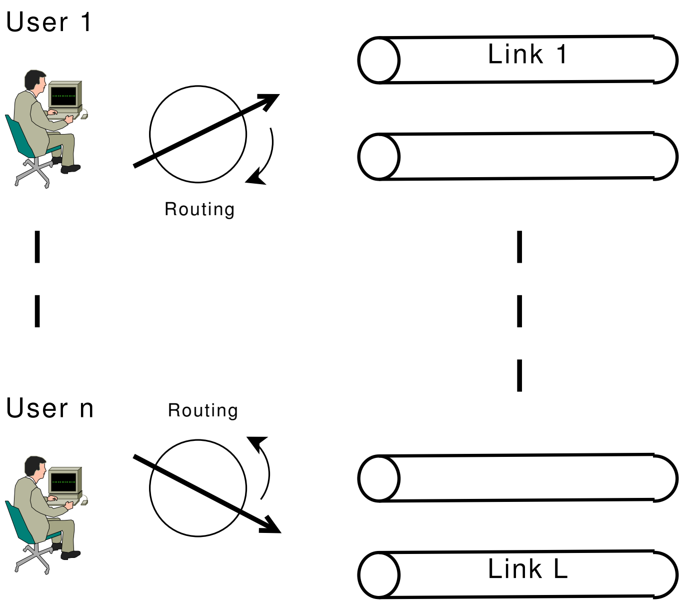

FIGURE 6.1: PARALLEL LINKS NETWORK WITH ROUTING COVERED IN THE ANALYSIS.

Since $q_r = p_l$ if $r \in l$ the link prices should have the form:

$$p_l(n) = \left(\frac{\sum_{r:r\in l} n_r}{c_l}\right)^{\alpha}.$$

We now state the stability result for this type of network:

Proposition 6.3. *For the network of parallel links under consideration, let the system be given by* (6.2) *with the routing policy* (6.5)*. Under the stability condition* (6.4)*, the state n converges to* 0 *in finite time.*

Proof. Consider the candidate Lyapunov function:

$$V(n) = \frac{1}{\alpha+1}\sum_l \frac{1}{c_l^{\alpha}}\left(\sum_{r:r\in l} n_r\right)^{\alpha+1}. \tag{6.8}$$

The above function is continuous and non-negative in the state space, radially unbounded and is only 0 at the equilibrium $n = 0$. Its derivative along the trajectories verifies:

$$\dot{V} = \sum_l \frac{\left(\sum_{r:r\in l} n_r\right)^{\alpha}}{c_l^{\alpha}}\left(\sum_{r:r\in l} \dot{n}_r\right) = \sum_l p_l\left(\sum_{r:r\in l} \dot{n}_r\right) = \sum_r q_r \dot{n}_r. \tag{6.9}$$

Invoking Theorem 6.2 we conclude that $\dot{V} \leqslant 0$ in the state space, and it is only 0 when all prices are 0 which can only happen at the origin. This implies asymptotic stability of the fluid dynamics.

To obtain finite time convergence note that V verifies:

$$V(n) = \frac{1}{\alpha+1}\sum_l p_l^{\frac{\alpha+1}{\alpha}} c_l \leqslant (K \max_l p_l)^{\frac{\alpha+1}{\alpha}},$$

where $K > 0$ is an appropriate constant.

We can obtain the following bound:

$$V^{\frac{\alpha}{\alpha+1}} \leqslant K \max_l p_l \leqslant K \sum_l p_l \leqslant K \sum_r q_r,$$

and thus:

$$-\delta \sum_r q_r \leqslant -\frac{\delta}{K} V^{\frac{\alpha}{\alpha+1}}.$$

Combining this with the result from Theorem 6.2 we get:

$$\dot{V} \leqslant -\frac{\delta}{K} V^{\frac{\alpha}{\alpha+1}}.$$

Integrating the above inequality yields:

$$V(t)^{\frac{1}{1+\alpha}} \leqslant V(0)^{\frac{1}{1+\alpha}} - \frac{\delta}{K(1+\alpha)} t,$$

and we conclude that V, and therefore n, reach 0 in finite time, proportional to $V(0)^{\frac{1}{1+\alpha}}$. □

Remark 6.4. Note that in the above proof the time of convergence to 0 starting from an initial condition $n(0)$ scales linearly with n since $V(0)$ is of order $n(0)^{\alpha+1}$. We can therefore conclude that the stochastic system will also be stable, via the stability of the fluid limit.

6.4 Combining admission control and routing

To end this chapter, let us analyze the possibility of extending the admission control rules derived in Chapter 5 to the connection routing setting.

Recall that, in order to perform admission control, we associate with each user an utility function $U_i(\varphi^i)$ with φ^i being the total rate the user gets from the network. Admission control over a route was performed by comparing the marginal utility with the route price.

In the new setting, the end user may choose among several routes, and thus the natural way to merge the results of 5 with the connection level routing is the following combined law:

Admit new connection if $\min_{r\in\mathscr{R}_i} q_r < U_i'(\varphi^i)$

If admitted: route connection through the cheapest path

The combined fluid dynamics for admission control and routing can then be written as:

$$\dot{n}_r = \left(\lambda_i \mathbf{1}_{A_{ir}} \mathbf{1}_{\{U_i'(\varphi^i) > q_r\}} - \mu_i \varphi_r\right)^+_{n_r}. \tag{6.10}$$

The above dynamics should converge to 0 whenever the stability condition (6.4) is met. The interest of admission control is in the case where (6.3) is violated, i.e. the network is overloaded. In that case, we have the analogous of Proposition 5.4.

Proposition 6.4. *The equilibrium points of* (6.10) *are optimal solutions of the following problem:*

$$\max_{\varphi_r} \sum_i U_i(\varphi^i)$$

subject to:

$$\begin{aligned} R\varphi &\leqslant c, \\ \varphi^i &\leqslant \rho^i \quad \textit{for each } i. \end{aligned}$$

The proof is similar to 5.4. The above optimization problem is similar to the Saturated User Welfare Problem 5.1 but the constraint $\varphi^i \leqslant \rho^i$ is imposed on the aggregate rate of each user and its total average load, instead of the average load per route. Stability conditions however, proved harder to obtain in this case, and the problem is open for future work.

7
Conclusions of Part I

In this part of the thesis, we proposed a new paradigm for resource allocation in networks, which intends to bridge the gap between classical NUM applied to congestion control and a user centric perspective. New notions of fairness appear, as user utilities are evaluated in aggregates of traffic, which can model different interesting situations.

We showed how the control of the number of connections can be used to impose these new notions of fairness, and how the users can cooperate in order to drive the network to a fair equilibrium.

Moreover, we showed how admission control and routing based on typical congestion prices can be used to protect the network in overload, and simultaneously impose fairness between its users.

Finally, we showed practical implementations of the mechanisms derived in our work, and simulations based on these implementations show that the proposals accomplish their goals.

Several lines of work remain open, both theoretical and technical. For instance, generalizing the stability results for multipath admission control, and the stability region of the routing policy proposed are two important theoretical questions. In practical terms, we plan to explore new network implementations based on current congestion notification protocols, that will help make these decentralized admission control mechanisms scalable to large networks

Part II

Resource allocation and connection level models for multirate wireless networks

8

Resource allocation in multirate wireless environments

Wireless local area networks, in particular those based on the IEEE 802.11 standard (commonly known as WiFi) [IEEE 802.11, 2007], are becoming increasingly prevalent to deliver Internet access. More and more user equipments have WiFi capability, and there is an increasing demand for bandwidth and communication speed in these networks.

The IEEE 802.11 standard has been very successful, standardizing the behavior of the physical (PHY) and medium access control (MAC) layers of communication. These layers define how to send the data over the radio channel, and more importantly, how to share the common access medium between the different users in a decentralized way. The approach, based on random access protocols [Bertsekas and Gallager, 1991], proved successful in wired networks such as Ethernet, and is present again in 802.11.

One important feature these networks introduce is the capability of using different transmission rates in the physical layer, in order to communicate with stations under different radio channel conditions. This enables a station to connect to its access point (AP) even in bad channel conditions, by reducing its modulation speed and transmitting data more slowly.

A crucial performance issue however remains to be analyzed: how this feature interacts with higher layer protocols such as TCP, and how this impacts the resource allocation among competing users. The 802.11 layer was designed independently of the upper layers of the network, and thus a cross-layer approach for the analysis is needed to examine the efficiency and fairness of the whole system.

The main purpose of this part of the Thesis is thus to analyze the efficiency and fairness of current protocols under the Network Utility Maximization perspective, identifying possible problems, and analyzing some possible solutions. We begin in this chapter by studying the resource allocation provided by TCP over a simplified multirate layer.

The chapter is organized as follows: In Section 8.1 we discuss the previous work on the subject, and then in Section 8.2 we present our model for TCP resource allocation in multirate networks and derive some conclusions.

8.1 Background and previous work

Wireless local area networks (WLANs) are nowadays present in most networking deployments around the world. These hotspots provide Internet connectivity to nearby users via the use of the radio frequency (RF) medium. The main standard used nowadays is the IEEE 802.11 [IEEE 802.11, 2007], which is commonly known as WiFi.

Among other things, the standard defines the way multiple users access the shared medium through the use of Medium Access Control (MAC) protocols. In particular, the IEEE 802.11 standard defines the Distributed Coordination Function (DCF) MAC protocol, which is the most widely deployed of the possible choices in the standard. This protocol defines a *random access* policy, based on Carrier Sense Multiple Access with collision avoidance (CSMA/CA). A detailed explanation of how DCF works will be presented in Chapter 11. However, we will explain the basic ideas here in order to motivate the problem.

In a random MAC protocol such as CSMA/CA, each user tries to access the medium whenever it senses the medium is not being used by another station (carrier sense). If upon a packet arrival, the station finds the medium busy, it will wait until the end of the transmission. When the medium becomes free, many stations may be waiting to transmit, so the following distributed mechanism is performed: each station waits a random number of *time slots* before attempting transmission. This is done by decrementing a *backoff counter* which was previously chosen at random in some interval, independently by each station. This mechanism allows to desynchronize stations after the medium becomes free. When some station decides to transmit, the rest will sense the medium busy and wait for the next opportunity.

It may also happen that two or more stations attempt transmission at the same time. In this case, the recovery of individual transmitted frames becomes impossible: this is called a *collision* and the data is lost. After each packet transmission, an acknowledgment packet is sent if the data arrived correctly. A missing acknowledgment is interpreted as collision by the stations and the packet is retransmitted at the following attempt.

In a typical scenario, the total number of stations is not known to each individual station. It is clear that the backoff counter must be adapted to the number of stations, in order to lower the access probability whenever the network is congested. This is performed in DCF by an exponential backoff mechanism, in which the backoff interval is doubled after each unsuccessful transmission, and reset to a certain predefined value whenever the transmission is successful.

As we mentioned before, after some station gets access to the medium, it may transmit at different data rates, to ensure reception at the destination. Since the packet length in Internet data transfers is typically fixed (with ≈ 1500 bytes being the dominant value), transmitting at different data rates translates into using the medium a different amount of time. This, however, is never taken into account by the DCF mechanism, and as we shall see this can lead to unfairness in the system.

The pioneering work of [Bianchi, 2000] analyzed the performance of DCF in 802.11 through a careful modelling of collisions and the backoff response process stipulated by the standard. This led to accurate means of predicting effective long term rates of a set of stations sharing the medium.

The analysis of [Bianchi, 2000] is however limited, since it is assumed that all stations have the same data rate (the only one available at the time). More recently in [Kumar et al., 2007] the analysis was extended to consider multiple physical data rates present in a single cell. The analysis provides an accurate model of the 802.11 MAC layer under saturation assumptions, i.e. the transmitting stations always have packets to send. This is, however, a very idealized model of the upper layer protocols. Some simple ways to account for the transport layer were suggested in [Kumar et al., 2007, Lebeugle and Proutiere, 2005], but there is no full consideration of the TCP congestion control mechanisms that interact with the MAC layer to determine user level performance.

As we introduced in Chapter 2, congestion control in the upper layers is commonly modelled through the Network Utility Maximization (NUM) approach of [Kelly et al., 1998]. This approach has been extended in recent years to cross-layer optimization in wireless networks (see for example [Lin et al., 2006, Chiang et al., 2007] and references therein). A significant portion of this literature is however based on *scheduled* MAC layers. In these networks, coordination among transmitting stations is needed in order to choose the appropriate set of links to turn on at each time, in order to avoid interference. The solution of this problem can be tackled via dual decomposition, and the scheduling component of the problem can be solved, though it poses difficulties for decentralization. Moreover, a separate queue per destination is needed at each node of the network.

In the case of random MAC protocols, most of the work has been theoretical, with an idealized model of the access protocol and particularly of collisions. In [Lee et al., 2006] a particular model (based on ALOHA protocols) is analyzed, while a more recent line of work developed in [Jiang and Walrand, 2008, Proutière et al., 2008] shows how a fine-grained control of medium access probabilities allows in principle to approach the performance of the scheduled case. While this research has deep theoretical implications, it has limited impact in current practice since it does not represent protocols such as IEEE 802.11.

Additionally, we argue that for wireless LAN technologies, the emphasis on collisions is misplaced. In recent versions of the IEEE 802.11 standard, the loss of performance due to collisions is not as important as one might expect, due to two main reasons: on one hand, most of the traffic is typically downlink, with an Access Point (AP) providing connectivity, and sending traffic to its attached stations. This access point does not collide with itself. Secondly, with the 802.11 default parameters, collision probabilities and resolution time are low when compared to data transmission, whenever the number of stations is not too large.

Far more relevant to user performance is the interaction between congestion control protocols like TCP and the multiple PHY data rates. In the following section we present a first contribution towards understanding this relationship, using a NUM approach. Recall that typical TCP protocols rely on losses to detect congestion. However, in the access point the buffer is served at different data rates depending on the destination, and so a refined model of the buffer AP is needed. We will show that coupling the TCP behavior with these multiple data rates can lead to inefficiencies in the system, with users able to transmit at higher data rates being severely penalized.

8.2 TCP resource allocation in a multirate wireless environment

We begin by considering a single cell scenario where N users, indexed by i, are downloading data from a single access point. Each station establishes a downlink TCP connection, and we denote by x_i the sending rate. These rates will be controlled by TCP in response to congestion.

The packets of each connection will be stored in the interface queue of the AP, before being put into the shared medium to reach its destination. The combined PHY and MAC layers offers each destination a different modulation rate, which combined with other factors such as protocol overheads results in an *effective rate* C_i offered to the connection packets. We assume for now these rates as given, and postpone to Chapter 11 the discussion on how to calculate them in 802.11 networks.

The first step in our analysis is to determine the average service rate y_i attained by each station in such a queue. As we mentioned before, we assume all packets having a fixed length L, and thus the time needed to transmit a single packet to station i is $T_i = L/C_i$. Typically the AP buffer is served with a FIFO discipline. Let $p_{HOL,i}$ denote the probability that the Head of

Line packet is for station i. We make the following assumption:

$$p_{HOL,i} = \frac{x_i}{\sum_j x_j},$$

meaning that the probability of having a packet for the station i as HOL is proportional to the input rates in the queue.

The average time between packet transmissions is $\bar{T} = \sum_j p_{HOL,j} T_j$. On average, station i will send a packet of length L with probability $p_{HOL,i}$ on each packet transmission, so the average rate for station i can be calculated as:

$$y_i = \frac{p_{HOL,i} L}{\bar{T}} = \frac{p_{HOL,i} L}{\sum_j p_{HOL,j} L/C_j} = \frac{x_i}{\sum_j x_j/C_j}. \tag{8.1}$$

The above equation is similar to equation (5) in [Kumar et al., 2007], which is established through similar arguments but considering collisions. In (8.1) we omitted the collision terms wince we are focusing in the downlink case.

To complete the loop, we now model the TCP behavior that determines the input rates x_i. Recall from Chapter 2 that TCP congestion control algorithms can be modelled as performing the following adaptation:

$$\dot{x}_i = k(x_i)(U_i'(x_i) - p_i),$$

where $U_i(x)$ is an increasing and strictly concave utility function, p_i is the congestion price (normally the loss probability), and $k(x_i) > 0$ a scaling factor. In our analysis we shall restrict ourselves to the $\alpha-$family of utility functions of Definition 2.4.

A simple model for the link loss rate for user i is the ratio between excess input rate and total rate, namely:

$$p_i = \left(\frac{x_i - y_i}{x_i}\right)^+ = \left(1 - \frac{1}{\sum_j x_j/C_j}\right)^+ = p,$$

where we have used (8.1) for y_i, and $(\cdot)^+ = \max(\cdot, 0)$ as usual. Note that with this model, loss probabilities for all users are equalized by the buffer.

The complete closed loop dynamics now follows:

$$\dot{x}_i = k(x_i)(U_i'(x_i) - p), \tag{8.2a}$$

$$p = \left(1 - \frac{1}{\sum_j x_j/C_j}\right)^+. \tag{8.2b}$$

We would like to characterize the equilibrium of these dynamics in term of a NUM problem. For this purpose, consider the following function $\Phi : \mathbb{R}_+^N \to \mathbb{R}$ given by:

$$\Phi(x) = \sum_i \frac{x_i}{C_i} - 1 - \log\left(\sum_i \frac{x_i}{C_i}\right),$$

whenever $\sum_i \frac{x_i}{C_i} > 1$ and 0 otherwise.

Lemma 8.1. *Φ is a convex function of x.*

Proof. Note that $\Phi(x)$ can be written as the composition map $\Phi(x) = g(f(x))$ where $f(x) = \sum_i x_i/C_i$ is a linear function of x and $g(u) = (u-1-\log(u))\mathbf{1}_{\{u>1\}}$, with $\mathbf{1}_A$ being the indicator function of the set A. It is easy to see that for $u > 0$, $g'(u) = (1-1/u)^+$ which is a nonnegative and increasing function of u. Therefore, g is increasing an convex and thus Φ is convex [Boyd and Vanderberghe, 2004, Section 3.2.4]. □

Consider now the following convex optimization problem:

Problem 8.1 (TCP Multirate Primal Problem)**.**

$$\max_{x_r \geqslant 0} \sum_i \frac{1}{C_i} U_i(x_i) - \Phi(x) \tag{8.3}$$

The above problem is in the form of the primal network Problem 2.3, the objective function being a total utility minus a network cost, captured in the function Φ. We have the following:

Theorem 8.1. *The equilibrium of the dynamics* (8.2) *is the unique optimum of Problem 8.1. Moreover this equilibrium is globally asymptotically stable.*

In order to prove the Theorem, let $V(x)$ denote the objective function of Problem 8.1. The following properties of V are needed:

Lemma 8.2. *The function V has compact upper level sets $\{x \in \mathbb{R}_+^N : V(x) \geqslant \gamma\}$, and is radially unbounded, i.e.* $\lim_{||x|| \to \infty} V(x) = -\infty$.

Proof. We split the proof in several cases. First we analyze the case $0 < \alpha < 1$, where we have the following bound:

$$\begin{aligned}
\sum_i \frac{1}{C_i} U_i(x_i) = \sum_i \frac{w_i x_i^{1-\alpha}}{C_i(1-\alpha)} &= \sum_i \frac{w_i C_i^{-\alpha}}{1-\alpha} \left(\frac{x_i}{C_i}\right)^{1-\alpha} \\
&\leqslant K \left(\max_i \frac{x_i}{C_i}\right)^{1-\alpha} \\
&\leqslant K \left(\sum_i \frac{x_i}{C_i}\right)^{1-\alpha}.
\end{aligned}$$

Here, $K = N \max_i \left\{\frac{w_i C_i^{-\alpha}}{1-\alpha}\right\}$ and we have used the fact that $x^{1-\alpha}$ is increasing.

Now consider $y = \sum_i x_i/C_i$, and assume that $y > 1$ so the function Φ does not vanish. Using the previous bound we have:

$$V(x) \leqslant K y^{1-\alpha} - y + 1 + \log(y) = h(y). \tag{8.4}$$

Since $\alpha > 0$, it is clear that $h(y) \to -\infty$ when $y \to \infty$. We conclude that $V(x) \to -\infty$ when $y \to \infty$, and since y is simply a weighted L^1 norm over the positive orthant, the same conclusion will hold when $\|x\| \to \infty$ for any (equivalent) norm.

We also have:

$$\{x : V(x) \geqslant \gamma\} \subset \{x : y < 1\} \cup \{x : h(y) \geqslant \gamma\},$$

and the above sets are both compact, the second due to the fact that h is continuous and $h(y) \to -\infty$ when $y \to -\infty$.

For the case $\alpha = 1$ we use the bound $x_i \leqslant C_i y$ with y as before, and therefore:

$$\sum_i \frac{1}{C_i} w_i \log x_i \leqslant \sum_i \frac{1}{C_i} w_i \log C_i y \leqslant K_1 + K_2 \log(y),$$

which gives a bound analogous to (8.4) with $h(y) = K_1 + (K_2 + 1)\log(y) - y + 1$ which also goes to $-\infty$ for $y \to \infty$, and the rest follows.

The third case is even simpler since for $\alpha > 1$ the utility is bounded above by 0, so we have $V(x) \leqslant -y + 1 + \log(y)$ and the result follows. □

We now prove the main result:

Proof of Theorem 8.1. By Lemma 8.1 and the strict concavity of the utilities we have that the objective function is strictly concave, and so Problem 8.1 has a unique optimum, that must satisfy the optimality conditions:

$$\frac{1}{C_i} U_i'(x_i) - \frac{\partial}{\partial x_i} \Phi(x) = 0 \quad \forall i.$$

By substituting the expression for Φ and differentiating, the above is equivalent to:

$$\frac{1}{C_i} \left[U_i'(x_i) - \left(1 - \frac{1}{\sum_j x_j / C_j} \right)^+ \right] = 0 \quad \forall i.$$

Identifying the last term as $p = p(x)$ in (8.2) the optimality conditions become:

$$U_i'(x_i) - p = 0 \quad \forall i,$$

which is the equilibrium condition of (8.2).

We now consider $V(x)$ as a Lyapunov function of the system. Differentiating along the trajectories we get:

$$\dot{V} = \nabla V \cdot \dot{x} = \sum_i \frac{k(x_i)}{C_i} (U_i'(x_i) - p)^2 \geqslant 0,$$

and so V is increasing along the trajectories. Moreover, $\dot{V} = 0$ only when $x = x^*$, the solution of Problem 8.1. Therefore the equilibrium is asymptotically stable. Since by Lemma 8.2 V is radially unbounded, stability holds globally. □

The function Φ in Problem 8.1 plays the role of a penalty function. It increases to infinity whenever $\sum_i x_i/C_i > 1$. Note that this last inequality can be interpreted in terms of the "time share" allocated to users. This suggests that the above problem can be interpreted as an approximation of the following Network Utility Maximization:

Problem 8.2 (Scaled multirate network problem)**.**

$$\max_x \sum_i \frac{1}{C_i} U_i(x_i),$$

subject to:

$$\sum_i \frac{x_i}{C_i} \leqslant 1.$$

This NUM problem has two variations with respect to the standard Network Problem 2.2. The first is that the constraint is rewritten in terms of x_i/C_i, which as mentioned above can be interpreted as the time proportion the shared medium is used by connection i. The sum of the allocated time proportions less than one is a natural way to generalize the wired capacity constraints to a shared multirate substrate.

The second important difference is the scaling factor $1/C_i$ for the user utility. This is *not* a natural feature of the problem, instead it is a consequence of the congestion control and resource sharing mechanisms actually in use in these networks. The main consequence of this fact is a bias against users of high physical rates, which are given less weight in the net utility. The effect of this bias is a radical equalization of rates dominated by the slowest stations, as shown by the following example.

Example 8.1. Assume 3 users are downloading data from a single AP, and they have equal utilities $U(x) = -\frac{1}{\tau^2 x}$ which models TCP/Reno response with τ being the connection round trip time. Assume their lower layer effective rates are $C_i = 10$. In this case the solution of Problem 8.2 is $x_i^* = 3.333$ for all three users. Now if for instance user 3 changes its radio conditions and obtains a lower value of effective rate such as $C_3 = 1$, the new allocation is:

$$x_1^* = x_2^* = x_3^* = 0.8333.$$

Note that despite the fact that only one user worsened its radio conditions, all three users are downgraded, and in particular the fastest ones are more heavily penalized due to the user 3 inefficiency. This problem has been observed in practice in 802.11 environments, and we will exhibit it by simulation in Chapter 11. Note that if instead of solving Problem 8.2 we plug the values into 8.1, the solution would be $x_i^* = 0.89$ for all i thus showing that the barrier function approximation is very close.

The key in the above example is that all users share the same utility function. In that case it is easy to show the following result:

Proposition 8.1. *In the case where all users share a common utility function $U_i = U$, the solution of Problem 8.2 is:*

$$x_i^* = \frac{1}{\sum_j 1/C_j}. \tag{8.5}$$

The allocated rates are therefore equalized to the harmonic mean of the effective data rates. This is in accordance with results obtained in [Kumar et al., 2007] for multiple rate 802.11 networks, where collisions are considered. In fact, in that work the rate in equation (8.5) appears as an upper bound on the realistic rate of permanent connections. However, we remark that in [Kumar et al., 2007] the TCP layer is not modelled; rather, it is assumed that the AP has equal probability of serving all users.

Here we have modelled TCP, and we find that (8.5) only holds under the assumption of equal utilities, but independent of the TCP flavor (the value of α). However, if users have different utility functions, as is the case when TCP-Reno connections have different RTTs, this solution is no longer valid, and the allocation must be calculated by solving Problem 8.2.

The preceding discussions suggest that ways of removing the artificial bias from Problem 8.2 should be explored. This is the main purpose of the following chapter.

9

Efficiency and fairness in multirate wireless networks

Up to this point, we focused on analyzing the resource allocation established by current standard protocols. This led us to establish that in this case resources are allocated as in the Scaled multirate network Problem 8.2, which introduces a bias against the users or stations with more effective usage of the shared medium.

We would like to devise mechanisms to drive the system to a NUM inspired resource allocation, but without the preceding bias, leading to a more effective, yet fair, resource sharing.

Moreover, we would like these mechanisms to be decentralized, and compatible with current practices, in particular with typical congestion control protocols, as well as wireless standards, in order for them to be deployable.

In this chapter we focus on this task. First, in Section 9.1 we analyze the case of a single cell from a theoretical perspective. Then in Section 9.2 we describe a suitable packet level algorithm to drive the system to the desired equilibrium. Finally, in Section 9.3 we discuss more general topologies, in particular mixed wired-wireless ones, and discuss how to extend the results to this case.

9.1 The single cell case

The natural optimization problem to solve in the single cell scenario is similar to Problem 8.2 but without the terms $1/C_i$ that scale the different utilities. We formulate it as follows:

Problem 9.1 (Wireless Multirate Network Problem)**.**

$$\max_x \sum_i U_i(x_i),$$

subject to the constraint:

$$\sum_i \frac{x_i}{C_i} \leqslant 1 \tag{9.1}$$

As formulated, the above NUM problem becomes a special case of those in the literature of *scheduled* wireless networks (see e.g [Lin et al., 2006] and references therein), where the set of feasible rates is taken to be the convex hull of rates achievable by independent (non-interfering) sets of links. In the current scenario, only individual links are schedulable without interference, at rate C_i. The convex hull then becomes (9.1). However, in contrast with these references, we will seek a solution to Problem 9.1 that does not use a complicated scheduling mechanism in the AP, which would imply a significant departure from current 802.11 networks.

In order to do so, let us write the Lagrangian of Problem 9.1, which is:

$$\mathscr{L}(x,p) = \sum_i U_i(x_i) - p\left(\sum_i \frac{x_i}{C_i} - 1\right).$$

The KKT conditions for this problem are:

$$U_i'(x_i) - \frac{p}{C_i} = 0,$$
$$p\left(\sum_i \frac{x_i}{C_i} - 1\right) = 0.$$

The first equation is of particular interest: it tells us that even if there is a single price associated with the constraint (9.1), each individual connection must be charged with a scaled version of this price, with the scaling factor $1/C_i$. That is, if we call $p_i = p/C_i$, the price seen by source i, this price will be higher for connections with lower rates, thereby charging the sources according to their own channel inefficiency.

To see the difference with Problem 8.2 more clearly, let us consider again the case where all users share the same utility function $U_i(x) = U(x)$ from the $\alpha-$fair family with equal weights $w_i = 1$. In this case, the KKT conditions become:

$$U_i'(x_i) = x_i^{-\alpha} = \frac{p}{C_i},$$
$$p\left(\sum_i \frac{x_i}{C_i} - 1\right) = 0.$$

Since $p > 0$, imposing equality in the second equations yields:

$$p^{-\frac{1}{\alpha}} \sum_i C_i^{\frac{1}{\alpha}-1} = 1,$$

from where we obtain the optimal rates:

$$x_i^* = \frac{C_i^{\frac{1}{\alpha}}}{\sum_j C_j^{\frac{1}{\alpha}-1}}. \tag{9.2}$$

We see that, in contrast to (8.5), rates are no longer equalized in the solution, and users with larger C_i will receive a larger share of resources.

We analyze now some important cases. First we observe that, when $\alpha \to \infty$, we recover the allocation (8.5) which is also the max-min fair allocation for the system. As noted before, this can be very inefficient, and moreover can heavily penalize the higher rate users.

When $\alpha \to 0$ we recover the max-throughput allocation, which is simply to allocate all resources to the users with the higher effective rates C_i. This however starves all the slower stations.

The case $\alpha = 1$, which is proportional fairness, has the following nice property, which is verified directly from equation (9.2):

Proposition 9.1. *In the case where all connections share the same utility* $U(x) = w\log(x)$ *(*$\alpha = 1$*), the equilibrium of Problem 9.1 is:*

$$x_i^* = \frac{C_i}{N} \quad \forall i = 1, \ldots, N.$$

In particular, the allocated rate for user i *depends only on its own effective rate and the total number of users in the cell.*

Equivalently: under proportional fairness *time* is shared equally among users, but those with a more efficient use of time (i.e. higher C_i) can obtain a proportionally greater rate. This protects fast users from the lower rate ones.

If we return to Example 8.1, we see that when all users have $C_i = 10$, the allocation is $x_i^* = 3.333$, but when user 3 changes its rate to $C_3 = 1$, the allocated rates x_1^* and x_2^* do not change, and $x_3^* = 0.333$, so only user 3 is penalized by the change.

When utilities are chosen to represent TCP-Reno connections, that is $\alpha = 2$, we do not have complete protection, but the situation is nevertheless improved from the max-min case. The following is a numerical example.

Example 9.1. Consider again the same situation of Example 8.1, and utilities chosen from the $\alpha-$fair family with $\alpha = 2$ and $w_i = 1$. When all users have $C_i = 10$, the optimal allocation from

Problem 9.1 is $x_i^* = 3.333$ as before. When user 3 changes its radio conditions to $C_i = 1$ the allocation is:

$$x_1^* = x_2^* = 1.93, \; x_3^* = 0.61.$$

The total network throughput increases by $\approx 80\%$ with respect to Example 8.1, and the fastest users are not as heavily penalized.

Having argued that Problem 9.1 is indeed a reasonable way to allocate resources, we concentrate now in finding a decentralized algorithm that drives the system to the desired allocation.

Consider the following primal-dual algorithm, similar to the ones introduced in Chapter 2:

$$\dot{x}_i = k(x_i)\left(U_i'(x_i) - \frac{p}{C_i}\right), \tag{9.3a}$$

$$\dot{p} = \left(\sum_i \frac{x_i}{C_i} - 1\right)_p^+. \tag{9.3b}$$

where as usual, $k(x_i) > 0$ and $(\cdot)_p^+$ is the positive projection.

The following Theorem is a direct consequence of the results in [Arrow et al., 1958] and [Feijer and Paganini, 2009]:

Theorem 9.1. *The equilibrium of the dynamics* (9.3) *is the unique optimum of Problem 9.1. Moreover, this equilibrium is globally asymptotically stable.*

Therefore, congestion control theory provides us with a satisfactory algorithm to drive the system to the equilibrium. Note that the AP must integrate equation (9.3b) and produce a price p, whereas each connection should react according to equation (9.3a); in particular each connection should react to a scaled version of the price p/C_i, depending on its effective rate.

If we want to implement such an algorithm in practice, the main issue is how to communicate the C_i, which are dependent on the radio conditions and known only by the AP, to the end points of the connections. This is what we study in the following section.

9.2 Multirate random early detection

The purpose of this section is to develop a packet-level algorithm that mimics the behavior of the dynamics (9.3). The first step towards developing such an algorithm is to find an interpretation of the magnitudes involved.

We begin with the price p. In the wired case, typical dual algorithms have interpreted the price variable as the queueing delay [Low and Lapsley, 1999, Low et al., 2002]. This is also the case here, in this modified version. By integrating $\dot{p}$ in equation (9.3b) we see that p tracks the amount of time the shared medium is not capable of coping with the demands, and thus the incoming workload is accumulated in the queue generating delay.

More formally, let b_i denote the amount of data of connection i in the AP buffer (assume it is non empty). Then:

$$\dot{b}_i = x_i - y_i = x_i - \frac{x_i}{\sum_j x_j / C_j}, \tag{9.4}$$

where we have used equation (8.1) for the drainage rate y_i.

The delay in the queue for an incoming packet will be the time needed to serve all the data before its arrival. Since the data b_i is served at an effective rate C_i we have that:

$$d = \sum_i \frac{b_i}{C_i}.$$

By differentiating and substituting the expressions for $\dot{b}_i$ we arrive at:

$$\dot{d} = \sum_i \frac{\dot{b}_i}{C_i} = \sum_i \frac{x_i}{C_i} - 1.$$

Moreover, when the buffer is empty, $d = 0$ and it can only increase (whenever $\sum_i \frac{x_i}{C_i} > 1$), which is exactly what is allowed by the positive projection.

Therefore, the price p in equation (9.3b) can be interpreted as the queueing delay in the AP. If in particular all the effective rates are equal ($C_i = C$), we recover the delay based model of [Low and Lapsley, 1999, Low et al., 2002] for wired networks.

The problem with using the queueing delay as the price variable is that there is no simple way to scale it in order to transmit the scaled version $p_i = p/C_i$ to the sources. The queueing delay is uniform across the sources, and therefore it is the TCP layer that must know the C_i in order to react accordingly. However, in order to do this, the source must be aware of its MAC level rate, which is infeasible since sources may be far away in the network and may not even be aware that they are congested at a wireless link. Moreover, typical TCP implementations react to packet losses.

We now discuss a practical method to overcome these limitations, without resorting to a complicated scheduling mechanism. In order to drive the system to the optimum of Problem 9.1, we propose to use a simple Active Queue Management policy, which we call the Multirate Random Early Detection (MRED) algorithm.

Instead of using queueing delay as the price, we propose to use as a proxy the buffer length b. To generate the price, the AP discards packets for connection i randomly with probability p_i proportional to b/C_i. This gives a linear Random Early Detection (RED) algorithm [Floyd and Jacobson, 1993], but with probabilities related to the effective data rates C_i.

Note that this mechanism can be implemented in the AP, resorting only to local information. The AP must know the destination address for the incoming packet, the effective data rate with that destination and the buffer contents in order to decide whether to discard the packet.

We now make a model for the system. Let $p_i = \kappa b / C_i$ be the loss probability of connection i, with $\kappa > 0$ the proportionality constant of MRED. The TCP source reacts to this loss probability, and it can be modelled by a primal controller (equation (2.6)). A simple model for the total buffer content is:

$$\dot{b} = \left(\sum_i x_i - \sum_i y_i \right)_b^+ ,$$

i.e. the rate of change of the content in the buffer is the difference between total input rate and total output rate.

Substituting the expression for y_i in equation (8.1) we get the following dynamics for the system:

$$\dot{x}_i = k(x_i) \left(U_i'(x_i) - \frac{\kappa b}{C_i} \right), \tag{9.5a}$$

$$\dot{b} = \left(\sum_i y_i \right) \left(\sum_i \frac{x_i}{C_i} - 1 \right)_b^+ . \tag{9.5b}$$

Note that the second equation in (9.5) is similar to the equation for p in the primal-dual algorithm, with an added state dependent gain $\sum_i y_i > 0$. In particular, the equilibrium values of (9.5) x_i^* and $p^* = \kappa b$ satisfy the KKT conditions of Problem 9.1. As for stability, global stability results for these dynamics are harder to obtain, due to the varying gain term. However, we argue now that locally around the equilibrium the system above behaves exactly as the dual dynamics (9.3b). Therefore, the system will be locally asymptotically stable.

Consider equation (9.5b), and denote by δb, δx, δy the system variables around equilibrium. We have that, near the equilibrium, the system verifies:

$$\dot{\delta b} = \left(\sum_i y_i^* \right) \left(\sum_i \frac{\delta x_i}{C_i} - 1 \right) + \left(\sum_i \delta y_i \right) \left(\sum_i \frac{x_i^*}{C_i} - 1 \right),$$

and the second term is 0 due to the equilibrium condition of the original dynamics. Therefore:

$$\dot{\delta b} = \gamma \left(\sum_i \frac{\delta x_i}{C_i} - 1 \right),$$

which is exactly the linear behavior of the dual dynamics (9.3b) with a fixed gain $\gamma > 0$. Since the primal-dual dynamics are asymptotically stable, the new system (9.5) will be locally asymptotically stable.

We have thus proved that:

Proposition 9.2. *The dynamics* (9.5) *are locally asymptotically stable and its equilibrium is the solution of Problem 9.1.*

Note that the algorithm (9.5) is easier to implement at the packet level than the original primal dual dynamics, which involve measuring the buffering delay. Since only local stability results were obtained, in Chapter 11 we will explore its behavior by simulation.

9.3 Extension to wireless access networks

The analysis of Section 9.1 is valid in a single cell setting. We would like to generalize it to more complicated networks. In particular, we are interested in the case of wireless access systems, i.e. a wired backbone which has non-interfering wireless cells as stub networks. This is a common setting in practice for wireless access in office buildings and college campuses.

In such networks, one obtains a combination of classical capacity constraints for wired links (that do not interfere with any other links), and constraints of the type (9.1) for links of the same stub cell interfering with one another, potentially with different effective data rates. For such networks, we would like to develop a price scaling method that enables decentralized users to allocate resources to maximize utility.

Of course, one could further consider more general interference models, such as different wireless cells interfering with each other, or more arbitrary interference patterns as has been considered in the scheduling literature [Lin et al., 2006]. These, in addition to the complexity of scheduling, lead to optimization problems that are difficult to decentralize. For this reason, we chose to focus on a narrower setting which nevertheless covers scenarios of practical importance, and which can be addressed through a smaller departure from current practice, in particular using currently deployed MAC layers.

Consider then a network composed of links $l = 1, \ldots, L$. These links can be wired or wireless, and each one has an effective transmission rate C_l. In the case of wired links, C_l is the link capacity. In the wireless case, it is the effective data rate discussed before. Let $i = 1, \ldots, n$ represent the connections, with rate x_i, and R the classical routing matrix, i.e. $R_{li} = 1$ if connection i traverses link l.

To represent the contention inherent to the networks under analysis, we group the links l into contention sets: two links belong to the same contention set if they cannot be transmitting simultaneously, and we define a matrix given by $G_{kl} = 1$ if link l belong to contention set k, and 0 otherwise. We call G the contention matrix.

If a link is wired, the contention set is a singleton, since it does not interfere with any other link. In the case of a wireless cell, the contention set is composed of all links that depend on the same access point. We shall assume that each link belongs to only one contention set: this is the restriction we impose on the interference model in order to obtain tractable solutions.

The network capacity constraints, following the idea of expressing constraints as "time proportions” of allocated medium of (9.1), can be written as:

$$Hx \leqslant \mathbf{1}, \quad H = GC^{-1}R,$$

with R and G as defined above, $C = \mathrm{diag}(C_l)$ and $\mathbf{1}$ a column vector of ones. Note that each constraint is associated with a contention set, and each row of the above equation corresponds to the time proportions associated with each user in the contention set summing less than 1.

It is easier to see how this framework enables us to model different situations via some examples, which we present below.

Example 9.2 (Wired network). If all links are wired, the contention matrix G is the identity matrix. By taking R and C as before, we recover the classical wired constraints:

$$\sum_i R_{li} x_i \leqslant C_l.$$

Example 9.3 (Single wireless cell). If there is only one wireless AP with N users in the cell, we can take R as the identity matrix, C as the diagonal matrix with wireless effective capacities and $G = \mathbf{1}^T$, since there is only one contention region containing all the links. We then recover the constraint:

$$\sum_i \frac{x_i}{C_i} \leqslant 1,$$

discussed in Section 9.1.

Example 9.4 (Wireless distribution system). To see a more complete example, consider the network composed of wired and wireless links shown in Figure 9.1. This topology appears in outdoor wireless distribution scenarios, where the backhaul nodes distributes the access to several hotspots (two in this example) using one channel or frequency range of the wireless medium. The local hotspots communicate on another frequency with the hosts, thus non interfering with the backhaul communication.

We can model the capacity constraints of this network within the above framework by taking:

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

$$C = \mathrm{diag}(C, C_{AP_1}, C_{AP_2}, C_1, C_2, C_3, C_4).$$

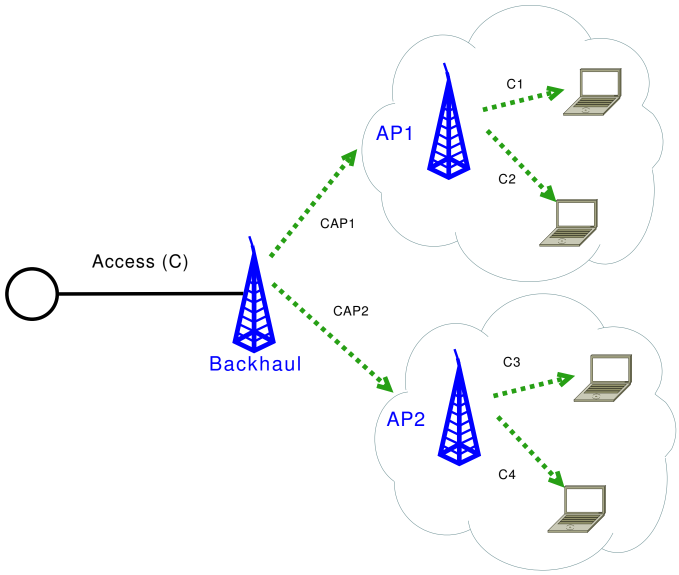

FIGURE 9.1: TOPOLOGY OF A MIXED WIRED-WIRELESS DISTRIBUTION SYSTEM WITH 4 END-USERS.

Within the above framework, we can now pose the general Network Utility Maximization problem for this class of networks, which is:

Problem 9.2 (General Wired-Wireless Network Problem)**.**

$$\max_x \sum_i U_i(x_i)$$

subject to:

$$Hx \leqslant \mathbf{1}.$$

The previous problem seeks an optimal allocation within the natural constraints of the network, expressed in terms of allocated time proportions. These constraints are similar to the ones used in the scheduling literature. We will show that for the special structure under consideration, a decentralized solution can be obtained, that involves simpler buffer management solutions and standard congestion control, provided a suitable price scaling as the one introduced in Section 9.1 is applied.

Consider now the Lagrangian of Problem 9.2:

$$\mathscr{L}(x,p) = \sum_i U_i(x_i) - p^T(Hx - \mathbf{1}), \tag{9.6}$$

where $p = (p_1, \ldots, p_K)^T$ is the vector of prices. We see therefore that we have one price for each contention set.

By denoting $q = H^T p$, the KKT conditions of Problem 9.2 are:

$$U_i'(x_i) = q_i \quad \forall i,$$
$$p_k \left(\sum_i \sum_l \frac{G_{kl} R_{li}}{C_l} x_i - 1 \right) = 0,$$

where q_i is given by:

$$q_i = \sum_{l:i \in l} \sum_{k:l \in k} \frac{p_k}{C_l}. \tag{9.7}$$

Therefore, the connection must react to a price which is the sum of the prices of the contention sets it traverses, divided by the link capacities it uses within this set.

Again, to solve Problem 9.2 we can use a primal-dual algorithm with the following dynamics:

$$\dot{x}_i = k(x_i)(U_i'(x_i) - q_i), \tag{9.8a}$$
$$\dot{p} = (Hx - \mathbf{1})_p^+, \tag{9.8b}$$
$$q = H^T p. \tag{9.8c}$$

These dynamics, are globally asymptotically stable due to the results in [Arrow et al., 1958] and [Feijer and Paganini, 2009]. Its equilibrium is exactly the solution of the General Wired-Wireless Network Problem 9.2.

The key remark here is that, if each contention set in the matrix H is associated with only one AP, as it was in the preceding examples, the price dynamics of (9.8) tracks the queueing delay at this AP. Therefore, a decentralized mechanism can be achieved, since the contention price that must be transmitted to the source will be the queuing delay at each AP traversed by the source, scaled by the effective capacity of the link associated with this AP. A packet dropping mechanism such as MRED can be used to scale and translate this queueing delay into a suitable loss probability, which aggregated along the routes will produce the correct q_i for each source.

If links are allowed to interfere across APs, the situation is more complicated, since the queueing delays become correlated, and sources must react to congestion prices which are not on their route, thus decentralization is not possible without message passing between the APs. As mentioned before, we will leave out this case and analyze how we can decentralize the algorithm in some examples of practical importance.

Example 9.5 (Mixed wired-wireless access network). A typical configuration for wireless access coverage is to distribute access points in non overlapping channels across the region to cover, and wire them to the Internet access node. This produces a tree like topology as Figure 9.2. There, the different APs are connected to a central switch, which is also connected to the

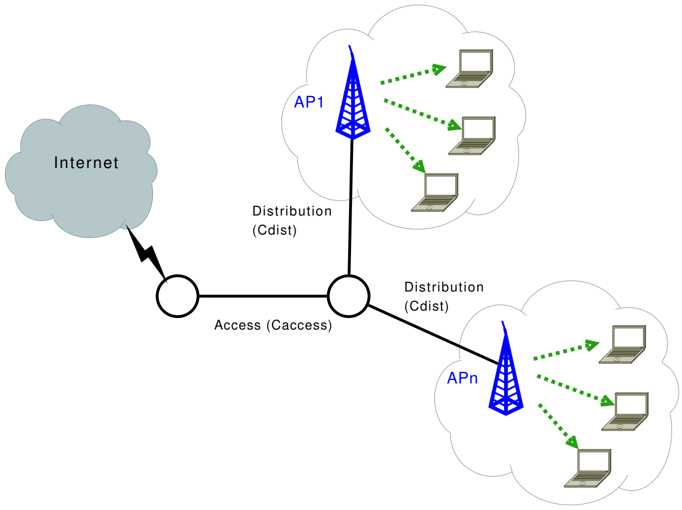

FIGURE 9.2: TOPOLOGY OF A MIXED WIRED-WIRELESS ACCESS NETWORK.

router handling the Internet connection. End users are then connected to the APs via 802.11 for example. In this case, each user traverses three contention sets: the contention in its own AP and the two wired links.

Assuming the link capacities and user distributions shown in Figure 9.2, the corresponding price for connection i is calculated according to equation (9.7) as:

$$q_i = \frac{p_{access}}{C_{access}} + \frac{p_{dist_j}}{C_{dist_j}} + \frac{p_{AP_j}}{C_i}$$

where p_{access} and p_{dist_j} are the queueing delays of the wired links traversed by packets of connection i, whereas p_{AP_j} is the queueing delay of the FIFO queue at the corresponding AP. All prices are scaled by their corresponding link capacities, which are locally known.

By using for instance the MRED algorithm of Section 9.2, we can transmit this price to the source, and impose the notion of fairness of Problem 9.2 by emulating the dynamics (9.8).

Example 9.6 (Wireless distribution system). A variation of the above example occurs when the area to cover is large, as is the case in large outdoor deployments. In this case, the distribution links that connect each AP with the wired network are replaced by a wireless cell that handles the backhaul, recovering the situation of Example 9.4, shown in Figure 9.1.

In this case, the corresponding price for connection i can again be calculated according to (9.7) as:

$$q_i = \frac{p_{access}}{C_{access}} + \frac{p_{BH}}{C_{AP_j}} + \frac{p_{AP_j}}{C_i}$$

The main difference with the above example is that p_{BH} is common to all distribution links, it reflects the FIFO queueing delay at the backhaul node, and C_{AP_j} is the capacity the backhaul

AP uses to communicate with the AP of connection i. Again, by using Multirate RED in this tree topology we can impose the notion of fairness of Problem 9.2.

In Chapter 11 we will analyze the behavior of these algorithms in some of this examples through packet level simulations.

To end this chapter, we would like to point out which is the common characteristic all these networks share that enables a simple distributed solution. The essence is that each link belongs to a single contention set, and each such set is served by a common queue. Under this assumption, the inner sum in (9.7) consists of a single term, i.e. we can write:

$$q_i = \sum_{l:i\in l} \frac{1}{C_l} p_{k(l)},$$

where $k(l)$ is the contention set associated with link l, and $p_{k(l)}$ the corresponding price. By performing the correct capacity scaling, this price can be appropriately relayed to the sources.

10

Connection level analysis of wireless networks

In this chapter we turn to the analysis of multirate wireless networks at the connection level, focusing on the case of a single cell. We develop a stochastic model for the evolution of the number of connections present in the cell, that tries to capture the time and spatial behavior of a connection arrivals, as well as the resource allocation that the lower layers (TCP, multirate MAC) impose on the rate of these connections.

Based on the models of previous chapters, we construct a model for random arrivals of connections in the coverage region of the cell. Our approach generalizes the connection level stochastic model described in Section 3.3, in order to take into account that arrivals in different regions of the cell lead to different radio conditions, and thus distinct effective data rates.

The main purpose is to derive stability conditions for this stochastic process, as well as some performance measures, in particular the connection level throughput achieved by different resource allocations methods.

The organization of this chapter is as follows. In Section 10.1 we describe our stochastic model for the connections in the cell. Then, in Section 10.2 we analyze the model assuming that the underlying allocation is done by standard TCP and MAC level algorithms, as studied in Chapter 8. Finally, in Section 10.3 we derive results for the connection level timescale assuming that the network operates under the more efficient resource allocations of Chapter 9.

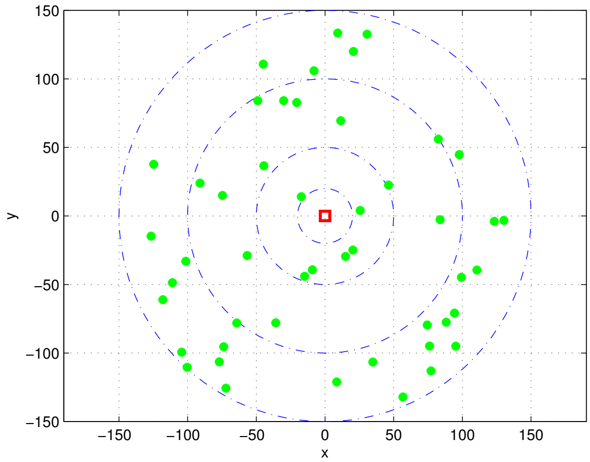

FIGURE 10.1: A TYPICAL 802.11 CELL WITH GEOMETRICALLY RANDOM CONNECTION DEMANDS AND CIRCULAR RATE REGIONS.

10.1 Model description

In this section we describe the underlying model and assumptions that will be used in this chapter. We would like to model a single cell scenario where downlink connections arrive randomly, with random workload sizes. Since the effective data rates that connections get at the MAC layer are determined by radio conditions, we would like our model to capture this situation by representing connection arrivals associated to a location in the cell. This location will determine the effective data rate.

Assume that the AP is located at the origin of the plane $\mathbb{R}^2$. Let $R_j \subset \mathbb{R}^2$ be the region of the plane where users can achieve a transmission rate C_j. As an example, in an outdoor setting the R_j could be concentric discs around the origin, with decreasing physical layer rates as depicted in Figure 10.1. Our model, however, requires no assumptions on the shape of the different rate regions.

We describe first the arrival of connections. Stations can be anywhere in the cell and connections may arrive randomly in time. We model this by assuming that connections follow a spatial birth and death process [Baccelli and Zuyev, 1997]. New connections appear in the cell as a Poisson process of intensity $\lambda(x), x \in \mathbb{R}^2$ which represents the frequency per unit of area. In particular, users arrive to region R_j with intensity:

$$\Lambda_j = \int_{R_j} \lambda(x)\,dx.$$

Each connection demands some random amount of workload which we assume exponential with mean $1/\mu$. For simplicity we assume that these workloads have the same mean in all regions. This is a reasonable assumption since the connection sizes are generally independent of the location. The use of exponential distributions for connection size is questionable, we do this in order to obtain a tractable model. We refer to [Bonald and Massoulié, 2001, Paganini et al., 2009] for a discussion.

We first note that the arrival process to each region j is a Poisson process with total intensity Λ_j, irrespectively of whether new connections are generated by different stations or they belong to the same station (in the same position in space). Both situations can be captured by the total arrival rate Λ_j.

To complete the system, and thus derive stability conditions and performance metrics at the connection level timescale, we must specify the service rate of each connection. In this regard, we will analyze both the biased resource allocation discussed in Chapter 8, and the alternative proposed in Chapter 9.

10.2 Connection level performance of current TCP-multirate environments.

Suppose that rates are allocated to ongoing connections following the TCP resource allocation of Problem 8.2. As argued in Section 8.2, this approximately models the behavior of the prevailing loss-based TCP congestion control when connections share a FIFO queue, serviced with multiple lower-layer rates.

For simplicity, assume that all connections share the same utility function, for which the solution of Problem 8.2 is given by equation (8.5), that is, the harmonic mean of the effective data rates of ongoing connections.

In this case, given that at a certain moment of time there are n_j connections of rate C_j, the rate for every connection is given by

$$x(n) = \frac{1}{\sum_j n_j / C_j}; \tag{10.1}$$

here we denote by n the vector of n_j.

Putting together all the previous considerations we have the following Markov chain model for the vector $n(t)$.

$$n \mapsto n + e_j \quad \text{with intensity} \Lambda_j, \tag{10.2a}$$

$$n \mapsto n - e_j \quad \text{with intensity} \mu n_j x(n), \tag{10.2b}$$

where e_j denotes the vector with a 1 in the $j-$coordinate and 0 elsewhere.

The first question of interest for this model is the stochastic stability region, i.e. the region of arrival intensities that produce an ergodic Markov process, and hence a stationary distribution. We are also interested in calculating the throughput of connections. Both issues will be studied by identifying the above model with a well known queue.

Substituting equation (10.1) in (10.2) we can rewrite the death rates of the Markov process as

$$\mu n_j x(n) = \mu C_j \frac{g_j n_j}{\sum_k g_k n_k},$$

where $g_j := \frac{1}{C_j}$.

With this notation, we can identify the transition rates of (10.1) with those of a *Discriminatory Processor Sharing* (DPS) queue [Altman et al., 2006], with total capacity 1, and where for each class j the arrival rate is Λ_j, the mean job size $v_j = \mu C_j$ and the DPS weight is g_j.

With this identification, the following result follows directly from the stability condition of the DPS queue:

Proposition 10.1. *The Markov process describing the number of connections is ergodic if and only if:*

$$\varrho = \sum_j \frac{\Lambda_j}{\mu C_j} < 1. \tag{10.3}$$

It is worth specializing equation (10.3) to the important case in which connections arrive uniformly distributed in the cell, i.e. $\lambda(x) = \lambda$, a constant. This can represent a situation where users do not know where the AP is. In that case, if A_j is the area of the region R_j, $\Lambda_j = \lambda A_j$ and the stability condition becomes

$$\frac{\lambda}{\mu} < \frac{1}{\sum_j \frac{A_j}{C_j}}. \tag{10.4}$$

This is of the form $\rho < C^*$ where ρ is the traffic intensity in $bits/(s \cdot m^2)$ and C^* can be thought as a cell traffic capacity, which captures the geometry, and is a weighted harmonic mean of the effective rates.

We have thus proved the following:

Proposition 10.2. *For a single cell with uniform random arrivals, and where each effective rate C_j covers a region of area A_j, the maximum cell capacity (in rate per area unit) is given by:*

$$C^* = \frac{1}{\sum_j \frac{A_j}{C_j}}.$$

The second issue we are interested in is performance, which we measure by the connection level throughput, whenever the system is stable (i.e. $\varrho < 1$). This can be evaluated by calculating the expected time in the system for jobs in a DPS queue.

The expected waiting time in a DPS queue was analyzed in [Fayolle et al., 1980], and extended by [Haviv and van der Wal, 2008]. Let τ_j be the time in the system for a job of class j.

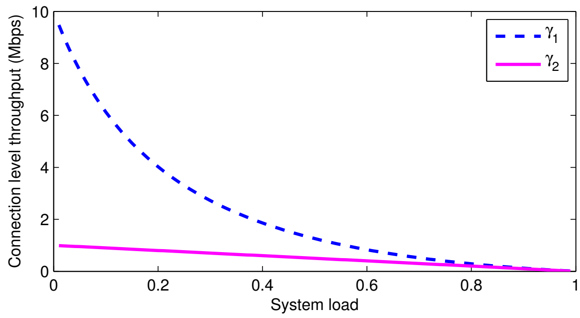

FIGURE 10.2: CONNECTION LEVEL THROUGHPUTS, 2 CLASSES, $C_1 = 10$, $C_2 = 1$ MBPS. ARRIVAL RATES ARE PROPORTIONAL TO COVERAGE AREAS.

For the case of two classes there is an explicit formula:

$$E(\tau_1) = \frac{1}{\mu C_1(1-\varrho)}\left(1 + \frac{\Lambda_2(C_1 - C_2)}{\mu C_2^2(2-\varrho)}\right),$$

$$E(\tau_2) = \frac{1}{\mu C_2(1-\varrho)}\left(1 + \frac{\Lambda_1(C_2 - C_1)}{\mu C_1^2(2-\varrho)}\right).$$

where ϱ is the system load as in equation (10.3).

Observing that a user can send an average $1/\mu$ bits during time τ_j we can measure the connection level throughput as:

$$\gamma_j = \frac{1}{\mu E(\tau_j)}$$

As an example, we calculate now the connection level throughput for a cell with two allowed rates as in Example 8.1, namely $C_1 = 10$ and $C_2 = 1$ Mbps, by applying the previous formula.

Example 10.1. We assume that connections arrive uniformly in space, so the proportion of slow connections is greater. Results are shown in Figure 10.2, that shows connection level throughput under varying cell load. Note that when the load increases, the connection level throughput of both classes equalizes to the detriment of faster users, with no appreciable gain for the slower ones; this is a consequence of the allocation that equalizes per-connection rates.

For the general case with multiple classes simple explicit formulas for the τ_j are not available. Rather, these values can be obtained by solving a system of linear equations:

$$E(\tau_j) = B_{j0} + \sum_{k=1}^{m} E(\tau_k)\Lambda_k B_{jk} \tag{10.5}$$

where the B_{jk} depend on the system parameters (we refer the reader to [Altman et al., 2006] for the expressions). Therefore, we can calculate the connection level throughputs γ_j by numerically solving this system for the effective data rates of the network. A case study for the IEEE 802.11 standard will be given in Chapter 11.

Results similar to the above example were given in [Bonald and Proutière, 2006], where a connection level model is also analyzed. Once again, however, in this work the lower layers are not modeled, and a simple time-sharing mechanism is assumed for the medium. Here we have found that the “downward" equalization of connection-level throughputs will occur with any $\alpha-$fair congestion control, in particular current implementations, provided utilities are the same for all sources.

10.3 Connection level performance of the unbiased allocation

We now analyze the connection level performance of the resource allocation of Problem 9.1, which removes the bias in utility against fast users, and as discussed in Section 9.2 can be implemented through a price scaling mechanism.

Again, we will assume that all users share the same utility function $U(x) = x^{1-\alpha}/(1-\alpha)$ of the $\alpha-$fair family. In this case the solution of Problem 9.1 is given by (9.2); given that at a certain moment of time there are n_j connections of rate C_j, we obtain the connection rates

$$x_i(n) = \frac{C_i^{1/\alpha}}{\sum_j n_j C_j^{1/\alpha-1}}. \tag{10.6}$$

The connection level process behaves again as a DPS queue, with arrival rates Λ_j, job sizes μC_j and weights depending on α as:

$$g_j = C_j^{1/\alpha-1}.$$

Therefore, for any α, the system will be stable when the loads verify the stability condition (10.3), and the expected job service times and connection level throughput can be calculated by the same method we described above.

It is worth noting that as $\alpha \to \infty$, we recover the weights of the current allocation analyzed in the previous section, which is max-min fair but can be highly inefficient, penalizing the highest rates which suffer the most slowdown.

For the case of $\alpha = 1$ corresponding to proportional fairness, the weights associated to each class become equal and the performance metrics can be explicitly calculated as:

$$E(\tau_j) = \frac{1}{\mu C_j(1-\varrho)},$$

and:

$$\gamma_j = C_j(1-\varrho).$$

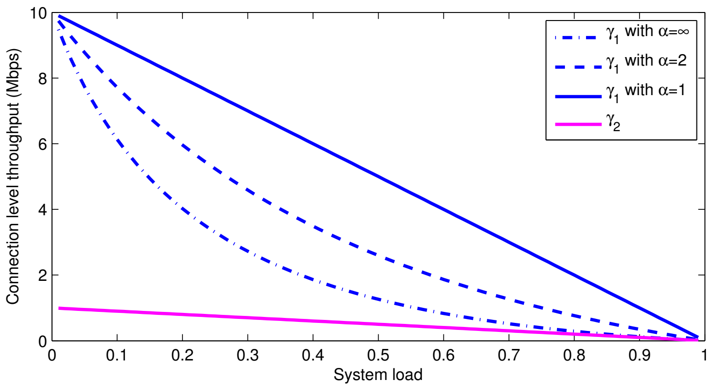

FIGURE 10.3: CONNECTION LEVEL THROUGHPUTS, 2 CLASSES, $C_1 = 10$, $C_2 = 1$ MBPS. FOR DIFFERENT FAIRNESS NOTIONS UNDER PRICE SCALING. CLASS 2 THROUGHPUT IS ONLY PLOTTED ONCE SINCE RESULTS ARE SIMILAR IN ALL THREE CASES.

This allocation has the nice property that connection level throughputs become proportional to the lower layer offered rate, the proportionality constant being the *slowdown* of the processor sharing queue, which only depends on the cell *total* load. In particular, the slowdown of each class with respect to its own effective data rate is homogeneous among classes, which suggests this is a better way of allocating resources.

The case of current TCP Reno-like algorithms will be an intermediate one, corresponding to $\alpha = 2$. In Figure 10.3 we compare the connection level throughputs of a cell with two allowed rates as in Example 8.1, namely $C_1 = 10$ and $C_2 = 1$ Mbps, in the case of max-min, proportional fair and Reno-like allocations.

In Chapter 11 we will specialize these results to evaluate the connection level throughputs obtained by TCP over an IEEE 802.11 cell.

11
Analysis of IEEE 802.11

In this chapter, we apply the previous models to quantify the behavior of TCP over an IEEE 802.11 MAC layer. We begin by calculating the effective data rates 802.11 provides to TCP connections, and then we proceed to analyze through simulation several examples.

11.1 Calculating the effective rates: the impact of overheads.

In the models derived in the preceding sections, an important quantity is C_i, the effective data rate at which TCP packets from a given user i are served by the underlying layers. When the AP wants to transmit a packet to user i of length L at a physical layer modulation rate PHY_i using 802.11, it must comply with a series of backoff and waiting times, as well as headers included by the PHY layer.

This means that the MAC layer offers a service to the upper layer consisting of a transmission rate $C_i \leqslant PHY_i$. This has been analyzed before [Bianchi, 2000, Kumar et al., 2007] and we will recall and extend this analysis here.

Every time a packet is intended to be transmitted in 802.11, the DCF algorithm for medium sharing comes into play. In particular, every transmission consists of a *carrier sense* phase, where the medium is sensed to determine whether other stations are transmitting. This is called the $DIFS$ time. After that, the station performs a backoff phase consisting of a random number of time slots, of duration σ. The number of time slots is uniformly chosen between

0 and CW, the contention window parameter. After that, a physical layer header is added, known as the PLCP header, which specifies the physical layer modulation rate to be used subsequently. Then the data is put into the medium at rate PHY_i. After transmission, the station waits for a $SIFS$ time to receive the corresponding acknowledgment, which in turn needs a MAC_ACK time to be transmitted.

Considering all of the above phases, the time it takes to send this packet has a fixed component given by

$$T_i^0 := DIFS + H + \frac{L}{PHY_i} + SIFS + MAC\_ACK_i, \tag{11.1}$$

that includes the time in the air and all overheads, plus a random number of time slots $K\sigma$, where $K \sim U\{0,\ldots,CW\}$. In Table 11.1 we show typical values of these parameters for 802.11g.

Parameter	**Value**
Slot time σ	$9\mu s$
$SIFS$	$10\mu s$
$DIFS$	$28\mu s$
PLCP Header H	$28\mu s$
PHY_i	6Mbps … 54Mbps
CW_{min}	15 slots
MAC_ACK	$50\mu s$

TABLE 11.1: IEEE 802.11G PARAMETERS

We are interested in the average rate obtained by a station to study the upper layer effective rate. Observing that each packet is treated independently, the transmission times of successive packets form a renewal process, and the renewal reward theorem [Feller, 1965] tells us that in the long range the average rate is:

$$C_i^0 = \frac{L}{EK\sigma + T_i^0} = \frac{L}{\frac{CW_{min}}{2}\sigma + T_i^0}, \tag{11.2}$$

where we substituted K for its mean. We also took $CW = CW_{min}$ since we are modeling downlink traffic from the AP, which does not collide with itself. We also assume the appropriate PHY_i has been used so that one can neglect packet transmission errors. The denominator of the preceding expression (mean total time) is denoted by T_i.

In Table 11.2 we show the corresponding MAC level rates C_i^0 for the different PHY rates allowed in 802.11g with parameters as in Table 11.1. Note the impact of overheads in the highest modulation rates: this is due mainly to the fact that physical and MAC layer overheads are fixed in time, independent of the modulation rate PHY_i chosen for the data. This implies that higher modulation rates can finish the data part of the packet more quickly, but they still have to send the fixed length headers and wait for the backoff slots.

When TCP connections are taken into account, another overhead must be considered: the TCP ACK packet. These packets were designed to have low impact on the reverse path, by having a length of 40 bytes. However, due to the overheads added by the MAC layer, the TCP ACK becomes non negligible, specially at high modulation speeds. We assume that one TCP ACK is sent in the uplink direction for every TCP packet sent downlink. We will also assume that collision probabilities are low between downlink packets and the TCP ACKs. Under these assumptions, the TCP ACK packet introduces another overhead time in the system. The effective data rate then becomes:

$$C_i = \frac{L}{T_i + TCP\_ACK_i} \tag{11.3}$$

where TCP_ACK_i is the average time to transmit a TCP_ACK packet and is given by:

$$TCP\_ACK_i := DIFS + \frac{CW_{min}}{2}\sigma + H + \frac{L_{ack}}{PHY_i} + SIFS + MAC\_ACK_i \tag{11.4}$$

where L_{ack} is typically 40 bytes. These effective data rates C_i are also shown in Table 11.2. Again, note the strong impact of the TCP ACKs in the performance of the protocol at high modulation rates, due to the fact that the lower layer protocol overheads are fixed in time.

PHY rates	**MAC rate** (C_i^0)	**Eff. rate** (C_i)	**Measured rate**
54	28.6	19.5	19.7
48	26.8	18.6	18.6
36	22.4	16.3	16.2
24	16.9	13.1	12.8
18	13.6	11.0	10.6
12	9.78	8.25	8.0
6	5.30	4.74	4.6

TABLE 11.2: MAC AND EFFECTIVE DATA RATES FOR THE CORRESPONDING PHY RATES OF 802.11G, WITH A PACKET SIZE OF $L = 1500$ BYTES. VALUES ARE IN MBPS. THE MEASURED RATES ARE ESTIMATED WITHIN $0.1 Mbps$ OF ERROR.

To validate the above expressions, we simulated in ns-2 [McCanne and Floyd, 2000] several independent replications of a long TCP connection in a single wireless hop scenario. The average throughput for each PHY rate is reported in the last column of Table 11.2, showing good fit with the predicted values.

In the following, we shall not try to modify the impact of overheads and consider them given, since they are included in the standards. For the purpose of modelling, we will use the

C_i values of Table 11.2, as the effective data rates provided by 802.11 to TCP connections, and use this values for the algorithm implementations.

11.2 Multirate RED implementation

As we discussed in Chapter 9, the price to which a TCP connection should react in order to attain the equilibrium of Problem 9.1 is the queueing delay. However, this price should be scaled by the effective data rate C_i the connection experiments in each link. Clearly, this is difficult to implement without resorting to scheduling. Moreover, typical TCP connections use loss based congestion control mechanisms, such as TCP-Reno. Therefore, we propose to use the MRED algorithm developed in Section 9.2 at each node to attain the optimum of Problem 9.1.

To test the proposal in a real environment, we implemented this algorithm in the Network Simulator `ns-2`. Our implementation is based on the library `dei80211mr` [Baldo et al., 2007]. Two important extensions were made to the library: the existing ARF mechanism was updated to cope with the possibility of a single node having different modulation rates for different destinations, which reflects the real behavior of current APs. The second modification was to implement the Multirate RED (MRED) queue, where the described early packet discard takes place.

Note that the cross-layer information needed for implementation of the mechanism is minimal: whenever a packet for next-hop j is received, it is discarded with probability $p_j = \kappa b/C_j$ where κ acts as a scaling parameter, b is the current queue length, and C_j is the corresponding effective rate for the current modulation rate the AP maintains with destination j (as in Table 11.2). In the case of wired links, the link capacity is used to scale this drop probability. The non-dropped packets are served then on a FIFO basis.

11.3 Simulation examples

We now present several simulation scenarios to illustrate the behavior of the proposed algorithm. First we compare the different allocations in a single-cell setting, and show the improvement in efficiency obtained by introducing the price scaling algorithm. Then we analyze the resource allocation in a setting where connections have different RTTs, and thus different utility functions, in order to show that the models introduced in Chapters 8 and 9 also capture the behavior in this situation. Then we validate and compare the connection level throughputs obtained under the different allocations. Finally, we present a case where the network is more complex, and the techniques introduced in Section 9.3 are validated.

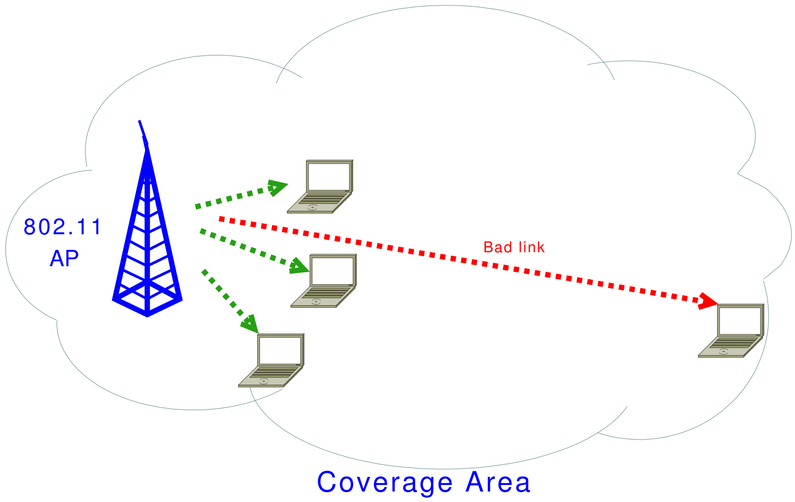

FIGURE 11.1: TOPOLOGY OF A SINGLE-CELL SCENARIO.

11.3.1 Single-cell scenario

We simulate the topology shown in Figure 11.1, which consists of a single cell 802.11g scenario in which 3 users are connected with a modulation rate $PHY_i = 54Mbps$, and some time later, a fourth user is added at the lowest possible modulation $PHY_4 = 6Mbps$. All four connections use TCP-Newreno and share equal Round Trip Times (RTTs), then having similar utility functions.

Results are shown in Figure 11.2. We simulated 50 independent replications of the experiment, and the average throughput for each connection type as well as 95% confidence intervals are shown. For these modulation rates, the effective data rates according to Table 11.2 are $C_i = 19.5Mbps$, $i = 1,2,3$ and $C_4 = 4.74Mbps$. In the first graph of Figure 11.2, we see that initially the fast users are sharing efficiently the total throughput. After the introduction of the slow user, all connections converge to the same throughput, which is approximately $x^* = 2.74Mbps$, the harmonic mean discussed in Proposition 8.1. In the second graph, we show the behavior of the system under the MRED algorithm. In this case, after the introduction of the slow user, the allocation converges approximately to $x_i^* = 4.2Mbps$, $i = 1,2,3$ and $x_4^* = 2.1Mbps$, which is the exact solution of Problem 9.1. Note that the total throughput in the network is increased by more than 30%.

11.3.2 Different RTTs scenario

The purpose of this example is to show that Problem 8.2 captures the behavior of the system when the TCP connections have different RTTs, and thus different utilities, and to show how

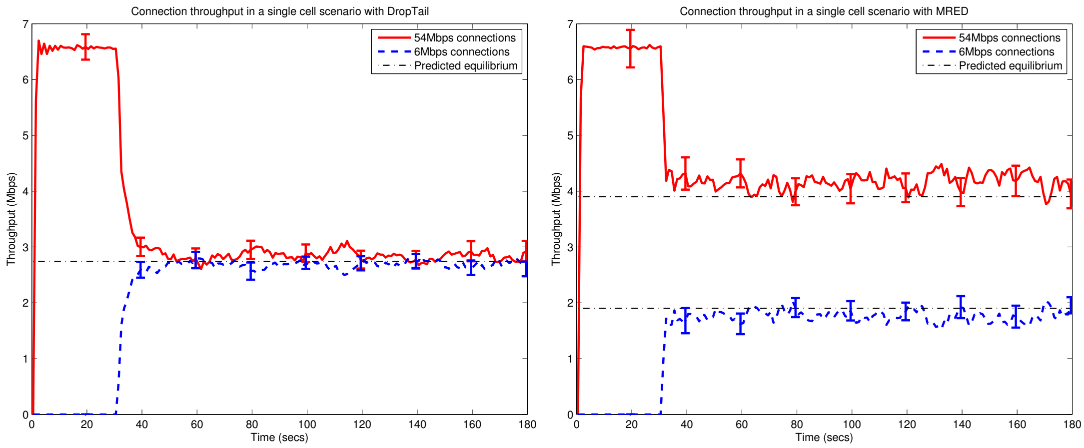

FIGURE 11.2: COMPARISON BETWEEN THROUGHPUTS: WITHOUT MRED (ABOVE), WITH MRED (BELOW).

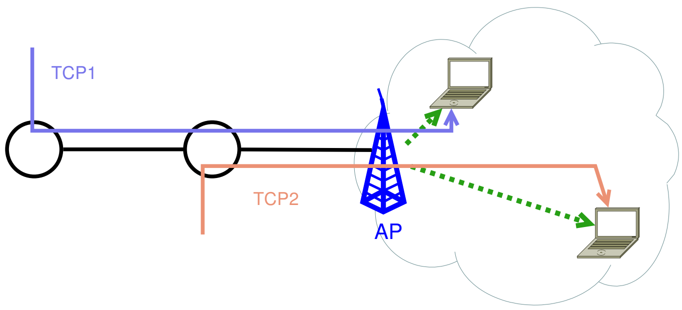

FIGURE 11.3: WIRED-WIRELESS TOPOLOGY.

efficiency can also be improved in this case with the MRED algorithm.

We consider the topology of Figure 11.3, where two connections with different RTTs share a wireless bottleneck link. In this example, connection 1 has a longer RTT than that of connection 2, and its station is closer to the AP, having a modulation rate $PHY_1 = 54Mbps$. The second connection has a modulation rate of $PHY_2 = 6Mbps$. Both connections use TCP-Newreno, which we model by the utility function $U(x) = -1/(\tau^2 x)$ with τ the connection RTT.

Plugging these values into Problem 9.1 using the effective data rates of Table 11.2, the allocation results are $x_1^* = 2.43Mbps$ and $x_2^* = 4.14Mbps$. In the first graph of Figure 11.4 we show the results of 50 independent replications of the experiment, which shows that indeed the connection throughputs converge approximately to the values predicted by Problem 8.2.

By using MRED in the AP we can change the allocation to the one proposed in Problem 9.1, removing the bias of Problem 8.2. The resulting allocation is $x_1^* = 4.38$ and $x_2^* = 3.67$. In the second graph of Figure 11.4 we show the corresponding simulation results. We see that the

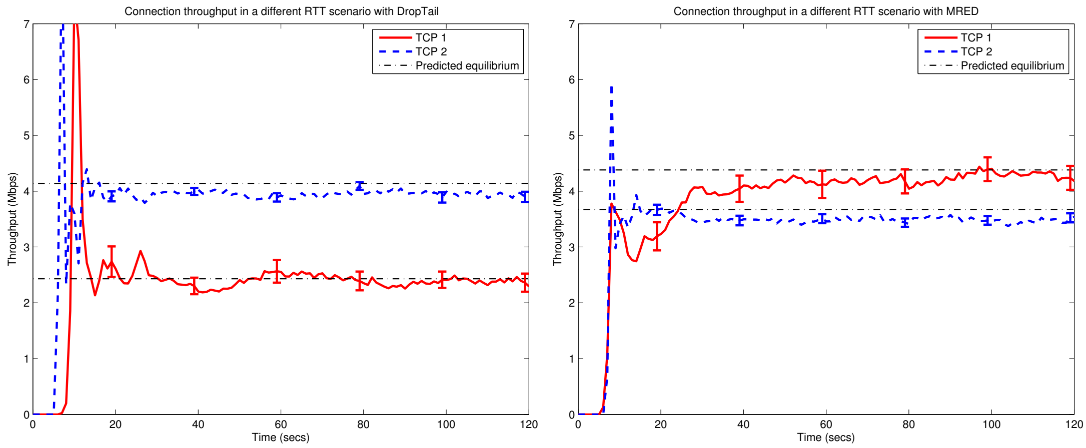

FIGURE 11.4: WIRED-WIRELESS TOPOLOGY SIMULATION. ABOVE: ORIGINAL ALLOCATION. BELOW: MRED ALGORITHM.

MRED algorithm approximately drives the system to the new equilibrium. Note that this new equilibrium is also more efficient.

11.3.3 IEEE 802.11 connection level throughputs

We now apply the results of Chapter 10 to evaluate the TCP connection level performance when working with a random workload and an underlying IEEE 802.11 MAC layer. As before, we focus on a downlink scenario where all TCP connections arrive at the coverage zone at a random point, with total arrival rate Λ_j for the rate C_j. These rates are chosen as in Table 11.2, which are valid for 802.11g.

As a first example, we consider a single cell scenario with two PHY rates. Users near the cell establish connections at the highest possible modulation rate of $54Mbps$ and the remaining users use the lowest possible modulation of $6Mbps$. We simulated the random arrival of connections in ns-2 and measured the connection level throughput for different values of the cell load. To take into account the fact that the low modulation rate has a greater coverage area, the arrival intensities were chosen proportional to the size of the coverage areas.

In Figure 11.5 we plot the connection level throughputs obtained by simulation and the predicted throughputs using the results of Chapter 10 for different values of the total cell load ϱ. The first graph shows the connection level throughputs when the proposed Multirate RED is not in use, and the second one shows the results for a cell using Multirate RED. In both cases results show a good fit with the model.

In each case, the connection level throughputs start at the effective data rate for each class, C_i, corresponding to the case where each connection arrives to an empty network, and thus is

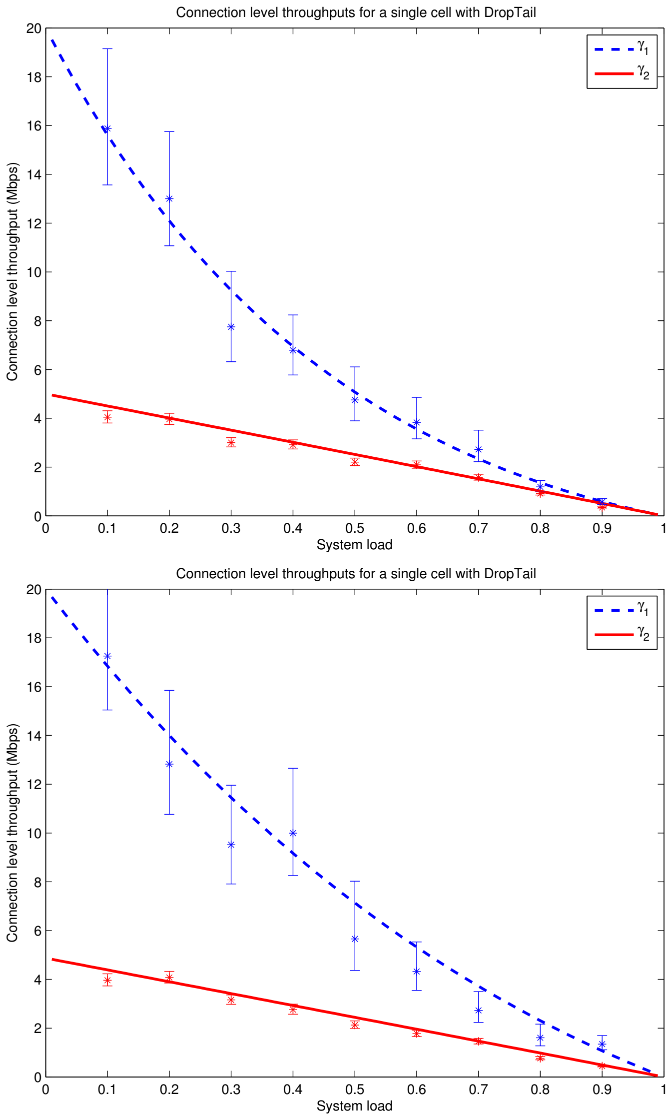

FIGURE 11.5: CONNECTION LEVEL THROUGHPUTS FOR AN IEEE 802.11G CELL WITH TWO MODULATION RATES. ABOVE: WITHOUT MRED, BELOW: WITH MRED IN USE.

able to obtain its full rate. When the offered load ϱ begins to grow the throughputs go to zero, as expected. We observe that in the case where MRED is not in use, the high modulation rate users are more heavily slowed down, and throughputs tend to equalize.

When all the data rates are allowed, the connection level throughputs can be calculated by solving the linear system of equations discussed in Section 10.2. In Figure 11.6 we plot the predicted connection level throughputs for such a setting, with increasing cell load. Again, the arrival rates for each class are chosen proportional to the estimated coverage areas. In the first graph, we show the results for current 802.11g cells, where price scaling is not used, and thus

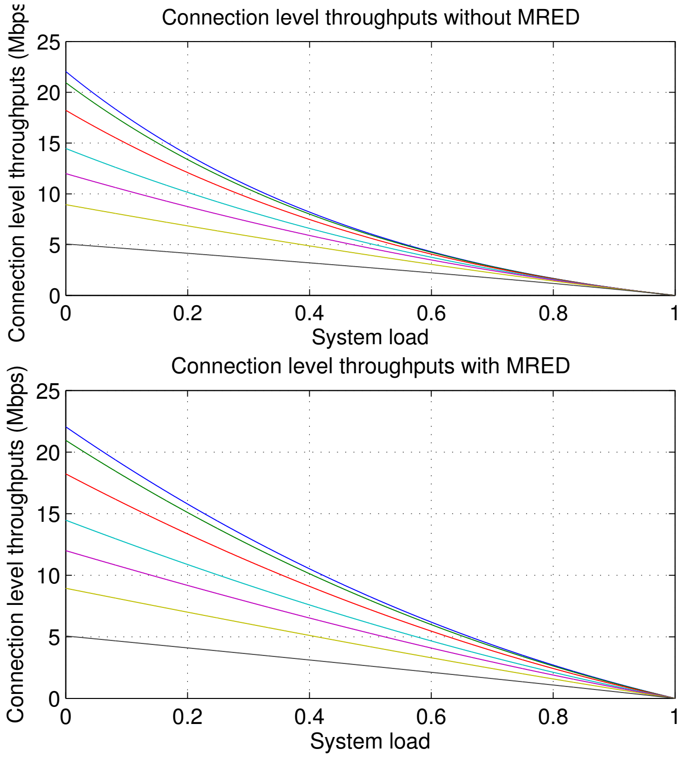

FIGURE 11.6: CONNECTION LEVEL THROUGHPUTS FOR AN IEEE 802.11G CELL. ABOVE: WITHOUT PRICE SCALING, BELOW: WITH MRED. EACH LINE CORRESPONDS TO A DIFFERENT PHY RATE IN DECREASING ORDER.

penalizing the higher rates. In the second graph we show the connection level throughputs under the price scaling mechanism, removing the bias. We can see that the higher rates get better throughput in all cell loads, while the lower ones are mostly unaffected by the change.

To evaluate the difference between the two mechanisms, in Figure 11.7 we plot the ratio between the higher and lower throughputs for three different situations: without MRED, with MRED in use, and the theoretical proportional fair situation. We can see that applying MRED with current TCP implementations gives an intermediate situation, improving on the bias against higher throughputs with respect to the current situation.

11.3.4 An application to a mixed wired-wireless tree topology

In this example, we simulate the topology of Example 9.5. This topology is illustrated in Figure 11.8 with two distribution APs and two users in each AP. The access link capacity is $c_{access} = 20Mbps$ representing a typical access capacity (e.g. a DSL line). The distribution links have $c_{dist} = 100Mbps$ and thus are overprovisioned. The wireless cells are identical and have two users each, with modulation rates $PHY_1 = PHY_3 = 54Mbps$ and $PHY_2 = PHY_4 = 6Mbps$.

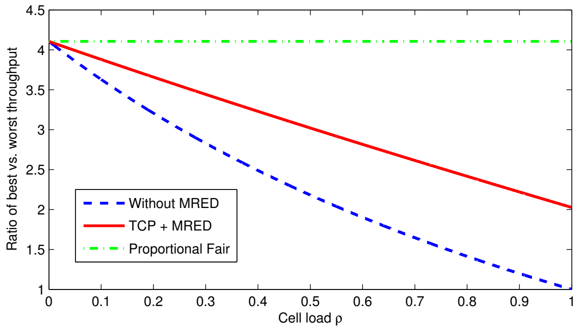

FIGURE 11.7: RATIO OF BEST VS. WORST THROUGHPUT FOR AN IEEE 802.11G CELL.

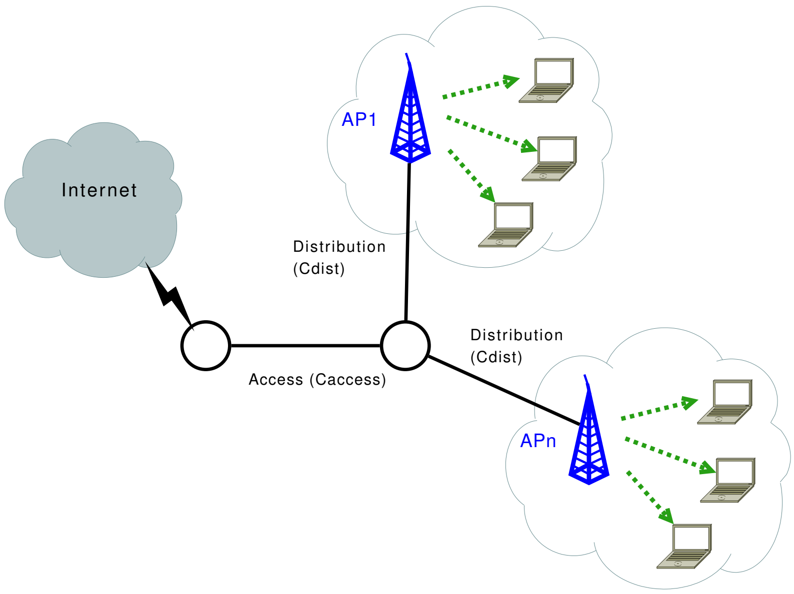

FIGURE 11.8: TOPOLOGY OF A MIXED WIRED-WIRELESS ACCESS NETWORK.

Each user has a single TCP connection and all connections have equal RTTs.

Plugging these values into Problem 9.2 gives the following allocation:

$$x_1^* = x_3^* = 6.4Mbps \quad x_2^* = x_4^* = 3.2Mbps$$

Note in particular that both the access link and the wireless cells are saturated in the resulting allocation. This is a difference with typical wired-only models with tree topologies. In particular, in this case, there is a positive price (queueing delay) both at the APs and the wired access link.

By using the MRED algorithm as discussed in Section 9.2, we can drive the system to this allocation. Results are shown in Figure 11.9, where again 50 independent replications were performed, and the average throughputs as well as 95% confidence intervals are shown. We

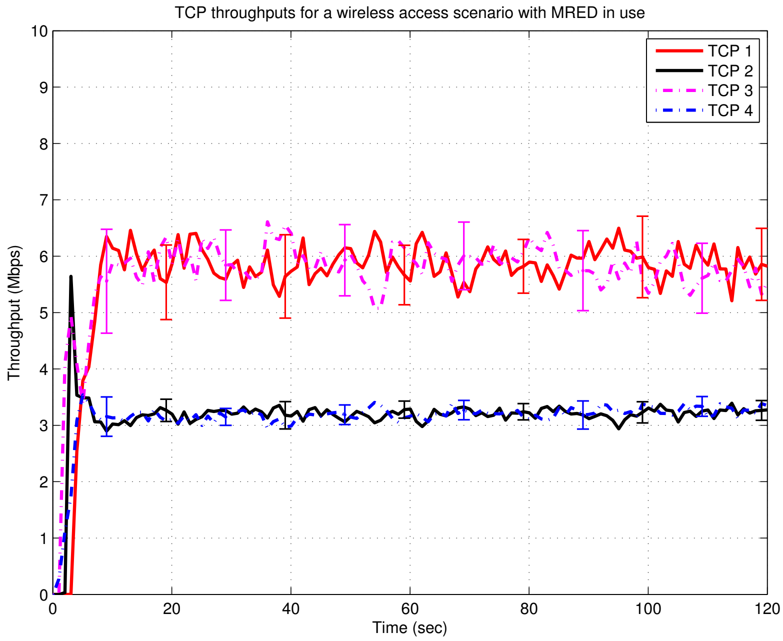

FIGURE 11.9: THROUGHPUTS OF TCP CONNECTIONS FOR A WIRELESS ACCESS SCENARIO WITH 4 USERS. MRED IS IN USE.

see that the throughputs approximately converge to the above equilibrium.

Note also that, if we choose not to use the MRED algorithm, the allocation would be given by the solution of Problem 8.2, which is $x_i^* \approx 3.8 Mbps$ for each user. In that case, the full access capacity would not be used.

12
Conclusions of Part II

In this part of the thesis, we applied the Network Utility Maximization framework to characterize the cross-layer interaction between the TCP transport protocol with an underlying MAC where multiple modulation rates coexist. This situation is present in typical IEEE 802.11 deployment scenarios. We described the resource allocation imposed by current wireless networks in this framework, showing that a bias is imposed against users of high modulation rates.

We then proposed an alternative resource allocation that generalizes the fairness and efficiency notions of TCP in wired networks to this context, and overcomes the inefficiencies of current protocols. We developed a simple mechanism to impose these more efficient equilibria in single cell scenarios and generalizations of this procedure to more complex topologies.

We also showed how the connection level dynamics can be analyzed through a Markov process, which can be identified in some cases with a well known queue. This enables us to characterize the stability regions and connection-level throughput obtained by the current resource allocation and our proposed alternative.

Finally we applied the previous results to the IEEE 802.11 MAC layer, establishing the effective data rates and validating the results by simulations.

Several lines of work remain open. An important one is how to extend the proposed mechanisms to the new 802.11 additions, where these issues become more important due to the higher data rates involved, and the use of packet aggregation. We also would like to study the performance of a system where the uplink traffic is not negligible, as is the case in several usage models of today, and collisions have to be taken into account.

13

General conclusions and future lines of work

Throughout this thesis, we have analyzed several problems of network resource allocation, under the framework of Network Utility Maximization. The work mainly focused on analyzing the connection level performance, efficiency and fairness of several models of network resource allocation, in several settings, both single and multi-path, and both wired and wireless scenarios.

We have analyzed in detail two important problems: on one hand the resource allocation provided by congestion control protocols whenever multiple connections per user are allowed. We did so in the context of wired networks, and we identified problems with the current resource allocation algorithms. In particular, we showed that individual greedy users can cheat on the resource allocation by opening multiple connections. We therefore introduced a new paradigm of resource allocation focused on the aggregate rates allocated to each user, which we called *user-centric fairness*. We then showed how simple and decentralized algorithms can be introduced into the network in order to drive the system to a suitable efficient and fair operating point. The algorithms rely on the control of the number of connections using the congestion prices already provided by the network. This control can be either done by the user, or as an admission control by the network. We modelled the two situation and provided stability results for the resulting systems. Also, the algorithms developed were tested with packet level simulations, validating our results, and providing evidence that real network implementations are possible.

The second important problem analyzed here is the resource allocation provided by congestion control algorithms over a physical layer that allows multiple transmission rates, which is a common situation when wireless networks are involved. We showed that the typical algorithms introduce undesired inefficiencies. We proposed a new resource allocation model, and the corresponding algorithms to impose this allocation in a single wireless cell scenario. We also analyzed how to extend these algorithms to more complicated topologies, in particular mixed wired-wireless network topologies common in practice. We showed stability results for the proposed algorithms, and applied them to the particular case of the IEEE 802.11 standard, which is relevant since it is one of the most widely used wireless technologies. Again, packet level simulations were provided, showing that the algorithms can be applied in practice.

The research relied on several mathematical tools. The most important were convex optimization, non-linear and linear control techniques, Lyapunov theory, Markov chain modelling and fluid limits. We have tried to keep balance between the mathematical models and implementation. In all of the algorithms, the development was always with a practical view in mind. sometimes discarding more complex algorithms for the sake of implementation. However, we have analyzed the performance of all the control algorithms proposed, in particular we provided rigorous proofs of their local or global asymptotic stability. In the case where we were unable to obtain these results, we have tested them via simulations.

Several lines of future research remain open. For the results on the first part of the thesis, the main problems are related to implementation issues. We have provided a theoretical analysis of the algorithms, and through packet level simulations, a proof of concept that these algorithms can be implemented in real networks. In future research, developing these algorithms in order to put them in production-level networks is more than a trivial task. In particular, developing suitable mechanisms to communicate prices between connections and the routers, in order to ease implementation of admission control algorithms.

For the wireless networks analyzed in the second part, it would be nice to extend the results presented here to the upcoming wireless standards, where new features appear. In particular, packet aggregation algorithms are introduced to lower the impact of having multiple rates in the cell. However, the results of this work provide evidence that, if not done properly, inefficiencies can nevertheless appear. Further study on this subject is required, which can lead to new algorithms that enhance performance in these networks.

A

Mathematical preliminaries

In this Appendix, we provide a brief review of some of the mathematical concepts that are needed to develop the results presented in this Thesis.

A.1 Convex optimization

We present below the main concepts of convex optimization. A good introduction to this subject is [Boyd and Vanderberghe, 2004]. We start by some definitions.

Definition A.1 (Convex set)**.** *A set $C \subset \mathbb{R}^n$ is called* convex *if the following holds:*

$$\alpha x + (1-\alpha)y \in C \quad \forall x, y \in C, 0 \leqslant \alpha \leqslant 1.$$

The above definition states that for any given two points in C, the line segment joining both points is entirely contained in the set. Convex sets form the basic regions where convex optimization is performed.

Definition A.2 (Convex function)**.** *A function $f : C \to \mathbb{R}$, where $C \subset \mathbb{R}^n$ is a convex set, is said to be* convex *if and only if:*

$$f(\alpha x + (1-\alpha)y) \leqslant \alpha f(x) + (1-\alpha) f(y) \quad \forall x, y \in C, 0 \leqslant \alpha \leqslant 1.$$

The function is said to be strictly convex *if the above inequality is strict whenever $x \neq y$ and $\alpha \in (0,1)$.*

Definition A.3 (Concave function)**.** *A function $f : C \to \mathbb{R}$, where $C \subset \mathbb{R}^n$ is a convex set, is said to be* concave *if and only if $-f$ is convex. Moreover, it is* strictly concave *if $-f$ is strictly convex.*

Convex functions can be visualized as a "bowl" shape: for any two points $(x, f(x))$ and $(y, f(y))$ on the graph of f, the line segment joining the two points stays above the graph of the function. Conversely, concave functions have the shape of an inverted bowl, with the graph of the function being above the line segments joining its points. Linear or affine functions are limit cases, and are both concave and convex, though clearly not in the strict sense.

In the development of the theory, we shall frequently encounter *convex optimization* problems, where we want to maximize a concave function subject to convex constraints, i.e. finding the optimum of a concave function over a convex set. The following Lemma is useful:

Lemma A.1. *Consider a concave function f defined over a convex set $C \subset \mathbb{R}^n$. Then the following holds:*

- *If x^* is a local maximum of f in C, then it is also a global maximum over C. If moreover f is strictly concave, then x^* is the unique global maximum on C.*
- *If C is compact (closed and bounded), then a global maximum exists over C.*

Therefore, convex optimization problems of this form are well defined, and a solution always exists provided that the constraint set is compact. The remaining question is how we can find this optimum. From now on, we shall focus on one class of convex optimization problems that we shall find frequently in our analysis.

Consider a concave function f, that we want to optimize, subject to *linear* constraints. The problem is the following:

Problem A.1.

$$\max_x f(x)$$

subject to the constraints:

$$Rx \leqslant c.$$

where the inequality above is interpreted componentwise.

We will assume that f is differentiable, and R is a matrix of appropriate dimensions. It is easy to see that the *feasible set*, i.e. the points x that satisfy the constraints form a convex set. The first step towards solving such a problem is to form the *Lagrangian function*, given by:

$$\mathscr{L}(x, p) = f(x) - p^T(Rx - c). \tag{A.1}$$

Here, $p = (p_l)$ is a vector, whose entries are called Lagrange multipliers, with the only constraint being $p \geqslant 0$. The superscript ^{T} denotes matrix transpose.

One of the main results on convex optimization, that we shall use frequently in our work, is the following:

Theorem A.1 (Karush-Kuhn-Tucker conditions)**.** *Consider Problem A.1, with Lagrangian given by* (A.1)*. Then, the optimum of the problem must satisfy the Karush-Kuhn-Tucker (KKT) conditions:*

$$\frac{\partial \mathscr{L}}{\partial x} = \nabla f(x) - R^T p = 0, \tag{A.2a}$$

$$p^T(Rx - c) = 0, \tag{A.2b}$$

$$Rx \leqslant c, \tag{A.2c}$$

$$p \geqslant 0. \tag{A.2d}$$

The proof of this Theorem can be found in [Boyd and Vanderberghe, 2004]. For a concave optimization problem with linear constraints, equations (A.2) completely characterize the solution. Note that equation (A.2a) is a gradient condition on the Lagrangian, where the solution must correspond to a stationary point. The second condition, given in (A.2b), is called the *complementary slackness* condition. It states that, if the solution is achieved with strict inequality in a given constraint, the corresponding Lagrange multiplier is 0. In this case the constraint is called *inactive*. If on the other hand, the solution is achieved in with equality in the constraint, the corresponding Lagrange multiplier may be strictly positive. An interpretation of this condition is given in Section 2.1 in terms of *prices*.

In the text, we refer often to the KKT conditions, and typically we impose (A.2a) and (A.2b), always assuming the remaining conditions as implicit in the statement.

The Lagrangian and the KKT conditions are part of a more extensive theory of convex optimization, which is *Lagrangian duality*. In this theory, convex optimization problems are analyzed through the *dual function* given by:

$$D(p) = \max_{x \in C} \mathscr{L}(x, p),$$

where C is the set defined by the constraints of Problem A.1. It is easy to see that, if x^* is a solution of A.1, then it must satisfy:

$$f(x^*) \leqslant \inf_{p \geqslant 0} D(p).$$

To see this, note that for any $x \in C$ and $p \geqslant 0$ we have:

$$f(x) \leqslant f(x) - p^T(Rx - C),$$

and thus:

$$\max_{x \in C} f(x) \leqslant \max_{x \in C} f(x) - p^T(Rx - C) = \max_{x \in C} \mathscr{L}(x, p) = D(p).$$

Since the left hand side does not depend on p, we conclude that:

$$\max_{x \in C} f(x) \leqslant \inf_{p \geqslant 0} D(p).$$

The left hand side is called the *primal* optimization problem. The right hand side defines what is called the *dual* optimization problem. Note that the dual problem is always convex and performed over the Lagrange multipliers. The above is called the *weak duality* theorem, that states that a solution of the dual problem is always an upper bound of the primal optimum. With additional hypotheses, called *Slater constraint qualification* conditions, the above inequality becomes an equality (equivalently, it is said that there is *zero duality-gap*). Slater qualification conditions are always satisfied when the objective function is concave and the constraints are linear. Thus, for Problem A.1, we will always have zero duality gap.

A.2 Lyapunov stability

We now present some results on Lyapunov stability theory that are used in several parts of this Thesis. We will summarize only the basic results, and refer the reader to [Khalil, 1996] for further developments.

Consider a dynamical system defined by the following initial value problem of a differential equation:

$$\dot{x} = f(x), \tag{A.3a}$$

$$x(t_0) = x_0. \tag{A.3b}$$

Here, $f : \mathbb{R}^n \to \mathbb{R}^n$ defines the dynamics of the system, and we assume that f satisfies the Lipschitz condition: for any given $x \in \mathbb{R}^n$ there exists a constant K_x such that:

$$\|f(y) - f(x)\| \leqslant K_x \|y - x\| \quad \forall y \in \mathbb{R}^n.$$

Here, $\|\cdot\|$ represents a suitable norm in $\mathbb{R}^n$.

With the above condition, the initial value problem defined by (A.3) has a well defined solution over an interval of $\mathbb{R}$ containing t_0. We are interested in the equilibrium solutions of this system, i.e. solutions of the form:

$$x(t) \equiv x_0 \quad \forall t.$$

For the system (A.3) to be in equilibrium, it is necessary that x is a solution of:

$$f(x) = 0.$$

The existence of equilibrium trajectories is of little practical importance, since this equilibrium cannot be observed in practice if they are not *stable*, in the sense that small perturbations of the system in equilibrium do not produce drastic departures from the equilibrium behavior. Below we define several important notions of stability. Without loss of generality, we shall assume that the coordinate system is chosen such that the solution of $f(x) = 0$ under study is at $x = 0$.

Definition A.4 (Stability)**.** *The equilibrium point is said to be* stable *if, given any* $\epsilon > 0$*, there exists* $\delta > 0$*, such that*

$$\|x(t)\| < \epsilon, \forall t \geqslant t_0,$$

whenever:

$$\|x_0\| < \delta.$$

Definition A.5 (Asymptotic stability)**.** *The equilibrium point is said to be* (locally) asymptotically stable *if it is stable and there exists a* $\delta > 0$*, such that*

$$\lim_{t \to \infty} \|x(t)\| = 0,$$

whenever:

$$\|x_0\| < \delta.$$

Definition A.6 (Global asymptotic stability)**.** *The equilibrium point is said to be* globally asymptotically stable *if the limit:*

$$\lim_{t \to \infty} \|x(t)\| = 0,$$

holds for any initial condition x_0*.*

Intuitively, stability implies that if the initial condition is perturbed a small amount from the equilibrium point, the trajectory still stays near this equilibrium point in the future. The asymptotic stability condition is stronger, and implies that when the perturbation is small, the trajectory will converge to the equilibrium point. Thus the equilibrium point acts as a local attractor of the system. The global stability condition states that the equilibrium point is reached eventually starting from any initial condition for the system.

The stability analysis of a given dynamical system can be performed in several ways. The first one is through linearization. Consider a linear dynamical system of the form (A.3). In this case, the dynamics can be rewritten as:

$$\dot{x} = Ax, \tag{A.4a}$$

$$x(t_0) = x_0, \tag{A.4b}$$

where A is a $n \times n$ matrix. In this case, the solution is given by:

$$x(t) = e^{A(t-t_0)} x_0 \quad \forall t \in \mathbb{R}^n,$$

where e^M is the exponential matrix. A direct analysis of this matrix leads to the following proposition:

Proposition A.1. *If all the eigenvalues of A have negative real parts, then the origin is a globally asymptotically stable equilibrium for the system* (A.4).

A matrix A that satisfies the above condition, i.e. its eigenvalues have strictly negative real parts, is called a *Hurwitz matrix*.

When the system is non-linear, but it can be suitably approximated by a linear system around the equilibrium, local asymptotic stability can still be obtained:

Proposition A.2. *Consider a dynamical system of the form* (A.3), *where f is continuously differentiable at the equilibrium $x = 0$. If the Jacobian matrix:*

$$A = \frac{\partial f}{\partial x}$$

is Hurwitz, then the origin is locally asymptotically stable.

The main idea is that, locally, the system can be approximated by the linear system given by $\dot{x} = Ax$, with A the jacobian matrix.

Stronger stability results for non linear systems are often achieved via *Lyapunov functions*. The Lyapunov theory of stability provides us with a sufficient condition based on a suitable chosen function that acts as an energy function for the system. The main result is the following:

Theorem A.2 (Lyapunov, 1892)**.** *Consider a continuously differentiable* positive definite *function $V(x)$, i.e. such that:*

$$V(x) > 0, \quad \text{if } x \neq 0,$$

and $V(0) = 0$.

Define the derivative along the trajectories of the system as:

$$\dot{V}(x) = \frac{d}{dt} V(x(t)) = \nabla V \cdot f(x).$$

Then the following conditions for stability hold:

1. *If $\dot{V}(x) \leqslant 0$, then the origin is a stable equilibrium point for the system.*

2. *If in addition, $\dot{V}(x) < 0 \ \forall x \neq 0$, then the equilibrium point is asymptotically stable.*

3. *If in addition to 1. and 2., V is radially unbounded, i.e.:*

$$V(x) \to \infty \quad \text{whenever } ||x|| \to \infty,$$

then the origin is globally asymptotically stable.

The idea behind Lyapunov functions is that, once a solution enters the set $\{x : V(x) \leqslant c\}$, it cannot leave, since V decreases along trajectories. We say that the set $\{x : V(x) \leqslant c\}$ is *forward invariant* for the dynamics. Since V is continuous and $V(0) = 0$, we can choose a set near $x = 0$ such that the solution never leaves this region. If moreover $\dot{V} < 0$, we can enclose the solution in a sequence of such sets thus proving asymptotic stability. If moreover V is radially unbounded, for any initial condition the set $\{x : V(x) \leqslant V(x_0)\}$ is compact, and thus the solution cannot leave the set and must converge to the equilibrium.

An important refinement of Lyapunov's result along the previous ideas is due to LaSalle. We need the following:

Definition A.7. *A set M is said to be* invariant *with respect to the dynamics* (A.3) *if:*

$$x(t_0) \in M \Rightarrow x(t) \in M \quad \forall t \in \mathbb{R}.$$

We say that a set M is forward invariant if:

$$x(t_0) \in M \Rightarrow x(t) \in M \quad \forall t \geqslant t_0.$$

Equilibrium points are the most basic invariant sets. However, the solution of the dynamics can have *limit cycles* or even more complicated behavior. The relationship between invariant sets and Lyapunov functions is the following result:

Theorem A.3 (LaSalle Invariance Principle)**.** *Let $\Omega \subset \mathbb{R}^n$ be a compact set that is forward invariant for the dynamics* (A.3)*. Let $V : \Omega \to \mathbb{R}$ be a continuously differentiable function such that $\dot{V}(x) \leqslant 0$ in Ω. Let $E = \{x : \dot{V}(x) = 0\}$, and M be the largest invariant set in E. Then, every solution of* (A.3) *with $x_0 \in \Omega$ approaches the set M as $t \to \infty$.*

The notion of convergence to a set is simply that $d(x(t), M) \to 0$ when $t \to \infty$, where $d(x, M) = \inf_{y \in M} d(x, y)$. Note that the above result does not require V to be positive definite. It merely states that, if we can find a Lyapunov function for the system on a compact and forward invariant set, the solution must converge to the set where $\dot{V} = 0$, and moreover, it must converge to an invariant set in this region. Note also that the construction of Ω may not be tied to V. However, a typical choice is $\{x : V(x) \leqslant c\}$ for an appropriate value of c. This leads to a corollary of the above Theorem, originally due to Krasovskii:

Corollary A.1. *Let $x = 0$ be an equilibrium point of* (A.3)*, and $V : \mathbb{R}^n \to \mathbb{R}$ a continuously differentiable and radially unbounded positive definite function such that $\dot{V}(x) \leqslant 0$ for all $x \in \mathbb{R}^n$. Let $S = \{x : \dot{V}(x) = 0\}$, and note that $0 \in S$. If the only solution of* (A.3) *that can stay identically in S is $x = 0$, then the origin is globally asymptotically stable.*

Note that when $\dot{V}$ is negative definite, $S = \{0\}$ and this is a restatement of Lyapunov's result. However, if $\dot{V}$ is only negative semidefinite, we have a stronger result, provided that $x(t) \equiv 0$ is the only admissible solution with $\dot{V} \equiv 0$.

A.3 Passive dynamical systems

A useful approach in analyzing the stability of interconnected dynamical systems, related to the Lyapunov theory, is the *passivity approach.* Consider a dynamical system with input $u \in \mathbb{R}^m$, output $y \in \mathbb{R}^m$ and state $x \in \mathbb{R}^n$, given by:

$$\dot{x} = f(x,u), \tag{A.5a}$$

$$y = h(x,u). \tag{A.5b}$$

Here $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is locally Lipschitz and $h : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^m$ is continuous, and we assume that the equilibrium conditions are $f(0,0) = 0$, $h(0,0) = 0$. Note in particular that we assume the system has the same number of inputs and outputs.

Definition A.8 (Passive system)**.** *The system* (A.5) *is called* passive *if there exists a positive semidefinite differentiable function $V(x)$, which only depends on the state, satisfying:*

$$\dot{V}(x) = \nabla V \cdot f(x) \leqslant u^T y \quad \forall (x,u) \in \mathbb{R}^n \times \mathbb{R}^m.$$

V is called the storage function *of the system.*

Additionally, the system is called state strictly passive *if:*

$$\dot{V}(x) \leqslant u^T y - \rho\psi(x) \quad \forall (x,u) \in \mathbb{R}^n \times \mathbb{R}^m.$$

where $\rho > 0$ is a constant and $\psi(x)$ is a positive semidefinite function such that:

$$\psi(x(t)) \equiv 0 \Rightarrow x(t) \equiv 0,$$

for all solutions of (A.5) *with arbitrary $u(t)$. The term $\rho\psi(x)$ is called the* dissipation rate.

There are several other notions of strict passivity, however, we shall restrict ourselves to state strict passivity, and simply say that a system is strictly passive whenever it is state strictly passive.

A direct consequence of passivity is the stability of the origin when the input is $u \equiv 0$.

Lemma A.2. *Assume that the system* (A.5) *is passive, and $u \equiv 0$. Then the origin is a stable equilibrium point. If moreover the system is state strictly passive, the origin will be asymptotically stable. If additionally V is radially unbounded, the origin is globally asymptotically stable.*

The proof of the Lemma follows from Lyapunov arguments using V, the storage function, as a candidate Lyapunov function for the system.

The full power of the passivity approach, however, becomes evident when studying interconnected feedback systems. Let H_1 and H_2 be two systems of the form (A.5), in a negative feedback connection, depicted in Figure A.1.

Note that the output of a system is the input of the other. We have the following:

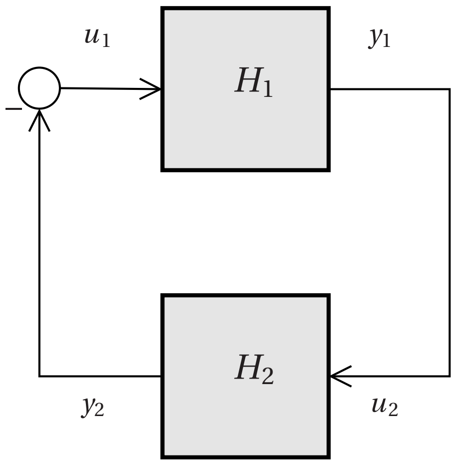

FIGURE A.1: NEGATIVE FEEDBACK INTERCONNECTION OF SYSTEMS.

Theorem A.4. *Consider the feedback system of Figure A.1, where H_i is given by:*

$$\dot{x}_i = f_i(x_i, u_i),$$
$$y_i = h(x_i, u_i),$$

for $i = 1, 2$. Assume that the feedback system has a well defined state space model:

$$\dot{x} = f(x, u),$$
$$y = h(x, u).$$

where $x = (x_1, x_2)^T$, $u = (u_1, u_2)^T$ and $y = (y_1, y_2)^T$, f is locally Lipschitz, h is continuous and $f(0,0) = h(0,0) = 0$.

If H_1 and H_2 are passive, then the origin is a stable point of the dynamics:

$$\dot{x} = f(x, 0)$$

If moreover, H_1 and H_2 are state strictly passive, then the origin is asymptotically stable. If additionally the storage functions V_i of each system are radially unbounded, then the origin is globally asymptotically stable.

The main idea for the proof is to use $V(x) = V_1(x_1) + V_2(x_2)$ as a Lyapunov function for the closed loop system. Finer results on asymptotic stability can also be obtained with other notions of passivity, or by means of the LaSalle invariance principle. For an application of these concepts to Internet congestion control, we refer the reader to [Wen and Arcak, 2003].

In the case of linear time invariant systems, a simple frequency domain condition can be obtained that enables us to check whether a system is (strictly) passive. Consider a linear input-output time invariant system, given by:

$$\dot{x} = Ax + Bu, \tag{A.6a}$$

$$y = Cx + Du, \tag{A.6b}$$

where A, B, C and D are matrices of appropriate dimensions. In the Laplace domain, such a system will have a transfer function matrix given by:

$$G(s) = C(sI - A)^{-1}B + D$$

We have the following definition:

Definition A.9 (Positive real transfer matrix)**.** *A $p \times p$ transfer function matrix $G(s)$ is called* positive real *if:*

- *All elements of $G(s)$ are analytic in* $\mathrm{Re}(s) > 0$.
- *Any imaginary pole of $G(s)$ is simple and has a positive semidefinite residue matrix.*
- *For all real ω such that $j\omega$ is not a pole of G, the matrix $G(j\omega) + G^T(-j\omega)$ is positive semidefinite.*

If moreover $G(s - \epsilon)$ is positive real for some $\epsilon > 0$, we say that G is strictly positive real.

Note that when $p = 1$ (single input-single output system) and G is Hurwitz, i.e. it has no poles in $\mathrm{Re}(s) \geqslant 0$, the definition above reduces to checking whether the Nyquist plot $G(j\omega)$ of the system satisfies $\mathrm{Re}(G(j\omega)) \geqslant 0$.

The relationship between positive real transfer functions and passivity is established by using the Kalman-Yakubovich-Popov Lemma (c.f. [Khalil, 1996]), that relates the existence of a suitable quadratic storage function V with the real positive condition. The result is the following:

Theorem A.5. *A linear time invariant system with a minimal realization transfer function $G(s)$ is passive if and only if $G(s)$ is positive real. Moreover, it is state strictly passive if and only if $G(s)$ is strictly positive real.*

Therefore, the (strict) passivity of a system can be checked through the frequency domain condition on its transfer function. Note that for single input single output systems this is extremely simple.

A.4 Markov chains

A useful mathematical concept for the modelling of telecommunication systems are Markov chain stochastic processes. We present now the main definitions and results of Markov Chain theory.

Let $(\Omega, \mathscr{F}, P)$ be a probability space. A *continuous time stochastic process* with *state space E* is a family of random variables $X(t,\omega)$, where:

$$X : \mathbb{R} \times \Omega \to E.$$

Here $X(t,\cdot)$ is a random variable in E and $X(\cdot,\omega)$ is a random function $t \mapsto X(t,\omega)$ called the trajectory of the process. The index parameter $t \in \mathbb{R}$ is often interpreted as time, and we call the random variable $X(t) = X(t,\cdot)$ the state of the process at time t. For simplicity, the explicit dependence on the experiment ω is dropped.

If the state space E is countable, an interesting class of stochastic process are *Markov chains*, which are defined by the following property:

Definition A.10. *A continuous time stochastic process $X(t)$ with denumerable state space E is a* Markov chain *if and only if:*

$$P(X(t+s)=j \mid X(t)=i, X(t_0)=i_0, \ldots, X(t_n)=i_n) = P(X(t+s)=j \mid X(t)=i) \tag{A.7}$$

for all $i, j \in E$, $0 \leqslant t_0 < t_1 < \ldots < t_n < t$, $i_k \in E$ and $s \geqslant 0$.

Equation (A.7) is called the *Markov property*, and it states that the behavior of the process after the present time t, given the value of the state at this time, is independent of the previous values of the state.

If moreover the *transition probabilities*:

$$P(X(t+s)=j \mid X(t)=i)$$

do not depend on t, we call the Markov chain *homogeneous*. The matrix:

$$P_{ij}(t) = P(X(t)=j \mid X(0)=i)$$

is called the time t transition matrix, and $\{P(t) : t \in \mathbb{R}^+\}$ the *transition semigroup*. Note that each row i of $P(t)$ must add up to 1, since these are the conditional probabilities of being in some state, given that at time 0 we are in state i.

By applying the Markov property (A.7) for intermediate states, the following relationship between transition matrices can be established:

$$P(t+s) = P(t)P(s),$$

which is called the *Chapman-Kolmogorov* equation.

Defining a Markov chain model via its transition semigroup can be hard, since we are dealing with a family of matrices. We can simplify this by noting that, if the derivative $P'(s)$ is well defined at $s = 0$, then the Chapman-Kolmogorov equations together with the condition $P(0) = I$ imply:

$$P'(t) = P(t)Q,$$

where $Q = P'(0)$. Therefore:

$$P(t) = e^{Qt}.$$

Thus, the matrix $Q = P'(0)$ completely determines the transition probabilities, and it is called the *infinitesimal generator* of the chain. The entries of the matrix Q are interpreted as transition *rates*. Note that in particular $q_{ii} = -\sum_{j \neq i} q_{ij}$. More formally, the transition probabilities of the Markov chain satisfy the following property:

Proposition A.3. *Let $X(t)$ be a Markov chain with infinitesimal generator Q. Then, for any $i \neq j \in E$ we have:*

$$P(X(t+h) = j \mid X(t) = i) = q_{ij}h + o(h).$$

We are interested in the limiting behavior of a Markov chain process. An important definition is the following:

Definition A.11 (Irreducible Markov chain)**.** *A Markov chain is called* irreducible *if for every $i, j \in E$ there exists $s \geqslant 0$ such that $P_{ij}(s) > 0$.*

The intuition is that we can reach every state from any other state in finite time with positive probability.

We now turn our attention to the distribution of the random variable $X(t)$. Note that:

$$P(X(t) = j) = \sum_i P(X(t) = j \mid X(0) = i)P(X(0) = i) = \sum_i P(X(0) = i)P_{ij}(t),$$

or in vector notation, if v is a row vector with $v_i(t) = P(X(t) = i)$, we can write:

$$v(t) = v(0)P(t).$$

An *invariant distribution* for a Markov chain is a probability distribution π on E such that, when interpreted as a row vector, verifies:

$$\pi = \pi P(t) \quad \forall t \geqslant 0.$$

Note that if $v(0) = \pi$, then the distribution of $X(t)$ remains the same for all t. It is not difficult to see that in this case, the Markov chain process is stationary, and we say that the process is in steady state. Note that the above condition is equivalent to finding a vector π with $\pi_i \geqslant 0$ and such that:

$$\pi Q = 0,$$
$$\pi \mathbf{1} = 1,$$

where Q is the infinitesimal generator and $\mathbf{1}$ is a column vector of ones.

Not every Markov chain has an invariant distribution. The existence of invariant distributions is linked to how often the states are visited. An important definition is the following:

Definition A.12 (Positive recurrent Markov chain)**.** *Let $i \in E$ and consider the following stopping times:*

$$\tau_i^{\rightarrow} = \inf\{t \geqslant 0 : X(t) \neq i\},$$
$$\tau_i = \inf\{t \geqslant \tau_i^{\rightarrow} : X(t) = i\}.$$

$\tau_i^{\rightarrow}$ *is the* exit time *from i and τ_i is the* return time *to i. A state i is called* positive recurrent *if and only if:*

$$E(\tau_i \mid X(0) = i) < \infty.$$

If all of the states are positive recurrent, we say that the chain is positive recurrent. In particular, for irreducible chains, either all states are positive recurrent or not.

Note that if the chain is positive recurrent, we expect to visit all of the states frequently, so the chain can effectively be found after some time in any of the states. This is linked to the asymptotic behavior of the chain, which we summarize as follows:

Theorem A.6 (Ergodic theorem for Markov chains)**.** *If a Markov chain is irreducible and positive recurrent, there exists a unique invariant probability distribution π given by:*

$$\pi_i = \frac{E(\tau_i^{\rightarrow} \mid X(0) = i)}{E(\tau_i \mid X(0) = i)},$$

which also satisfies $\pi_i > 0 \; \forall i$.

Moreover, for any initial distribution $\nu(0)$, the distribution of the process at time t, $\nu(t)$ converges to π as $t \to \infty$ in distribution.

Due to this result, we call irreducible and positive recurrent chains *stable*. This is because no matter what initial distribution we choose for the system, after some time, it will exhibit a steady state regime characterized by the distribution π.

Bibliography

[Alpcan and Basar, 2003] Alpcan, T. and Basar, T. (2003). A utility-based congestion control scheme for internet-style networks. In IEEE Infocom.

[Altman et al., 2006] Altman, E., Avrachenkov, K. and Ayesta, U. (2006). A survey on Discriminatory Processor Sharing. Queueing Systems: Theory and Applications *53*, 53–63.

[Arrow et al., 1958] Arrow, K., Hurwicz, I. and Uzawa, H. (1958). Studies in linear and nonlinear programming. Stanford University Press, Stanford.

[Baccelli and Zuyev, 1997] Baccelli, F. and Zuyev, S. (1997). Stochastic geometry models of mobile communication networks. In Frontiers in Queueing pp. 227–243. CRC Boca Raton, FL.

[Baldo et al., 2007] Baldo, N. et al. (2007). dei80211mr: a new implementation of 802.11 for ns2. http://www.dei.unipd.it/wdyn/?IDsezione=5090.

[Bertsekas and Gallager, 1991] Bertsekas, D. and Gallager, R. (1991). Data Networks. Prentice-Hall, Upper Saddle River, NJ.

[Bianchi, 2000] Bianchi, G. (2000). Performance analysis of the IEEE 802.11 distributed coordination function. IEEE Journal on Selected Areas in Communications *18*, 535–547.

[Bonald and Massoulié, 2001] Bonald, T. and Massoulié, L. (2001). Impact of Fairness on Internet Performance. In ACM SIGMETRICS/Performance pp. 82–91,.

[Bonald and Proutière, 2006] Bonald, T. and Proutière, A. (2006). Flow-level stability of utility-based allocation in non-convex rate regions. In Proceedings of the 40th Annual Conference on Information Systems and Sciences (CISS 2006) pp. 327–332,.

[Boyd and Vanderberghe, 2004] Boyd, S. and Vanderberghe, L. (2004). Convex Optimization. Cambridge University Press, Cambridge, UK.

[Brakmo and Peterson, 1995] Brakmo, L. and Peterson, L. (1995). TCP Vegas: end-to-end congestion avoidance on a global Internet. IEEE Journal on Selected Areas in Communications *13*, 1465–1480.

[Briscoe, 2006] Briscoe, B. (2006). Flow Rate Fairness: Dismantling a Religion. Computer Communication Review *37*, 63–74.

[Briscoe, 2008] Briscoe, B. (2008). Problem Statement: Transport protocols don't have to do fairness. Internet Draft draft-briscoe-tsvwg-relax-fairness-01 Internet Engineering Task Force.

[Cheng et al., 2004] Cheng, J., Wei, D. X. and Low, S. H. (2004). FAST TCP: motivation, architecture, algorithms, performance. In IEEE Infocom.

[Chiang et al., 2007] Chiang, M., Low, S. H., Calderbank, A. R. and Doyle, J. C. (2007). Layering as optimization decomposition: A Mathematical Theory of Network Architectures. In Proceedings of the IEEE vol. 95, pp. 255–312,.

[de Veciana et al., 1999] de Veciana, G., Lee, T.-J. and Konstantopoulos, T. (1999). Stability and Performance Analysis of Networks Supporting Services with Rate Control - Could the Internet Be Unstable? In IEEE Infocom pp. 802–810,.

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

[Feijer and Paganini, 2009] Feijer, D. and Paganini, F. (2009). Krasovskii's method in the stability of network control. In Proceedings of the American Control Conference.

[Feller, 1965] Feller, W. (1965). An Introduction to Probability Theory and Its Applications. John Wiley & Sons, New York.

[Floyd and Henderson, 1999] Floyd, S. and Henderson, T. (1999). Request for Comments 2582: The NewReno Modification to TCP's Fast Recovery Algorithm.

[Floyd and Jacobson, 1993] Floyd, S. and Jacobson, V. (1993). Random early detection gateways for congestion avoidance. IEEE/ACM Transactions on Networking *1*, 397–413.

[Han et al., 2006] Han, H., Shakkotai, S., Hollot, C., Srikant, R. and Towsley, D. (2006). Multipath TCP: A joint congestion control and routing scheme to exploit path diversity in the Internet. IEEE/ACM Transactions on Networking *14*, 1260–1271.

[Hardin, 1968] Hardin, G. (1968). The tragedy of the commons. Science *162*, 1243–1248.

[Haviv and van der Wal, 2008] Haviv, M. and van der Wal, J. (2008). Mean waiting times for phase-type discriminatory processor sharing. European Journal of Operations Research *189*, 375–386.

[IEEE 802.11, 2007] IEEE 802.11 (2007). IEEE 802.11-2007, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. `http://www.ieee802.org/11/`.

[Jacobson, 1988] Jacobson, V. (1988). Congestion avoidance and control. ACM Computer Communication Review *18*, 314–329.

[Jiang and Walrand, 2008] Jiang, L. and Walrand, J. (2008). A Distributed CSMA Algorithm for Throughput and Utility Maximization in Wireless Networks. In Proceedings of the Forty-Sixth Annual Allerton Conference on Communication, Control, and Computing.

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

[Kelly et al., 1998] Kelly, F., Maulloo, A. and Tan, D. (1998). Rate control in communication networks: shadow prices, proportional fairness and stability. Journal of the Operational Research Society *39*, 237–252.

[Kelly, 1997] Kelly, F. P. (1997). Charging and rate control for elastic traffic. European Transaction on Telecommunications *8*, 33–37.

[Kelly and Williams, 2004] Kelly, F. P. and Williams, R. (2004). Fluid model for a network operating under a fair bandwidth sharing policy. Annals of Applied Probability *14*, 1055–1083.

[Khalil, 1996] Khalil, H. K. (1996). Nonlinear Systems. Prentice-Hall, Upper Saddle River, NJ.

[Kumar et al., 2007] Kumar, A., Altman, E., Miorandi, D. and Goyal, M. (2007). New insights from a fixed-point analysis of single cell IEEE 802.11 WLANs. IEEE/ACM Transactions on Networking *15*, 588–601.

[Lebeugle and Proutiere, 2005] Lebeugle, F. and Proutiere, A. (2005). User-level performance of WLAN hotspots. In Proceedings of the 19th International Teletraffic Conference.

[Lee et al., 2006] Lee, J.-W., Chiang, M. and Calderbank, R. A. (2006). Jointly optimal congestion and contention control based on network utility maximization. IEEE Communication Letters *10*, 216–218.

[Lin and Shroff, 2006] Lin, X. and Shroff, N. B. (2006). Utility Maximization for Communication Networks with Multi-path Routing. IEEE/ACM Transactions on Automatic Control *51*, 766–781.

[Lin et al., 2006] Lin, X., Shroff, N. B. and Srikant, R. (2006). A Tutorial on Cross-Layer Optimization in Wireless Networks. IEEE Journal on Selected Areas in Communications *24*, 1452–1463.

[Low and Lapsley, 1999] Low, S. H. and Lapsley, D. (1999). Optimization Flow Control, I: basic algorithm and convergence. IEEE/ACM Transactions on Networking *7*, 861–874.

[Low et al., 2002] Low, S. H., Paganini, F. and Doyle, J. C. (2002). Internet Congestion Control. IEEE Control Systems Magazine *22*, 28–43.

[Massoulié, 2007] Massoulié, L. (2007). Structural properties of proportional fairness: stability and insensitivity. Annals of Applied Probability *17*, 809–839.

[Massoulié and Roberts, 1999] Massoulié, L. and Roberts, J. (1999). Arguments in favour of admission control of TCP flows. In International Teletraffic Conference 16.

[Mathis et al., 1997] Mathis, M., Semke, J., Mahdavi, J. and Ott, T. (1997). The macroscopic behavior of the TCP congestion avoidance algorithm. ACM Computer Communication Review *27*, 525–536.

[McCanne and Floyd, 2000] McCanne, S. and Floyd, S. (2000). ns Network Simulator. `http://www.isi.edu/nsnam/ns/`.

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

[Paganini, 2002] Paganini, F. (2002). A global stability result in network flow control. Systems and Control Letters *46*, 153–163.

[Paganini and Mallada, 2009] Paganini, F. and Mallada, E. (2009). A unified approach to congestion control and node-based multipath routing. IEEE/ACM Transactions on Networking *17*, 1413–1426.

[Paganini et al., 2009] Paganini, F., Tang, K., Ferragut, A. and Andrew, L. (2009). Stability of networks under general file size distribution and alpha fair bandwidth allocation. In 47th Annual Allerton Conference on Communication, Control and Computing.

[Proutière et al., 2008] Proutière, A., Yi, Y. and Chiang, M. (2008). Throughput of Random Access without Message Passing. In Proceedings of the 44th Conference on Information Science and Systems (CISS 08).

[RFC 793, 1981] RFC 793 (1981). Request for Comments 793: Transmission Control Protocol.

[Robert, 2003] Robert, P. (2003). Stochastic Networks and Queues. Stochastic Modelling and Applied Probability Series, Springer-Verlag, New York.

[Srikant, 2004] Srikant, R. (2004). The Mathematics of Internet Congestion Control. Birkhäuser, Boston, MA.

[Voice, 2007] Voice, T. (2007). Stability of multi-path dual congestion control algorithms. IEEE/ACM Trans. on Networking *15*, 1231–1239.

[Wen and Arcak, 2003] Wen, J. T. and Arcak, M. (2003). A Unifying Passivity Framework for Network Flow Control. In IEEE/Infocom.