A Theoretical Study of Internet Congestion Control: Equilibrium and Dynamics

Thesis by

Jiantao Wang

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

California Institute of Technology

Pasadena, California

2005

(Defended July 28, 2005)

To My Parents

Acknowledgements

I would like to express my deepest gratitude and appreciation to my advisors John Doyle and Steven Low. Without their generosity and assistance, the completion of this thesis would not have been possible. John provided me the great opportunity to pursue my doctoral degree at Caltech. His great vision, incredible insight, and amazing big picture have always been a source of inspiration to my research. Steven has given me constant encouragement and excellent guidance through my research. There were countless individual meetings and email exchanges from him providing me detailed help from formulating the research idea, to solving the problem and to presenting the solution rigorously. It is hard to imagine having done my Ph.D. without his help.

My gratitude also extends to Mani Chandy, Richard Murray, and Babak Hassibi for serving on my thesis committee despite their busy schedules and for providing valuable feedback.

I extend my gratitude to all my friends and colleagues in Netlab. To Ao Tang for the fruitful and enjoyable collaboration on various research projects. To Lun Li for the collaboration and discussion in the TCP/IP routing project. To Xiaoliang Wei for the simulation and discussion on the FAST stability project. These projects contribute to a major part of this dissertation. I would like to thank Cheng Jin for numerous conversations on FAST TCP. I also appreciate the helpful discussions with Hyojeong Choe about the stability of Vegas and with Joon-Young Hyojeong about the global stability of FAST. I would like to acknowledge Sanjeewa Athuraliya for introducing me to the *ns-2* simulator. Thanks also go to Christine Ortega for her arrangement of all my conference trips.

I would also like to acknowledge my friends in CDS: William Dunbar, Shane Ross, Francois Lekien, Yindi Jing, Nizar Batada, and Xin Liu. I would also like to thank Yong

Wang who helped me to settle down when I first came to the States. I thank Aristotelis Asimakopoulos and Lun Li for sharing an office with me.

I would like to thank my girlfriend Huirong Ai for her love and support. I would also like to express my gratitude to my parents, and this dissertation is dedicated to them.

Abstract

In the last several years, significant progress has been made in modelling the Internet congestion control using theories from convex optimization and feedback control. In this dissertation, the equilibrium and dynamics of various congestion control schemes are rigorously studied using these mathematical frameworks.

First, we study the dynamics of TCP/AQM systems. We demonstrate that the dynamics of queue and average window in Reno/RED networks are determined predominantly by the protocol stability, not by AIMD probing nor noise traffic. Our study shows that Reno/RED becomes unstable when delay increases and more strikingly, when link capacity increases. Therefore, TCP Reno is ill suited for the future high-speed network, which has motivated the design of FAST TCP. Using a continuous-time model, we prove that FAST TCP is globally stable without feedback delays and provide a sufficient condition for local stability when feedback delays are present. We also introduce a discrete-time model for FAST TCP that fully captures the effect of *self-clocking* and derive the local stability condition for general networks with feedback delays.

Second, the equilibrium properties (i.e., fairness, throughput, and capacity) of TCP/AQM systems are studied using the utility maximization framework. We quantitatively capture the variations in network throughput with changes in link capacity and allocation fairness. We clarify the open conjecture of whether a fairer allocation is *always* more efficient. The effects of changes in routing are studied using a joint optimization problem over both source rates and their routes. We investigate whether minimal-cost routing with proper link costs can solve this joint optimization problem in a distributed way. We also identify the tradeoff between achievable utility and routing stability.

At the end, two other related projects are briefly described.

Contents

List of Figures

List of Acronyms

ACK	Acknowledgement
AIMD	Additive Increase Multiplicative Decrease
AQM	Acitve Queue Management
ATM	Asynchronous Transfer Mode
ECN	Explicit Congestion Notification
FDDI	Fiber Distributed Data Interface
FIFO	First In First Out
HTTP	Hyper Text Transfer Protocol
LAN	Local Area Network
IP	Internet Protocol
ISP	Internet Service Provider
MTU	Maximum Transmission Unit
RED	Random Early Detection
RFC	Request for Comment
RTT	Round-Trip Time
SACK	Selective ACKnowledgement
SONET	Synchronous Optical NETwork
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
WWW	World Wide Web

Chapter 1

Introduction

1.1 Challenges of developing theories for the Internet

The Internet is a worldwide-interconnected computer network that transmits data by packet switching based on the TCP/IP protocol suite. Originated from the NSFnet with a handful of nodes, it has undergone explosive growth during the last two decades. Today, it connects hundreds of millions of machines, reaches billions of people, and forms a globally distributed information-exchanging system. With services provided by the Internet, encyclopedic information on every subject can be easily searched and accessed, millions of people from every corner of the world can interact with each other via e-mail and instant messengers, and businesses can be conducted in new and more efficient ways. As the most important innovation of the last century, the Internet has fundamentally changed our lifestyle.

The huge success of the Internet is achieved with improving designs and enriching protocols. While keeping pace with the advances in communication technology and non-stop demand for additional bandwidth and connectivity, the Internet continuously experiences changes and updates in almost all aspects; see [165] for well-documented details. Now, the Internet has evolved into a large-scale, heterogeneous, distributed system with complexity unparalleled by any other engineering system. For example, its scale, measured by the number of connected hosts, has grown from two thousand at the end of 1985 to over three hundred million in 2005 with a growth rate of 80% per year. Its heterogeneity exists and increases at almost every layer. In the link layers, there are wired, wireless, fiber, and satellite links with bandwidths ranging from several Kbps to 10Gbps, and there are propagation delays from nanoseconds to hundreds of milliseconds. Different networking technologies such as Ethernet LANs, token ring, FDDI, ATM, and SONET are used simultaneously, while most of them did not even exist when the original Internet architecture was conceived. In the application layer, new applications are continuously emerging, such as multi-player network gaming, World Wide Web, streaming multimedia, peer-to-peer file sharing, etc. Accompanying this increasing complexity is the decentralizing control of the Internet. With the decommissioning of the NSFnet in 1995, large commercial ISPs began to build and operate their own backbones. The Internet topology and inter-domain routing became much more complex and hard to understand while every ISP is driven by profit.

As an evolving complex system with unprecedented scale and great heterogeneity, the Internet presents an immense challenge for networking researchers to model and analyze how it works. The innovation and development of the Internet are the results of an engineering design cycle largely based on intuitions, heuristics, simulations, and experiments. Formulating theories for such a complex heuristic system afterwards seems infeasible at the first glance, which is partly the reason why theories for the Internet are lagging far behind of its applications. However, in recent years large steps have been taken to build rigorous mathematical foundations of the Internet in several areas, such as Internet topology [92, 93], routing [51, 135], congestion control [98, 138], etc.

Previous Internet research has been heavily based on measurements and simulations, which have intrinsic limitations. For example, network measurements cannot tell us the effects of new protocols before their deployment. Simulations only work for small networks with simple topology due to the constraints of the memory size and processor speed. We cannot assume that a protocol that works in a small network will still perform well in the Internet. Furthermore, it is easier to verify the correctness of a mathematical analysis than to check the feasibility of protocols in large-scale complex networks.

A theoretical framework can greatly help us understand the advantages and shortcomings of current Internet technologies and guide us to design new protocols for identified problems and future networks. Papachristodoulou et al. [128] also argued that protocol design should be based on rigorous repeatable methodologies and systematic evaluation frameworks. Design based on intuition can easily underestimate the importance of certain system features and lead to a suboptimal solution, or even disastrous implementation. One such example is the original design of HTTP protocol quoted from Floyd and Paxson [42]:

> "The HTTP protocol used by the World Wide Web is a perfect example of a success disaster. Had its designers envisioned it in use by the entire Internet, and had they explored the corresponding consequences with analysis or simulation, they might have significantly improved its design, which in turn could have led to a more smoothly operating Internet today."

In summary, developing theories for the Internet is very important and challenging, as the design and analysis of protocols need rigorous frameworks. Recently, a unified framework to study Internet congestion control has been proposed and will be described in Section 2.3. We will study the equilibrium and dynamics of TCP systems based on this framework.

1.2 Related work in congestion control

In recent years, large steps have been taken in bringing analytical models into Internet congestion control. We survey some important work in this subsection.

The steady-state throughput of TCP Reno has been studied based on the stationary distribution of congestion windows, e.g., [38, 90, 122, 109]. These studies show that the TCP throughput is inversely proportional to end-to-end delay and to the square root of packet loss probability. Padhye et al. [124] refined the model to capture the fast retransmit mechanism and the time-out effect, and achieved a more accurate formula. This equilibrium property of TCP Reno is used to define the notion of *TCP–friendliness* and motivates the equation based congestion control TFRC [54].

Misra et al. [114, 115] proposed an ordinary differential equation model of the dynamics of TCP Reno, which is derived by studying congestion window size with a stochastic differential equation. This deterministic model treats the rate as fluid quantities (by assuming that the packet is infinitely small) and ignores the randomness in packet level, in contrast to the classical queueing theory approach, which relies on stochastic models. This model has been quickly combined with feedback control theory to study the dynamics of TCP systems, e.g., [60, 100], and to design stable AQM algorithms, e.g.,[8, 61, 82, 166, 133]. Similar flow models for other TCP schemes are also developed, e.g., [24, 101] for TCP Vegas, and [69, 157] for FAST TCP. We will study the dynamics of TCP Reno and FAST TCP with these models in Chapter 3 and 4.

The analysis and design of protocols for large-scale network have been made possible with the optimization framework and the duality model. Kelly [77, 80] formulated the bandwidth allocation problem as a utility maximization over source rates with capacity constraints. A distributed algorithm is also provided by Kelly et al. [80] to globally solve the penalty function form of this optimization problem. This algorithm is called the primal algorithm where the sources adapt their rates dynamically, and the link prices are calculated by a static function of arrival rates.

Low and Lapsley [97] proposed a gradient projection algorithm to solve its dual problem. It is shown that this algorithm globally converges to the exact solution of the original optimization problem since there is no duality gap. This approach is called the dual algorithm, where links adapt their prices dynamically, and the users' source rates are determined by a static function.

There is a large body of research in congestion control based on this utility maximization framework. Local stability with feedback delay is studied for the primal algorithm in [106, 71, 151]. For more results on global stability and stability of other algorithms, please see [161, 134, 158, 127]. For discussion on implementation of such algorithms in the Internet with ECN, see [6, 5, 89, 102, 125]. Mehyar et al. [112] analyzed converge regions when there are price estimation errors. The extension of this framework into multi-cast and multi-path routing is provided in [75, 74, 95]. The joint optimization over both routing and source rates is studied in [78, 154].

Low [96] provided a duality model that leads to a unified framework to understand and design TCP/AQM algorithms. This framework viewed the TCP source rates as primal variables and congestion measures as the dual variables, and interpreted the congestion control as a distributed primal-dual algorithm over the Internet to solve the utility maximization problem. The existing TCP/AQM protocols can be reverse-engineered to determine the underlying utility functions. The equilibrium properties of a TCP/AQM system, such as throughput and fairness, can be readily understood by studying the corresponding optimization problems with these utility functions. We can also start with a general utility function and design TCP/AQM to achieve this utility, e.g., FAST TCP [69]. The details of this duality model will be briefly covered Section 2.3.

The optimization framework can not be used in certain situations, e.g., networks with heterogeneous protocols [147]. It is worth noting that there are some other approaches to studying Internet congestion control. For example, non-cooperative game theory is used in [164, 49, 4, 32], and stochastic models are used in [148, 9] with large number flows.

1.3 Summary of main results

The main results of this dissertation are summarized in this subsection. There are two fundamental topics in this thesis: equilibrium and dynamics. First, we are concerned with the dynamics of existing TCP algorithms and examine in particular the local and global stabilities of the postulated equilibria using feedback control theory. Second, we study the equilibrium properties such as fairness, throughput, and routing using the utility optimization framework. At the end of the dissertation, we briefly describe two related projects: equilibrium of heterogeneous protocols and characteristics of CHOKe. The existing optimization framework is not applicable in these two cases, and new tools are introduced to study them.

1.3.1 Dynamics and stability

Stability is an important property of congestion control systems. There is currently no unified theory to understand the behavior of a distributed nonlinear feedback system with delay when the system loses stability. It is therefore undesirable to let TCP/AQM systems operate in an unstable regime, and unnecessary if stability can be maintained without sacrificing performance. In fact, instability can cause three problems. First, it increases jitters in source rate and delay and can be detrimental to some applications. Second, it subjects short-duration connections, which are typically delay and loss sensitive, to unnecessary delay and loss. Finally, it can lead to under-utilization of network links if queues jump between empty and full. The studies of TCP Reno and FAST TCP are shown bellow.

1.3.1.1 Local stability of TCP/RED

TCP Reno and its variants are the only congestion control schemes deployed in the Internet. It has been observed that TCP/RED may oscillate wildly, and it is difficult to reduce the oscillation by tuning RED parameters [110, 26]. Although the AIMD strategy employed by TCP Reno and noisy link traffic certainly contribute to the oscillation, we show that their effects are small in comparison with protocol instability. We demonstrate that this oscillation behavior of queue and average window is determined predominantly by the instability of TCP Reno/RED.

We provide a general nonlinear model of Reno/RED systems, and study the local stability of Reno/RED with feedback delays. We also validate the model with simulations and illustrate the stability region of TCP Reno–RED. It turns out that Reno/RED becomes unstable when delay increases and more strikingly when network capacity increases! This work is published in [99, 100] and will be presented in Chapter 3 of this dissertation.

1.3.1.2 Modelling and dynamics of FAST TCP

The oscillation persists in TCP/RED systems, even if we smooth out AIMD. Our research suggests that Reno/RED is ill suited for future high-speed networks, which motivates the design of new distributed algorithms for large bandwidth-delay product networks. The recent development in optimization and control theory for Internet congestion control played an important role in the design of new TCP algorithms. It provides a framework to understand and design protocols with the desired equilibrium and dynamic properties. FAST TCP [69] is one of such algorithms that are designed based on this theoretical framework.

The modelling and dynamics of FAST TCP is studied in this dissertation. Based on the existing continuous–time flow model, we prove that FAST TCP is globally stable for arbitrary networks when there is no feedback delay. However, this model predicts instability for homogeneous sources sharing a single link when feedback delay is large, while experiments suggest otherwise. We conjecture that this inconsistence is partly due to the *self–clocking* effect, which is not captured by this model. A discrete–time model is introduced to fully capture the effects. Using this discrete-time model, we derive a sufficient condition for local asymptotic stability for general networks with feedback delay. The condition says that local stability depends on delays only through their heterogeneity, which implies in particular that FAST TCP is locally asymptotically stable when all sources have the same delay no matter how large the delay is. We also prove global stability for a single bottleneck link in the absence of feedback delay. The techniques developed in this work are new and applicable to other protocols. These results have been published in [156, 157] and will be presented in Chapter 4.

1.3.2 Equilibrium and performance

Recent studies have shown that any TCP congestion control algorithm can be interpreted as carrying out a distributed primal-dual algorithm over the Internet to maximize aggregate utility, and a user's utility function is implicitly defined by its TCP algorithm [80, 97, 101, 96]. The equilibrium properties of TCP/AQM systems such as throughput, performance, and fairness can be studied via the corresponding convex optimization problem.

1.3.2.1 Relations among throughput, fairness, and capacity

The relations among these equilibrium quantities are studied under the optimization framework in this dissertation. More specifically, we try to answer whether a fair allocation is always inefficient and whether increasing capacity always raises throughput. We are especially interested in a class of utility functions [116]

$$U(x_i, \alpha) = \begin{cases} (1-\alpha)^{-1}\, x_i^{1-\alpha} & \text{if } \alpha \neq 1 \\ \log x_i & \text{if } \alpha = 1 \end{cases}, \qquad (1.1)$$

where α is a non-negative parameter. This utility function is special because it includes all the previously considered allocation policies: maximum throughput ($\alpha = 0$), proportional fairness ($\alpha = 1$, achieved by TCP Vegas and FAST), minimum potential delay ($\alpha = 2$, approximately achieved by TCP Reno), and max–min fairness ($\alpha = \infty$). The parameter α can be interpreted as a quantitative measure of fairness [107, 16]. An allocation is *fair* if α is large and *efficient* if aggregate throughput is large.

All examples in the literature suggest that a fair allocation is necessarily inefficient. We derive explicit expressions for the changes in throughput when the parameter α or the capacities change. We characterize exactly the tradeoff between fairness and throughput in general networks. This characterization allows us both to produce the first counter-example and trivially explain all the previous supporting examples. Surprisingly, the class of networks in our counter-example is such that a fairer allocation is *always* more efficient. In particular it implies that max–min fairness can achieve a higher aggregate throughput than proportional fairness.

Intuitively, we might expect that increasing link capacities always raises aggregate throughput. We show that not only can throughput be reduced when some link increases its capacity, but more strikingly, it can also be reduced when *all* links increase their capacities by the same amount. If all links increase their capacities proportionally, however, throughput will indeed increase. These examples demonstrate the intricate interactions among sources in a network setting that are missing in a single-link topology. This work is published in [144, 146].

1.3.2.2 Joint utility optimization over TCP/IP

The previous subsection studies the effects of changes in fairness and link capacity. In this section, we will study the effects of routing changes by investigating the joint utility maximization over source rates and their routes and try to understand the cross-layer interaction of TCP-AQM, minimum-cost routing, and resources allocation.

Routing in the current Internet within an Autonomous System is computed by IP and uses single-path, minimum-cost routing, which generally operates on a slower time scale than TCP/AQM. The joint utility maximization over both source rates and their routes can be formulated as

$$\max_{R \in \mathcal{R}} \max_{x \geq 0} \; \sum_i U_i(x_i) \quad \text{s. t. } Rx \;\leq\; c, \tag{1.2}$$

where $\mathcal{R}$ is the set of all feasible single-path routing matrices. Its Lagrangian dual is

$$\min_{p \geq 0} \sum_i \max_{x_i \geq 0} \left(U_i(x_i) - x_i \; \min_{R^i \in \mathcal{R}^i} \sum_l R_{li} p_l \right) + \sum_l c_l p_l, \tag{1.3}$$

where $\mathcal{R}_i$ denotes the set of available routes for source i. A striking feature of the associated dual problem is that the maximization over routes takes the form of minimal-cost routing with prices as link costs. This raises the question whether TCP/IP might turn out to be a distributed primal-dual algorithm to solve this joint optimization with proper choice of link costs.

We show that the primal problem (1.2) is NP-hard and in general can not be solved by minimal-cost routing. When the congestion prices generated by TCP–AQM are used as link costs, TCP/IP indeed solves the dual problem (1.3) if it converges to an equilibrium. However, this utility optimization problem is non-convex, and a duality gap generally exits between (1.2) and (1.3). Equilibrium of TCP/IP exists if and only if there is no such gap. We also show that this gap can be described as the penalty for not splitting traffic across multiple paths in single-path routing.

When such equilibrium exists, it is generally unstable under pure dynamic routing. It can be stabilized by adding a static component to the link costs, but at the expense of a reduced achievable utility in equilibrium. We demonstrate this inevitable tradeoff between utility maximization and routing stability with a simple ring network. We also present numerical results to validate this tradeoff in a general network topology. These results also suggest that routing instability can reduce aggregate utility to less than that achievable by pure static routing.

We show that if the link capacities are optimally provisioned, then *pure static* routing is enough to maximize utility even for general networks. Moreover single-path routing achieves the same utility as multi-path routing at optimality. This work is presented in

[153, 154].

1.3.3 Other related results

The following two projects are studied outside the optimization framework. We will briefly describe the model, approach, and results in Chapter 7. See [147, 142, 155, 143, 145] for details of these two projects.

1.3.3.1 Network equilibrium with heterogeneous protocols

An important assumption in the duality model is that all the TCP sources are homogeneous, that means that they all adapt to the same type of congestion signals, e.g., loss probability in TCP Reno and queueing delay in FAST [69]. During the incremental deployment of new congestion control protocols such as FAST, there is an important and inevitable phase where heterogeneous TCP algorithms reacting to different congestion signals coexist in the same network. In this situation, the current optimization framework breaks down, and the resulting equilibrium can no longer be interpreted as a solution to a utility maximization problem. Characterizing the equilibrium of a general network with heterogeneous protocols is substantially more difficult than in the homogeneous case.

We prove that, under mild assumptions, equilibrium still exists despite the lack of an underlying optimization problem using the Nash theorem in game theory. In contrast to the homogeneous protocol case with a unique equilibrium, there can be uncountably many equilibria with heterogeneous protocols as illustrated by our examples. However, we can also show that almost all networks have finitely many equilibria, and they are necessarily locally unique. Multiple locally unique equilibria can arise in two ways. First, the set of bottleneck links can be non-unique. The equilibria associated with different sets of bottleneck links are necessarily distinct. Second, even when there is a unique set of bottleneck links, network equilibrium can still be non-unique, but is always finite and odd in number. They cannot all be locally stable unless the equilibrium is globally unique. We also provide various sufficient conditions for global uniqueness. This work also appears in [147, 142].

1.3.3.2 Control unresponsive flow–CHOKe

All our previous studies have assumed that all sources utilize a certain TCP scheme to adapt their rates based on network congestion. The number of non-rate-adaptive (e.g., UDP-based) applications is growing in the Internet. Without a proper incentive structure, these applications may result in more severe congestion by monopolizing the network bandwidth to the detriment of rate-adaptive applications. This has motivated a new AQM algorithm CHOKe [126], which is stateless, simple to implement, and yet surprisingly effective in protecting TCP from unresponsive UDP flows.

We present a deterministic fluid model that explicitly models both the feedback equilibrium of the TCP/CHOKe system and the spatial characteristics of the queue. We prove that, provided the number of TCP flows is large, the UDP bandwidth share peaks at $(e+1)^{-1} = 0.269$ when UDP input rate is slightly larger than link capacity and drops to zero as UDP input rate tends to infinity. We clarify the spatial characteristics of the leaky buffer under CHOKe that produce this throughput behavior. Specifically, we prove that, as UDP input rate increases, even though the total number of UDP packets in the queue increases, their spatial distribution becomes more and more concentrated near the tail of the queue and drops rapidly to zero toward the head of the queue. In stark contrast to a non-leaky FIFO buffer where UDP bandwidth share would approach 1 as its input rate increases without bound, under CHOKe, UDP simultaneously maintains a large number of packets in the queue and receives a vanishingly small bandwidth share, the mechanism through which CHOKe protects TCP flows. This work is published in [155, 143, 145].

1.4 Organization of this dissertation

The rest of this dissertation is organized as follows:

Chapter 2 provides background information in congestion control research. First, various existing Transmission Control Protocols and Active Queue Management schemes are briefly described. Then a general network model of TCP/AQM systems is presented. We also review the resource allocation problem based on utility maximization. The duality model, which interprets the TCP/AQM as a distributed primal–dual algorithm, is presented with details. These models form the basis of our studies on network dynamics and equilibria in the following chapters.

Chapter 3 and 4 include our studies on the dynamics of TCP/AQM systems. We show that Reno/RED becomes unstable when delay increases and when network capacity increases. This motivated the design of FAST. The modelling of FAST and several stability results are presented.

Chapter 6 presents our research on the equilibrium properties of TCP systems. The relation between fairness and efficiency, and the relation between link capacity and source throughput are studied in an analytical way.

Chapter 5 describes the joint utility maximization problem over both source rates and their routes, and tries to answer whether TCP/IP with minimal-cost routing distributedly solves this problem by proper choice of link costs.

Chapter 7 briefly covers two other related projects. In Section 7.1, we study the equilibrium structures of networks with heterogeneous congestion control protocols that react to different congestion signals. In Section 7.2, we analyze CHOKe, which is a new AQM that aims to protect TCP sources from unresponsive flows. Both the feedback equilibrium of the TCP/CHOKe system and the spatial characteristics of the leaky queue are studied.

Chapter 8 concludes this dissertation and points out several future research directions.

Chapter 2

Background and Preliminaries

Internet congestion occurs when the aggregate demand for certain resources (e.g., link bandwidth) exceeds the available capacity. Results of this congestion include long transfer delay, high packet loss, constant packet retransmission, and even possible congestion collapse [63], in which network links are fully utilized, but the throughput, which an application obtains, is close to zero. It is clear that in order to maintain good network performance, certain mechanisms must be provided to prevent the network from being severely congested for any significant period of time.

One intuitive solution is to use network provision to provide more resources. However, Jain [66] had shown that large memory, high-speed links, and fast processors would not solve the congestion problem in computer networks. Although the bandwidth exponentially increased in the last decade, the request for additional bandwidth remained, and new applications consumed much more bandwidth than expected, e.g., peer-to-peer file sharing [52, 43]. The need for good congestion control schemes has been intensified by the increasing capacity of the Internet instead of being alleviated, while what we want to achieve is performance, stability, and fairness in a more heterogeneous environment [66]. Therefore, congestion control is still a very important subject even in the future high-speed network.

Congestion control studies the design and analysis of distributed algorithms to share network resources among competing users. The goal is to match the demand with available resources to reduce congestion and under-utilization and to allocate the resources fairly. There are two components in Internet Congestion Control. The first is a source algorithm implemented in Transmission Control Protocol to dynamically adjust the sending rate based on congestion along its path. The other is the Active Queue Management algorithm running on the routers, which updates the congestion information and feeds it back to sources implicitly or explicitly in the form of packet loss, delay, or marking. We will briefly describe several such algorithms in the following subsections.

2.1 Transmission Control Protocol (TCP)

The early version of TCP used for the Internet before 1988 did not have a proper congestion control scheme built in, and its main purpose was to guarantee reliable data transfer across the unreliable best-effort network. This resulted in frequent congestion collapses throughout the mid-1980s until the algorithm to dynamically adapt source rate based on packet loss was introduced by Jacobson [63]. The algorithm has undergone many minor, but important changes, e.g., [64, 140, 108, 3, 40, 57]. It has several slightly different implemented versions such as TCP Tahoe, Reno, NewReno, and SACK, which have similar essential features of additive increase and multiplicative decrease. We will not distinguish them and will refer to them as TCP Reno in this dissertation.

2.1.1 TCP Reno

TCP Reno has performed remarkably well and has prevented severe congestion as the Internet expanded by five orders of magnitude in size, speed, load, and connectivity. Measurements in core routers have indicated that about 90% of all the traffic is generated by TCP Reno sources [137]. TCP Reno is the only deployed congestion control scheme in the current Internet, and it is very important for us to have a solid understanding of how it works. In this subsection, we will describe the congestion control mechanism of TCP Reno.

A TCP Reno source sends packets using a sliding window algorithm, see [129] for details. Its sending rate is controlled by the congestion window size, which is the maximum number of packets that have been sent, yet not acknowledged. When the congestion window is exhausted, the source must wait for an acknowledgement before sending a new packet. This is the ”self-clocking” feature [98], which automatically slows down the source when a network becomes congested and round-trip time (RTT) increases. Since there is roughly one window of packets sent out for every RTT, the source rate is controlled by the window size divided by RTT. The key idea in this algorithm is to additively increase congestion window size for additional bandwidth and multiplicatively decrease it while network congestion is detected.

A connection starts with a small window size of one packet, and the source increments its window by one every time it receives an acknowledgement. This doubles the window every RTT and is called *slow start*, see Figure 2.1. In this phase, the source exponentially increases its rate and can grab the available bandwidth quickly. (It is *slow* compared to the old design where the source sends as many packets as the receiver’s advertised window size.) When the window size reaches the slow-start threshold (ssThreshold), the source enters the *congestion avoidance* phase, where it increases its window by the reciprocal of the current window size for each acknowledgement (ACK). This increases the window by one in each round-trip time and is referred to as additive increase. When a loss is detected through duplicate ACKs, the source halves its window size, updates the value of ssThreshold, and performs a *fast recovery* by retransmitting the lost packets. When a loss is detected through timeout expiration, the congestion window is reset to one, and the source re-enters the *slow-start* phase. The mathematical model and dynamics of TCP Reno will be studied in detail in Chapter 3.

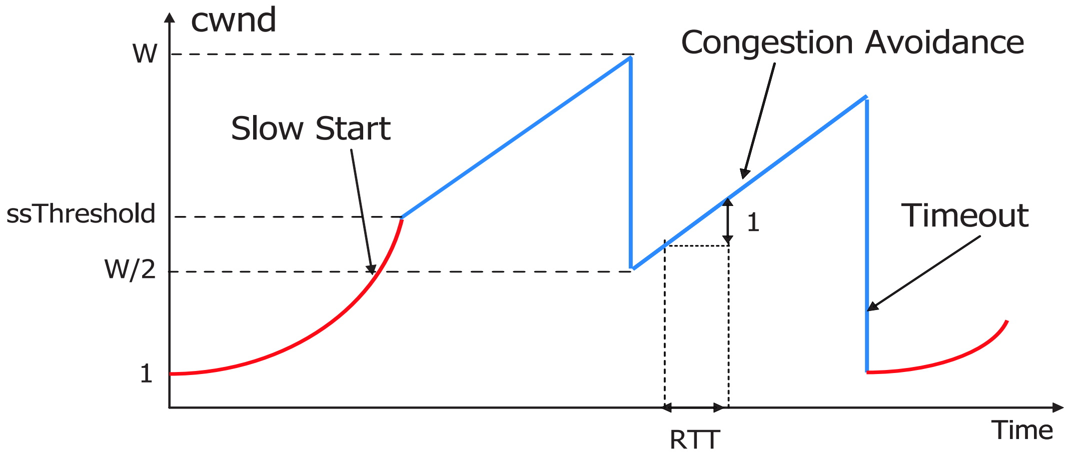


Figure 2.1: Congestion window of TCP Reno.

There are some drawbacks in using packet loss as an indication of congestion. First, high utilization can be achieved only with full queues, i.e., when the network operates at the boundary of congestion [98]. This is ill-suited to the heavy-tailed TCP traffic, as observed in [162, 91, 167]. While most TCP connections are “mice” (small, requiring low latency [53]), a few “elephants” (long TCP connections, tolerating large latency) generate most of the traffic. First, operating around a state with full queue, the mice suffer unnecessary loss and queuing delay. Second, the performance of a loss-based TCP source will be degraded in the situation where losses are due to other effects (e.g., wireless links).

There are also some other TCP alternatives we will briefly describe below.

2.1.2 TCP Vegas

Instead of using packet loss as a measure of congestion, there is another class of congestion control algorithms that adapt their congestion window size based on end-to-end delay. This approach is originally described by Jain [65] and is represented by TCP Vegas [19, 20] and FAST TCP [69].

There are several key differences between TCP Vegas and TCP Reno. In slow-start phase, TCP Vegas incorporates its congestion detection mechanism into slow-start with minor modifications to grow the window size more cautiously. When packet loss is detected, TCP Vegas uses a new retransmission mechanism and treats the receipt of certain ACKs as a trigger to check if a timeout should happen [19]. The most important difference between them is that TCP Vegas updates its congestion window size based on end-to-end delay.

TCP Vegas source estimates its round-trip propagation delay as the minimal RTT, measuring the current RTT for each ACK received. Then, it can figure out the number of its own packets buffered along the path as the product of end-to-end queueing delay and its sending rate. The source will try to keep this number in a region, specified by two parameters α and β. The window size linearly increases, decreases, or maintains the same by comparing this number with α and β. The aim is to maintain a small number of packets in the buffer to fully utilize the link and experience a small queueing delay.

Low et al. [101] provided a duality model for TCP Vegas and studied its equilibrium in detail. It is shown that TCP Vegas achieves weighted proportional fairness at the equilibrium when there is sufficient buffer. Choe and Low [24] studied the dynamics of the TCP Vegas algorithm, showing that it can become unstable in the presence of network delay, and provided modification for better stability.

2.1.3 FAST TCP

It is shown [100, 59] that the current congestion control algorithms, TCP Reno, and its variants do not scale with bandwidth-delay products of the Internet as it continues to grow, and will eventually become performance bottlenecks. This has motivated the design of FAST TCP [69, 70], which targets high-speed networks with long latency. Unlike other congestion control algorithms, it is designed based on a theoretical framework [98, 102] and aims to achieve high throughput while maintaining a stable and fair equilibrium.

FAST TCP adjusts its congestion window size based on queueing delay instead of packet loss. In networks with large bandwidth-delay products, packet losses are rare events, and each packet loss only provides one bit of information. The queueing delay can be measured for each ACK packet, and the results provide multi-bit information. The measured queueing delay is processed with a low-pass filter to provide more accurate and smooth information about the congestion in the networks. This measured queueing delay is fed into an equation to decide the changes in the congestion window size. In the congestion avoidance phase, FAST periodically updates the congestion window according to [69]:

$$\mathtt{w} \;\longleftarrow\; \min\left\{2\mathtt{w},\ (1-\gamma)\mathtt{w} \,+\, \gamma\left(\frac{\mathtt{baseRTT}}{\mathtt{RTT}}\mathtt{w}+\alpha\right)\right\}$$

where $\gamma \in (0,1]$, $\mathtt{baseRTT}$ is the minimum RTT observed so far, and α is a constant.

Although FAST TCP and TCP Vegas have different window update algorithms and dynamics, they share the same equilibrium properties. Similar to TCP Vegas, FAST achieves weighted proportional fairness, and the constant α is also the number of packets a flow attempts to maintain in the network buffers at equilibrium.

There are some other important implementation features of FAST that are not described here, for example, burst control and window pacing. The details of the architecture, algorithms, extensive experimental evaluations of FAST TCP, and comparison with other TCP variants can be found in [68]. I will provide mathematical models of FAST TCP, and will study its dynamics in detail in Chapter 4.

There are also some other TCP congestion control proposals for high-speed networks, which will not be covered in detail here. The eXplicit Congestion control Protocol (XCP) [76], proposed by Katabi et al., is designed based on control theory and requires explicit feedback from the routers to achieve stability and fairness. The High Speed TCP (HSTCP) [39], proposed by Floyd, is a modification of current TCP to increase more aggressively and decrease more cautiously in large congestion window situations. The scalable TCP [81], proposed by Kelly, uses multiplicative increase and multiplicative decease instead of TCP Reno's AIMD. The BIC TCP [163], proposed by Xu et al., uses binary search increase and additive increase. See [21, 118] for experiments and performance comparisons between these new proposals.

2.2 Active Queue Management (AQM)

The AQM algorithm runs on a router, which updates and feedbacks congestion information to end-users. The feedback is usually in the form of packet loss, delay, or marking. There is a very large body of AQMs proposed, and I will just describe few common AQMs in this subsection.

2.2.1 Droptail

Droptail is the simplest AQM scheme in the current Internet. It is just a first-in-first-out(FIFO) queue with limited capacity, and it simply drops any incoming packets when the queue is full. Since it is simple and easy to implement, Droptail is the dominant AQM in the current Internet. This FIFO queue helps to achieve better link utilization and absorbs the bursty traffic.

The congestion information in a Droptail queue is updated by the queueing process and is represented by the size of the backlog buffer. The delay-based TCP algorithms, e.g., TCP Vegas and FAST, receive this information by sensing the changes in the round-trip delay. The dynamics of FAST TCP will be studied in Chapter 4 using Droptail routers with sufficient buffer.

For loss-based sources, the Droptail queue sends back one bit of information by a packet drop, which indicates that the router buffer is full and the network is congested. When working with TCP Reno, Droptail routers have two drawbacks: the *lock-out* and the *full-queue* phenomena, which are pointed out in Braden et al. [17]. The *lock–out* phenomenon involves a single or a few sources that monopolize the bandwidth. This situation is usually the result of synchronization [55, 117]. The *full-queue* phenomenon refers to the effect that the queue can be full (or almost full) for long periods of time, which produces large end-to-end delays.

One possible solution to overcome these problems is to detect congestion early and to convey congestion notification to sources before queue overflow. We describe one such solution below.

2.2.2 Random Early Detection (RED)

The Random Early Detection algorithm, or RED, is proposed by Floyd and Jacobson [41] to solve the synchronization and full queue problems of Droptail. In contrast to Droptail that drops packets deterministically when the buffer is full, the RED algorithm drops arriving packets probabilistically based on average queue size. The packet is dropped randomly to break up synchronized processes that lead to the lock-out phenomenon, and RED controls the average queue size to avoid queue overflow.

There are two components in the RED algorithm. The first is the estimation of average queue size using the exponential weighted average, which can also be interpreted as a low pass filter to get rid of noise. The other part of the algorithm decides whether to drop an incoming packet. There are three RED parameters `minth`, `maxth`, and `maxp` controlling the dropping probability as shown in Figure 2.2. When the average queue size is less than the minimum threshold `minth`, the dropping probability is zero. When it exceeds the maximum threshold `maxth`, all the incoming packets will be dropped. When it is in between, packets will be dropped with a probability that varies linearly from 0 to `maxp`.

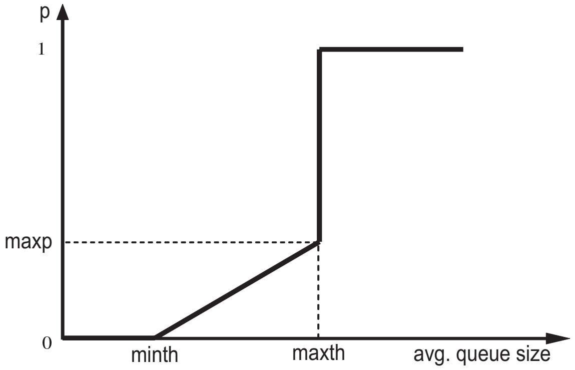


Figure 2.2: RED dropping function.

The RED can also mark the incoming packets instead of dropping them with the deployment of Explicit Congestion Notification (ECN) [131] to prevent packet loss and improve throughput. The basic idea of ECN is to give the network the ability to explicitly signal TCP sources of congestion using one additional bit in the IP packet header and to have the TCP sources reduce their transmission rates in response to the marked packets.

The dynamics of Reno/RED systems will be studied in details in Chapter 3. It is shown that the system becomes unstable when the delay increases or when the link capacity increases. It is very difficult to configure the RED parameters to achieve better performance.

There has been a large body of AQM schemes proposed recently. Some notable examples include, Stabilized RED [123], PI controller [58], REM [5] , AVQ [86], BLUE [35], etc.

2.2.3 CHOKe

CHOKe [126], which stands for “CHOose and Keep for responsive flows, CHOose and Kill for unresponsive flows”, is proposed by Pan et al. in 2001. It aims to penalize the unresponsive flows (e.g., UDP sources), to protect the rate-adaptive flows (e.g., TCP sources), and to ensure fairness.

The scheme, CHOKe, is particularly interesting in that it does not require any state information and yet can provide a minimum throughput to TCP flows. The basic idea of

CHOKe is explained in the following quote from [126]:

> When a packet arrives at a congested router, CHOKe draws a packet at random from the FIFO (first-in-first-out) buffer and compares it with the arriving packet. If they both belong to the same flow, then they are both dropped; else the randomly chosen packet is left intact and the arriving packet is admitted into the buffer with a probability that depends on the level of congestion (this probability is computed exactly as in RED).

The surprising feature of this extremely simple scheme is that it can bind the bandwidth share of UDP flows regardless of their arrival rate.

Its queue characteristics and the maximum throughput of unresponsive flows is studied in [143, 155, 145]. These results will be briefly covered in Section 7.2.

2.3 Unified frameworks for TCP/AQM systems

In this subsection, we will review the general frameworks for studying the equilibrium and dynamics of TCP/AQM systems. These models will be used throughout this dissertation.

2.3.1 General dynamic model of TCP/AQM

A network is modelled as a set of L links with finite capacities $c = (c_l, l \in L)$. They are shared by a set of N sources indexed by i. Each source i uses a subset $L_i \subseteq L$ of links. The sets L_i define an $L \times N$ routing matrix

$$R_{li} = \begin{cases} 1 & \text{if } l \in L_i \\ 0 & \text{otherwise} \end{cases} . \tag{2.1}$$

We use the deterministic flow model developed in [115, 99] to describe transmission rates. Two assumptions are made when using this model. First, the packets are infinitely small and the sending rate is differentiable (like fluid flow). Second, the congestion signal is fed back continuously.

Each source i has an associated transmission rate $x_i(t)$, and each link l has an aggregate incoming flow rate $y_l(t)$. Since all sources whose paths include link l contribute to $y_l(t)$, we have the equation:

$$y_l(t) \;\; = \;\; \sum_i R_{li} x_i(t - \tau_{li}^f), \tag{2.2}$$

where τ_{li}^f denotes the forward transmission delay from source i to link l.

Each link l has an associated *congestion measure* (or *price*) $p_l(t)$, which is a non-negative quantity maintained by AQM algorithms. The sources are assumed to have access to the aggregate price of all links in their route [1],

$$q_i(t) \;\; := \;\; \sum_l R_{li} p_l(t - \tau_{li}^b), \tag{2.3}$$

where τ_{li}^b denotes the backward transmission delay from link l to source i. The total round-trip time τ_i for source i thus satisfies

$$\tau_i = \tau_{li}^f + \tau_{li}^b$$

for every link l in its path.

As shown in [96], this model includes, to a good approximation, the mechanism present in existing protocols with a different interpretation for price in different protocols (e.g., marking or dropping probability in TCP Reno, queueing delay in TCP Vegas).

In this framework, a complete feedback-control system is specified by supplying two additional blocks: the source rates change according to aggregate prices in the TCP algorithm and the link prices update based on link utilization. The complete system determines both the equilibrium and dynamic characteristics of the TCP/AQM network.

Since the TCP/AQM is decentralized, the sources only have access to their local information. Therefore, the key restriction in the above control laws is that they must be

[1] This is true when delay is used as congestion price. It is approximately true for random marking and dropping when the probability is small.

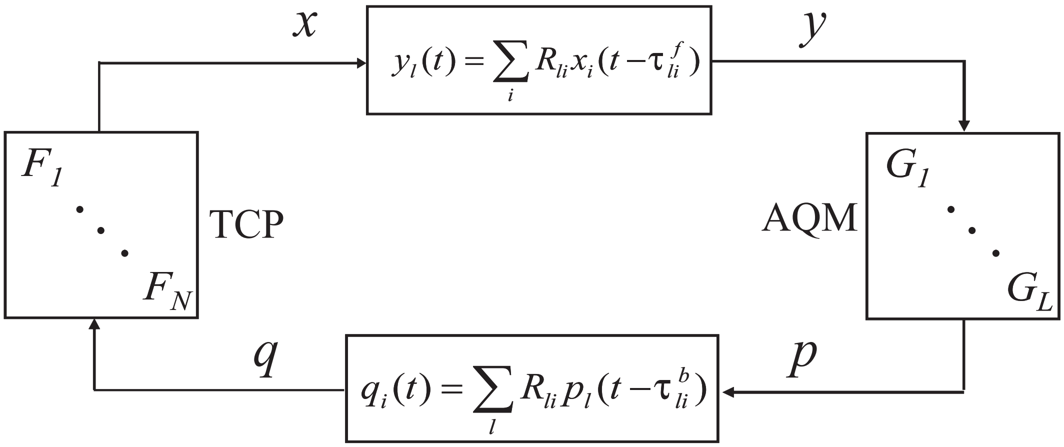
decentralized. Therefore, we can model the dynamics of TCP in a general form

$$\dot{x}_i(t) = F_i(x_i(t), q_i(t)). \tag{2.4}$$

Similarly, the dynamics at links can be written as[2]

$$\dot{p}_l(t) = G_l(p_l(t), y_l(t)). \tag{2.5}$$

The overall structure of this congestion control system is shown in Figure 2.3.



x
$y_l(t) = \sum_i R_{li} x_i(t - \tau_{li}^f)$
y
F_1
F_N
TCP
AQM
G_1
G_L
q
$q_i(t) = \sum_l R_{li} p_l(t - \tau_{li}^b)$
p

Figure 2.3: General congestion control structure.

We will study the dynamics of TCP within this general framework in Chapter 3 and 4. The equilibrium properties will be studied with the duality model introduced in the next subsection.

2.3.2 Duality model of TCP

In this section, it will be shown that the above feedback-control system solves a utility maximization problem at its equilibrium.

Suppose that the equilibrium rates and prices are given by x^*, y^*, p^*, and q^*. Based on

[2] A more accurate formulation is given in [96] that includes the internal variables of AQM in the parameters of G_l.

(2.3) and (2.2), we have following equilibrium relationships

$$y^* = Rx^*, \qquad q^* = R^T p^*. \tag{2.6}$$

Assume that equilibrium rates satisfy

$$x_i^* = f_i(q_i^*), \tag{2.7}$$

where $f_i(\cdot)$ is implicitly defined by $F_i(x^*, q_i^*) = 0$ or given by the source static law, e.g., [97]. $f_i(\cdot)$ is usually a positive, strictly monotonic decreasing function, since the source decreases its rate with increasing congestion.

Let $f_i^{-1}(x_i)$ be the inverse function of (2.7), and let a utility function $U_i(x_i)$ be its integral

$$U_i(x_i) \ := \ \int f_i^{-1}(x_i) dx_i. \tag{2.8}$$

This relation implies that $U_i(x_i)$ is a monotonic increasing and strictly concave function. It is easy to check that the equilibrium rate x_i^* uniquely solves

$$\max_{x_i \geq 0} U_i(x_i) - x_i q_i^*. \tag{2.9}$$

We interpret $U_i(x_i)$ as the benefit the source receives by transmitting at rate x_i and q_i^* as the price per unit. Then (2.9) is a maximization of the source's profit. This interpretation makes few assumptions regarding TCP and AQM and can be used for various TCP schemes.

The global optimization problem to maximize aggregate utility with capacity constraints is formulated by Kelly in [77, 80],

$$\begin{aligned} \max_{x \geq 0} \quad & \sum_i U_i(x_i) && (2.10) \\ \text{subject to} \quad & Rx \ \leq \ c. && (2.11) \end{aligned}$$

It has a unique solution, since it is maximizing a concave function over a convex set. Now we interpret the equilibrium price as the dual variables (or as the Lagrange multipliers) for the problem (2.10-2.11). Then its Lagrangian is

$$L(x,p) = \sum_i U_i(x_i) - \sum_l p_l(y_l - c) = \sum_i (U_i(x_i) - q_i x_i) + \sum_l p_l c_l. \tag{2.12}$$

The dual problem is

$$\min_{p \geq 0} \sum_i B_i(q_i) + \sum_l p_l c_l, \tag{2.13}$$

where

$$B_i(q_i) = \max_{x_i \geq 0} U_i(x_i) - x_i q_i. \tag{2.14}$$

Convex duality implies that at the optimum p^*, the corresponding x^*, which maximizes individual optimality (2.9), is exactly the unique solution to the primal problem (2.10-2.11) since (2.14) is identical to (2.9). Therefore, provided the equilibrium prices p^* can be made to align with the Lagrange multipliers, the equilibrium rate x^* solves the primal problem in a distributed way. It is proven in [96] that any link algorithm that satisfies

$$y_l^* \leq c_l \text{ with equality if } p^* > 0 \text{ for any } l \tag{2.15}$$

will guarantee this alignment. In this case, x^* is the unique primal optimal solution, and p^* is a dual optimal solution. It has been argued [96] that the condition (2.15) is satisfied by any AQM that stabilizes the queue, e.g., RED, REM, and Droptail. Therefore, various TCP/AQM protocols can be interpreted as different distributed primal-dual algorithms to solve the global optimization problem (2.10-2.11) with different utility functions.

The equilibrium structures of different congestion control schemes are characterized by their corresponding utility functions. This model provides us with a rigorous framework in which to study various equilibrium properties such as fairness, efficiency, and effects of different network parameters. In Chapter 6, I will present the methods and results following this approach.

This optimization framework can also be extended to study the interaction of TCP at a fast timescale and IP routing at a slow timescale. See Chapter 5 for details.

Chapter 3

Local Dynamics of Reno/RED

3.1 Introduction

It is well known that TCP Reno/RED can oscillate wildly and it is extremely hard to reduce the oscillation by tuning RED parameters, e.g., [110, 25]. This oscillation could be the outcome of the AIMD bandwidth probing strategy employed by TCP Reno and noise-like traffic that are not effectively controlled by TCP (e.g., short lived TCP source). Recent models e.g., [36, 59], imply however that oscillation is an inevitable outcome of the protocol itself. We present more evidence to support this view. We argue that Reno/RED oscillates not only because of the AIMD probing and noise traffic, but more fundamentally, it is due to instability. Therefore, even if there is no AIMD, and the congestion window is periodically adjusted by the average of AIMD based on loss probability, the oscillation persists. We illustrate using *ns-2* simulations that, after smoothing out the AIMD component of the oscillation, the average behavior can either be steady with small random fluctuations (when the protocol is stable), or exhibit limit cycles of amplitude much larger than random fluctuations (when it is unstable). Moreover, this qualitative behavior persists even when a large amount of noise traffic is introduced, and even when sources have different delays. We conclude that it is the protocol stability that largely determines the dynamics of Reno/RED.

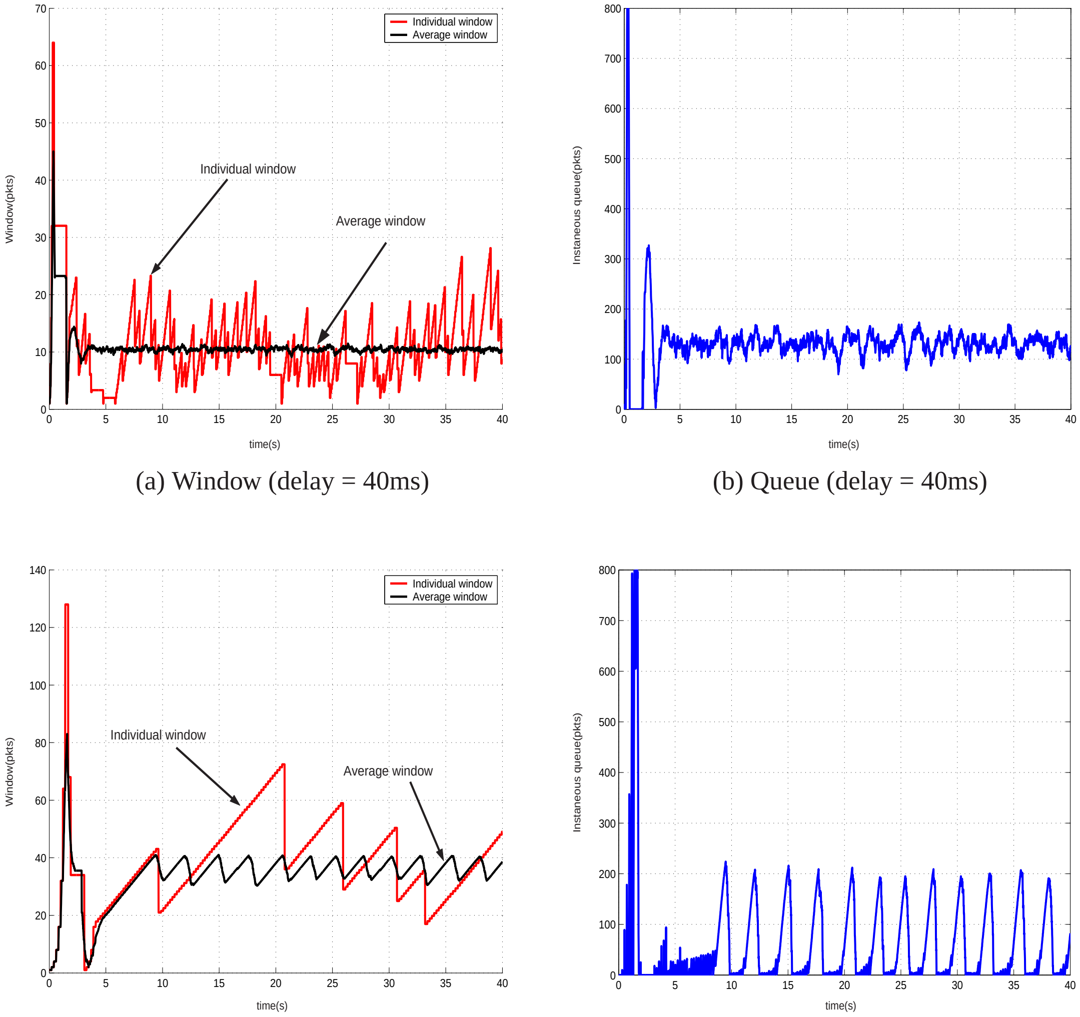
This motivates the stability characterization of Reno/RED. In Section 3.3 we develop a general nonlinear model of Reno/RED. The equilibrium structure of this system is analyzed using duality model, and a unique equilibrium exists because it is the unique solving of the underling utility maximization problem, see [96] for details. Here, we study local stability by linearizing the model around this equilibrium. The linear model generalizes the single link identical source model of [59]. We validate our model with simulations and illustrate the stability region of Reno/RED. We derive a sufficient stability condition for the special case of a single link with *heterogeneous* sources. It shows that Reno/RED becomes unstable when delay increases, or more strikingly, when link capacity increases!

In the linearized model, the gain introduced by TCP Reno increases rapidly with delay and link capacity. This induces instability and makes compensation by RED extremely difficult. In particular, RED parameters can be tuned to improve stability, but only at the cost of a large queue, even when they are dynamically adjusted. Our results suggest that Reno/RED is ill suited for future high-speed networks, which motivates the design of new distributed algorithms for high speed long latency networks.

3.2 Motivation

Why does Reno/RED oscillate? What is the effect of AIMD probing, noise traffic, and heterogeneity of delays on average congestion window and instantaneous queue size? In this section, we show that their effect is insignificant in comparison with that of protocol instability. This protocol instability is the dominant reason for oscillation in the Reno/RED system. Therefore, it is very important to study the protocol stability of Reno/RED system.

We simulate a single bottleneck network using *ns-2*. The bottleneck link has a capacity 9 pkts/ms with a constant packet size of 1000 bytes. The AQM running on this link is RED with ECN marking in *byte* mode (i.e., ACK packets are marked with negligible probability). The RED parameters are `maxp` = 0.1, `minth` = 50 pkts, `maxth` = 550 pkts, and weight for queues averaging $\alpha = 10^-4$. The link is shared by 50 persistent TCP Reno sources. We have run simulations with both one-way and two-way traffic, and the behavior is very similar. The results in Figures 3.1 and 3.2 are for two-way traffic, and those in Figure 3.3 are for one-way traffic. The measurements on the Internet [2] show that most connections have round-trip delays between 15-500ms. We perform simulations within this range of delays.



(a) Window (delay = 40ms)

(b) Queue (delay = 40ms)

(c) Window (delay = 200ms)

(d) Queue (delay = 200ms)

Figure 3.1: Window and queue traces without noise traffic.

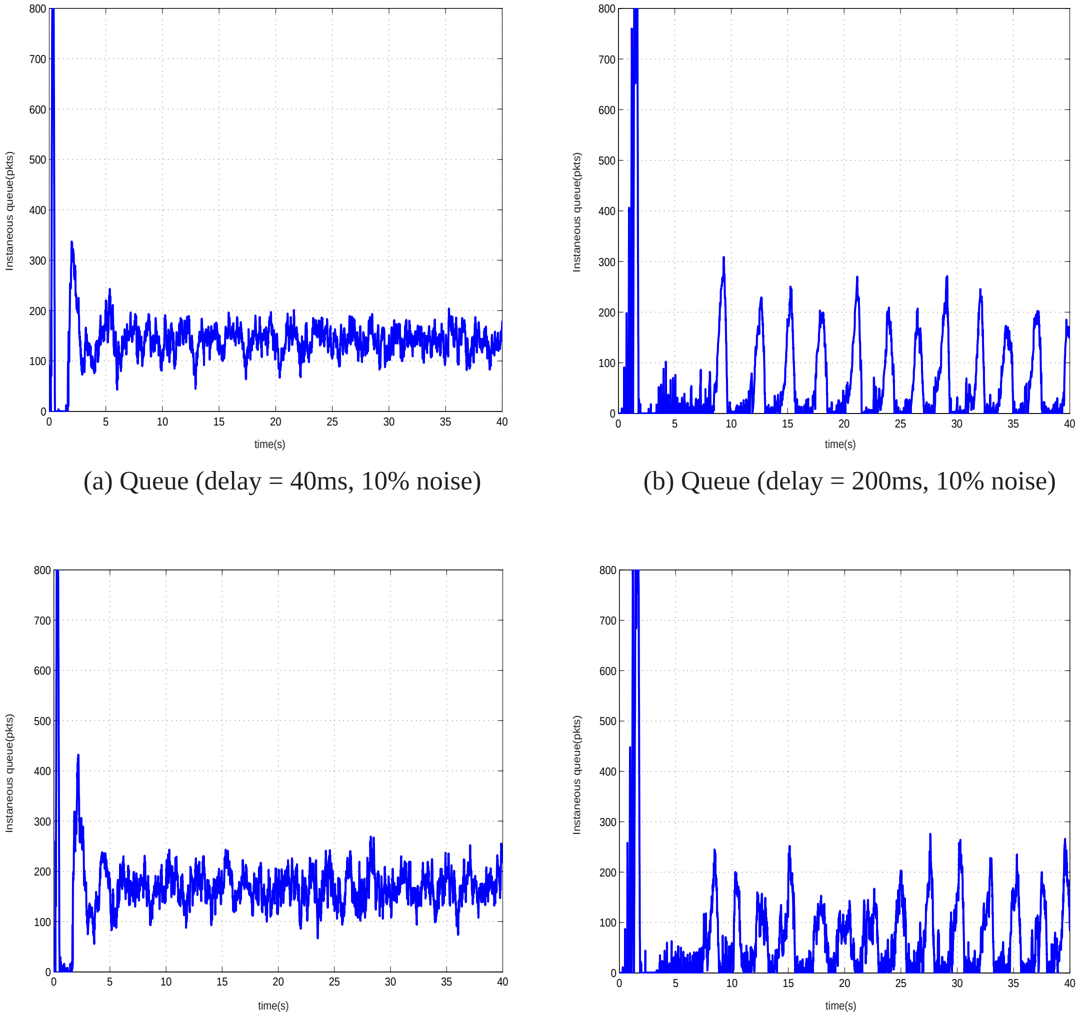
Figure 3.1 gives the result of two cases where connections have identical roundtrip propagation delay and generate traffic in both directions. Figure 3.1(a) shows an individual window and the average window that is mean window size of all 50 sources, as a function of time. They are typical traces when round-trip propagation delay is small (40ms in this case). Oscillations due to AIMD are prominent in the individual window, but disappear in the average window. Since the queue averages individual windows, it also displays a smooth trace with small random fluctuations, as shown in Figure 3.1(b). We consider the *average* behavior of the protocol stable (non-oscillatory) in this case.

Figures 3.1(c) and (d) show the corresponding windows and queue when round-trip propagation delay is increased to 200ms. Not only does the individual window oscillate with a larger amplitude, more importantly, its average displays a deterministic limit cycle. This also shows up in the queue trace. We say the protocol is in an *unstable* regime.

What is the effect of noise-like mice traffics that are not effectively controlled by Reno/RED? To get a qualitative understanding, we add additional short HTTP sources to the 50 persistent bi-directional TCP flows. Each HTTP source sends a single-packet request to its destination, which then replies with a file of size that is exponentially distributed. After the source completely receives the data, it waits for a random time that is exponentially distributed with a mean of 500 milliseconds and repeats the process. Both the request and the response are carried over TCP connections. Two sets of simulations are conducted: the first with 60 http sources generating 10% noise (i.e., persistent TCP sources occupied 90% of bottleneck link capacity), and the second set with 180 http sources generating 30% noise.

The queue traces when propagation delays are 40ms (stable) and 200ms (unstable), respectively, are shown in Figures 3.2(a) and (b) for a noise intensity of 10% and in Figure 3.2(c) and (d) for a noise intensity of 30%. The behavior of the queue is dominated by the stability of the protocol, not by noise-like mice traffic (compare with Figures 3.1(b) and (d)). In the stable regime (40ms delay), the noise traffic increases the average queue length slightly. This increases the marking probability and reduces the average window of the persistent TCP sources.

All our previous simulations are for sources with identical propagation delay. Will the dynamic behavior be very different when sources have different delays? We repeat the previous experiments, without noise, with 50 persistent connections having delays ranging from 40ms to 64ms at 1ms increments, with 2 sources to each delay value. We study their dynamic behavior when all delays are scaled up, or down, over a wide range. The behavior is qualitatively similar to the case of identical delay, with more severe queue oscillation. Figure 3.3(a) shows the instantaneous queue when the scaling factor is 0.3 (delays range from 12ms to 19.2ms), with an average delay of 15.6ms, averaged over all sources. Figure 3.3(b) shows the queue when the scaling factor is 4, with an average delay of 208ms.



(a) Queue (delay = 40ms, 10% noise)

(b) Queue (delay = 200ms, 10% noise)

(c) Queue (delay = 40ms, 30% noise)

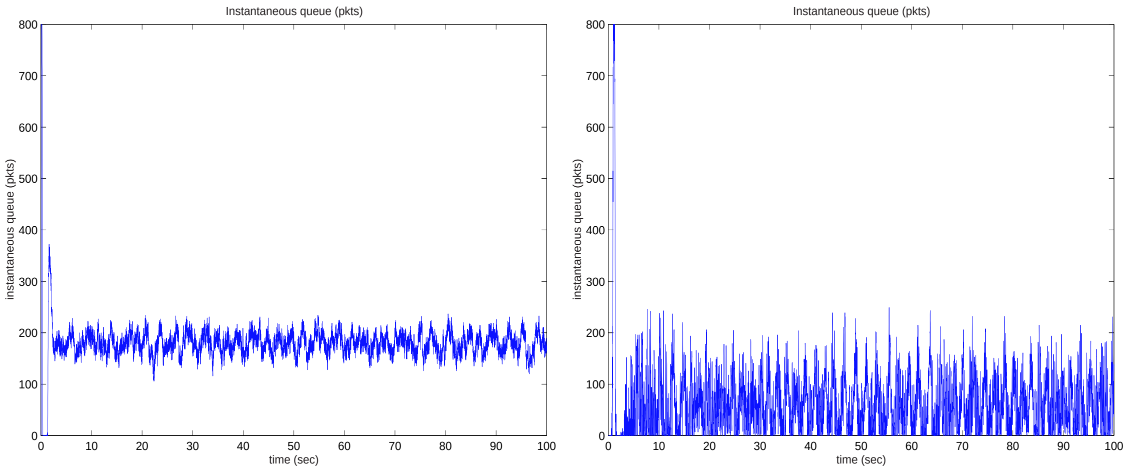
(d) Queue (delay = 200ms, 30% noise)

Figure 3.2: Queue traces with noise traffic.

Hence it is protocol stability that largely determines the dynamics of Reno/RED. We now characterize when Reno/RED is stable.

3.3 Dynamic model

In this section we develop a model of Reno/RED and use it to study the local dynamics of Reno/RED. We start with a nonlinear model, make a few remarks about its equilibrium properties, and then linearize the model around the equilibrium. We validate our linear



(a) Queue (delays from 12ms to 19ms)

(b) Queue (delays from 160ms to 254ms)

Figure 3.3: Queue traces with heterogeneous delays.

model with *ns-2* simulations and illustrate the stability region of Reno/RED. Finally we derive a stability condition for the special case of a single link with heterogeneous sources.

3.3.1 Nonlinear model of Reno/RED

The general nonlinear model for TCP/AQM systems has been presented in Section 2.3. Here more details are given in order to study Reno/RED systems.

A network is modelled as a set of L links with finite capacities $c = (c_l, l \in L)$. They are shared by a set of N sources. The interactions between them are specified by a routing matrix R where $R_{li} = 1$ if link l is in the path of source i, and $R_{li} = 0$ otherwise.

Denote $\tau_i(t)$ as the round-trip time of source i at time t; it is the sum of round-trip propagation delay d_i and the round-trip queueing delay $\sum_l R_{li} b_l(t)/c_l$. Source i's sending rate $x_i(t)$ can be formulated as

$$x_i(t) \quad = \quad \frac{w_i(t)}{\tau_i(t)}, \tag{3.1}$$

where $w_i(t)$ denotes the congestion window size. The aggregate flow rate at link l is

$$y_l(t) \quad = \quad \sum_i R_{li} x_i(t - \tau_{li}^f(t)), \tag{3.2}$$

where $\tau_{li}^f(t)$ is the forward delay from source i to link l.

The link congestion price $p_l(t)$ in the general model corresponds to the packet marking (or loss) probability in TCP Reno. The end-to-end marking probability observed at source i is actually $q_i(t) = 1 - \prod_{l \in L_i}(1 - p_l(t - \tau_{li}^b(t)))$ where $\tau_{li}^b(t)$ is the backward delay from link l to source i. We assume that $p_l(t)$ is small for all t so that, approximately, the end-to-end probability is

$$q_i(t) \quad = \quad \sum_l R_{li} p_l(t - \tau_{li}^b(t)). \tag{3.3}$$

The forward and backward delays are related to the round-trip time through

$$\tau_i(t) \quad = \quad \tau_{li}^f(t) + \tau_{li}^b(t)$$

for all $l \in L_i$.

We now model TCP Reno and RED to provide the (F, G) functions in the general framework. We focus on the AIMD algorithm of TCP Reno at the *congestion avoidance* phase. The congestion window change in this phase has been described in Section 2.1. At time t, source i transmits at rate $x_i(t)$ packets/second; hence, it receives acknowledgments at rate $x_i(t - \tau_i(t))$, assuming every packet is acknowledged. A fraction $(1 - q_i(t))$ of these acknowledgments are positive, each incrementing the window $w_i(t)$ by $1/w_i(t)$; hence the window $w_i(t)$ increases, on average, at the rate of $x_i(t - \tau_i(t))(1 - q_i(t))/w_i(t)$. Similarly negative acknowledgments are received at an average rate of $x_i(t - \tau_i(t))q_i(t)$, each halving the window, and hence the window $w_i(t)$ decreases at a rate of $x_i(t - \tau_i(t))q_i(t)w_i(t)/2$. Hence, the window evolves under Reno according to

$$\dot{w}_i(t) \quad = \quad x_i(t - \tau_i(t))(1 - q_i(t))\frac{1}{w_i(t)} - x_i(t - \tau_i(t))q_i(t)\frac{w_i(t)}{2}, \tag{3.4}$$

where $q_i(t)$ is given by (3.3).

To model RED, let $b_l(t)$ denote the instantaneous queue length at time t that evolves according to

$$\dot{b}_l(t) \;=\; y_l(t) - c_l, \tag{3.5}$$

where $y_l(t)$ is the flow rate given by (3.2) and c_l is the link capacity. Define the average queue length as $r_l(t)$. It is updated according to

$$\dot{r}_l(t) \;=\; -\alpha_l c_l \left(r_l(t) - b_l(t)\right), \tag{3.6}$$

where α is the averaging weight. Given the average queue length $r_l(t)$, the marking probability is given by

$$p_l(t) \;=\; \begin{cases} 0 & r_l(t) \le \underline{b}_l \\ \rho_l(r_l(t) - \underline{b}_l) & \underline{b}_l < r_l(t) < \overline{b}_l \\ 1 & r_l(t) \ge \overline{b}_l \end{cases}, \tag{3.7}$$

where $\underline{b}_l, \overline{b}_l$, and $\overline{p}_l$ are RED parameters, and $\rho_l = \overline{p}_l/(\overline{b}_l - \underline{b}_l)$.

In summary, Reno/RED is modelled by (3.4–3.7), and their interconnection through the network is modelled by (3.2–3.3).

The equilibrium of this system is studied in [96, 101]. The Reno/RED model is interpreted as carrying out a distributed, primal-dual algorithm to maximize the aggregate utility over the Internet. The utility function of TCP Reno is derived to be

$$U_i(x_i) \;=\; \frac{\sqrt{2}}{\tau_i}\,\tan^{-1}\left(\frac{\tau_i x_i}{\sqrt{2}}\right).$$

The equilibrium properties can be studied by solving the underlying convex program. It also implies that the Reno/RED system has a unique equilibrium.

3.3.2 Linear model of Reno/RED

We linearize the Reno/RED equations (3.4-3.7) to study its stability around equilibrium. We make several simplifying assumptions. First we assume that the routing matrix R has full row rank so there is a unique equilibrium loss probability vector p. Second, only congested links at the equilibrium are considered in the linear model. Moreover we assume that the system operates in region $\underline{b}_l < r_l(t) < \overline{b}_l$, so that the marking probability is affine in the average queue length, $p_l(t) = \rho_l(r_l(t) - \underline{b}_l)$.

We make a key assumption on the time-varying, round-trip delay. Round-trip delay appears in two places: first, in the relation between window $w_i(t)$ and rate $x_i(t)$, as expressed in (3.1), and second, in the time argument of flow rate $y_l(t)$, as expressed in (3.2), and the end-to-end marking probability $q_i(t)$, as expressed in (3.3). Inclusion of instantaneous queueing delay in the first place yields a qualitatively different model than if queueing delay is ignored or assumed constant. It means that the queue is not an integrator but has more complicated dynamics; see (3.9) below. As the proof of Theorem 3.1 shows, this dynamic is critical to the stability of Reno/RED. The resulting linear model matches simulations significantly better than if queueing delay is assumed constant. Time-varying delay in the second place makes linearization difficult, and we replace it by its (constant) equilibrium value (including equilibrium queueing delay). We approximate the delays $\tau_i(t)$, $\tau_{li}^f(t)$, and $\tau_{li}^b(t)$ by their equilibrium values in (3.2) and (3.3). With these assumptions, we linearize Reno/RED around the unique equilibrium. From (3.4), Reno becomes

$$\dot{w}_i(t) = \left(1 - \sum_l R_{li} p_l(t - \tau_{li}^b)\right) \frac{w_i(t-\tau_i)}{\tau_i(t-\tau_i)} \frac{1}{w_i(t)} - \frac{1}{2} \sum_l R_{li} p_l(t-\tau_{li}^b) \frac{w_i(t-\tau_i)\, w_i(t)}{\tau_i(t-\tau_i)}.$$

Let $w_i^*, p_l^*, \ldots$ be equilibrium quantities and $\delta w_i(t) = w_i(t) - w_i^*, \ldots.$ be the small variations near the equilibrium. Linearization yields

$$\delta \dot{w}_i(t) \;\; = \;\; -\frac{1}{\tau_i q_i^*} \sum_l R_{li} \delta p_l(t - \tau_{li}^b) \; - \; \frac{q_i^* w_i^*}{\tau_i} \delta w_i(t).$$

Around the equilibrium, the buffer process under RED evolves according to

$$\dot{b}_l(t) = \sum_l R_{li} \frac{w_i(t - \tau_{li}^f)}{\tau_i(t - \tau_{li}^f)} - c_l = \sum_l R_{li} \frac{w_i(t-\tau_{li}^f)}{d_i + \sum_k R_{ki} b_k(t - \tau_{li}^f)/c_k} - c_l.$$

Let $\tau_i = d_i + \sum_k R_{ki} b_k^*/c_k$ be the equilibrium round-trip time (including queueing delay).

Linearizing, we have

$$\delta \dot{b}_l(t) \;=\; \sum_i R_{li} \frac{\delta w_i(t-\tau_{li}^f)}{\tau_i} \;-\; \sum_k \sum_i R_{li} R_{ki}\, \delta b_k(t-\tau_{li}^f)\, \frac{w_i^*}{\tau_i^2 c_k}.$$

The second term above is ignored if we have neglected or assumed constant the queueing delay in round-trip time. The double summation sums over all links k that share any source i with link l. It says that the link dynamics in the network are coupled through shared sources. The term $\delta b_k(t-\tau_{li}^f)w_i^*/(\tau_i c_k)$ is roughly the backlog at link k due to packets of source i, under FIFO queueing. Hence the backlog $b_l(t)$ at link l decreases at a rate that is proportional to the backlog of this shared source i at another link k. This is because backlog in the path of source i reduces the *rate* at which source i packets arrive at link l, decreasing $b_l(t)$.

Putting everything together, Reno/RED is described by, in Laplace domain,

$$\begin{aligned}
\delta w(s) &= -(sI + D_1)^{-1} D_2 R_b^T(s) \delta p(s), \\
\delta p(s) &= (sI + D_3)^{-1} D_4 \delta b(s), \\
\delta b(s) &= (sI + R_f(s) D_5 R^T D_6)^{-1} R_f(s) D_7 \delta w(s),
\end{aligned}$$

where the diagonal matrices are $D_1 = \text{diag}\ (q_i^* w_i^*/\tau_i)$, $D_2 = \text{diag}\ (1/(\tau_i q_i^*))$, $D_3 = \text{diag}\ (\alpha_l c_l)$, $D_4 = \text{diag}\ (\alpha_l c_l \rho_l)$, $D_5 = \text{diag}\ (w_i^*/\tau_i^2)$, $D_6 = \text{diag}\ (1/c_l)$, and $D_7 = \text{diag}\ (1/\tau_i)$, and $R_f(s)$ and $R_b(s)$ are delayed forward and backward routing matrices, defined as

$$[R_f(s)]_{li} = \begin{cases} e^{-\tau_{li}^f s} & \text{if } l \in L_i \\ 0 & \text{otherwise} \end{cases}, \quad \text{and} \quad [R_b(s)]_{li} = \begin{cases} e^{-\tau_{li}^b s} & \text{if } l \in L_i \\ 0 & \text{otherwise.} \end{cases} \tag{3.8}$$

This model generalizes the single-link, identical-source model of [59] to multiple links with heterogeneous sources.

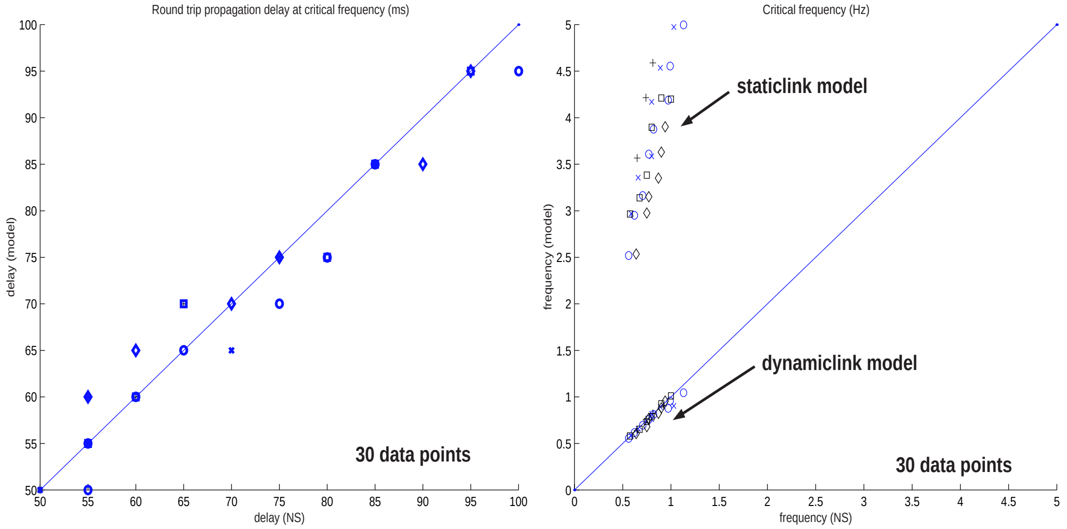
3.3.3 Validation and stability region

We present a series of experiments to validate our linear model when the system is stable or barely unstable, and to illustrate numerically the stability region.

We consider a single link of capacity c pkts/ms shared by N sources with identical round-trip propagation delay d ms. For $N = 20, 30, \ldots, 60$ sources, capacity $c = 8, 9, \ldots, 15$ pkts/ms, and propagation delay $d = 50, 55, \ldots, 100$ ms, we examine the Nyquist plot of the loop gain of the feedback system ($L(j\omega)$ in (3.9) below). For each (N, c) pair, we determine the delay $d_m(N, c)$, at 5ms increments, at which the smallest intercept of the Nyquist plot with the real axis is closest to -1. This is the delay at which the system (N, c) transits from stability to instability according to the linear model. For this delay, we compute the critical frequency $f_m(N, c)$ at which the phase of $L(j\omega)$ is $-\pi$. Note that the computation of $L(j\omega)$ requires equilibrium round-trip time τ, the sum of propagation delay $d_m(N, c)$, and equilibrium queueing delay. The queueing delay is calculated from the duality model [96]. Hence, for each (N, c) pair that becomes barely unstable at a delay between 50ms and 100ms, we obtain the critical (propagation) delay $d_m(N, c)$ and the critical frequency $f_m(N, c)$ from the analytical model. For all experiments, we have fixed the parameters at $\alpha = 10^{-4}$, $\rho = 0.1/(540 - 40) = 0.0002$.

We repeat these experiments in *ns-2*, using persistent TCP sources and RED with ECN marking. The RED parameters are (0.1, 40pkts, 540pkts, 10^{-4}), corresponding to the α and ρ values in the model. For each (N, c) pair, we examine the queue and window trajectories to determine the critical delay $d_{ns}(N, c)$ when the system transits from stability to instability. We measure the critical frequency $f_{ns}(N, c)$, the fundamental frequency of queue oscillation, from the fast fourier transform of the queue trajectory. Thus, corresponding to the linear model, we obtain the critical delay $d_{ns}(N, c)$ and frequency $f_{ns}(N, c)$ from simulations.

We compare model prediction with simulation. Figure 3.4(a) plots the critical delay $d_{ns}(N, c)$ from *ns-2* simulations versus the critical delay $d_m(N, c)$ computed from the linear model. Each data point corresponds to a particular (N, c) pair. The dashed line is where all points should lie if the linear model agrees perfectly with the simulation. Figure 3.4(b)



(a) Critical delay (ms)

(b) Critical frequency (Hz)

Figure 3.4: Linear model validation.

gives the corresponding plot for critical frequencies $f_{ns}(N, c)$ versus $f_m(N, c)$. The agreement between model and simulation seems quite reasonable (recall that delay values have a resolution of 5ms).

Consider a static link model where marking probability is a function of link flow rate

$$p_l(t) \quad = \quad f_l(y_l(t)).$$

Then the linearized model is

$$\delta p_l(t) \quad = \quad f_l'(y_l^*)\, \delta y_l(t),$$

where $f_l'(y_l^*)$ is the derivative of f_l evaluated at equilibrium. Also shown in Figure 3.4(b) are critical frequencies predicted from this static-link model (with $f_l'(y_l^*) = \rho = 0.0002$; this does not affect the critical frequency), using the same Nyquist plot method described above. It shows that queue dynamics are significant at the time-scale of interest.

Figure 3.5 illustrates the stability region implied by the linear model. For each N, it plots the critical delay $d_m(N, c)$ versus capacity c. The curve separates stable (below) from unstable regions (above). The negative slope shows that Reno/RED becomes unstable

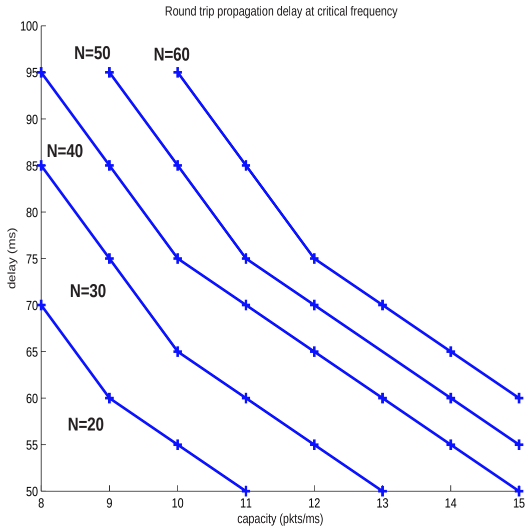


Figure 3.5: Stability region.

when delay or capacity is large. As N increases, the stability region expands, i.e., small load induces instability. Intuitively, a larger delay or capacity, or a smaller load, leads to a larger equilibrium window; this confirms the folklore that TCP behaves poorly at large window size.

3.4 Local stability analysis

We now characterize the stability region in the case of a single link with N *heterogeneous* sources. Writing forward delay as a fraction $\beta_i \in (0,1)$ of round-trip time, $\tau_i^f = \beta_i \tau_i$, and dropping link subscript l, the open-loop transfer function is

$$\begin{aligned} L(s) &= R_f(s) D_7 (sI + D_1)^{-1} D_2 R_b^T(s) (sI + D_3)^{-1} D_4 (sI + R_f(s) D_5 R^T D_6)^{-1} \\ &= \sum_i \frac{1}{\tau_i p^* (\tau_i s + p^* w_i^*)} \cdot \frac{\alpha c \rho}{s + \alpha c} \cdot \frac{1}{s + \frac{1}{c} \sum_n \frac{x_n^*}{\tau_n} e^{-\beta_i \tau_n s}} \cdot e^{-\tau_i s}. \end{aligned} \tag{3.9}$$

The first term on the right-hand side describes Reno dynamics, the second term describes RED averaging, the third term is the buffer process, and the last term represents network delay. The special case where all sources have identical round-trip times, $\tau_i = \tau$, and forward delays are zero, $\beta_i = 0$, is analyzed in [59]. They provide sufficient conditions for closed-loop stability and use them to tune RED parameters α and ρ.

We start with a lemma that collects some equilibrium properties. It can be proved directly from the fixed point of (3.4)–(3.7). Let $\overline{\tau} := \max_i \tau_i$, $\underline{\tau} := \min_i \tau_i$, $\hat{\tau} := \left(\sum_i 1/\tau_i\right)^{-1}$, and $\overline{\beta} := \max_i \beta_i$.

Lemma 3.1. *Let p^* be the equilibrium loss probability, and let w_i^* and x_i^* be the equilibrium window and rate respectively. Then $p^* = 2/(2+(c\hat{\tau})^2)$, $w_i^* = c\hat{\tau}$ for all sources i, $x_i^* = w_i^*/\tau_i$ and $\sum_i x_i^*/c = 1$.*

A sufficient condition for local stability is provided by the following theorem.

Theorem 3.1. *The closed-loop system described by (3.9) is stable if*

$$\rho \, \frac{\overline{\tau}^2}{\hat{\tau}\underline{\tau} p^{*2} w_1^*} \left(1 + \frac{1}{c\underline{\tau}\alpha} + \frac{1}{p^* w_1^*}\right) \;<\; \frac{\pi(1-\overline{\beta})^2}{\sqrt{4\overline{\beta}^2 + \pi^2(1-\overline{\beta})^2}}.$$

Proof. See [100] for detailed proof.

The left-hand side of the (sufficient) stability condition depends on network parameters (c and τ_i) as well as RED parameters (α and ρ). The right-hand side is a property of the network node that is independent of these parameters. For stability, the left-hand side must be small. This requires small capacity c and delays τ_i and large N, confirming the simulation results of the last section. To understand this, note that $c\hat{\tau}$ is the equilibrium window size of all sources. Assuming $w_1^* = c\hat{\tau} \gg 2$ so that $p = 2/w_1^{*2}$, then the stability condition can be re-written as

$$\rho \, \frac{w_1^{*3} N}{4} \left(\frac{w_1^*}{2} + 1 + \frac{N}{w_1^* \alpha}\right) \;<\; \frac{\pi(1-\overline{\beta})^2}{\sqrt{4\overline{\beta}^2 + \pi^2(1-\overline{\beta})^2}}.$$

This suggests that the system becomes unstable when window size w_1^* becomes large, agreeing with our empirical experience that TCP behaves poorly at large window size. Roughly, when c doubles, the equilibrium rate doubles, and hence the window is halved with twice the magnitude at twice the frequency, resulting in a quadratic increase in control gain and pushing the system into instability.

The dependence of the stability condition on c, τ, and N is most clearly exhibited in the case of identical sources, with $\tau = \tau_i = \overline{\tau} = \underline{\tau} = N\hat{\tau}$.

Corollary 3.1. *Suppose* $p = 2/w_1^{*2}$. *Then the stability condition in Theorem 3.1 becomes*

$$\rho\,\frac{c^3\tau^3}{4N^2}\left(\frac{c\tau}{2N}+1+\frac{1}{\alpha c\tau}\right) \quad < \quad \frac{\pi(1-\overline{\beta})^2}{\sqrt{4\overline{\beta}^2+\pi^2(1-\overline{\beta})^2}}.$$

The stability condition also suggests that a smaller ρ and a larger α enhance stability. A smaller ρ implies a larger equilibrium queue length [96]. A larger α incorporates the current queue length into the marking probability more quickly. See [100] for proofs of this Corollary.

3.5 RED parameter setting

It is suggested in [34] that the RED parameter `maxp` be dynamically adjusted: reduce `maxp` as N decreases and raise it otherwise. Raising `maxp`, or reducing `maxth-minth`, is equivalent to increasing ρ (= `maxp/(maxth-minth)`) in the direction consistent with the stability condition in Theorem 3.1. Theorem 3.1 sets an upper bound on ρ, given $N, c, and \tau$, and hence a lower bound on equilibrium queue length, to ensure stability. Adapting RED parameters *cannot* prevent the inevitable choice between stability and performance: either ρ is set small to stabilize the queue, around a large value, or, alternatively, it is set large to reduce the queue, at the risk of violent oscillation. What adaptation can hope to achieve is to dynamically find a good compromise when network condition changes.

The same stability analysis can also be applied to other AQM schemes, such as Virtual Queue [50, 85, 87] and REM/PI [5, 59], and clarifies the role of AQM. The stability proof relies on bounding a set of the form $K \cdot \text{co}\{h(v, \theta)\}$ to the right of $(-1, 0)$. The gain K and the trajectory h depend on TCP as well as AQM. For instance, for the case of a single link with capacity c shared by N identical sources with delay τ, TCP and network delay contribute a factor

$$h_{tcp} \quad = \quad \frac{e^{-jv}}{jv + p^* w_1^*}$$

to the trajectory h and a factor

$$K_{tcp} \quad = \quad \frac{c^2\tau^2}{2N} \tag{3.10}$$

to the gain K, assuming the equilibrium window is large so that $p^* = 2/w_i^2 = 2N/c\tau$. AQM compensates for the high gain introduced by TCP by shaping h and reducing K. With RED, for instance,

$$h(v,\theta) = \frac{1}{jv + \alpha c\tau} \frac{e^{-j\theta}}{v} \cdot h_{tcp}, \quad \text{and} \quad K = \frac{c\tau\alpha\rho}{1-\beta} \cdot K_{tcp}.$$

The first term in h is due to RED averaging, the second term is due to queue dynamics that also bounds $\theta \le \overline{\theta_0}$. Hence both the queue and RED add phase lag to h. More importantly, RED adds another $c\tau$ to the gain K, necessitating a small $\alpha\rho$ for stability and leading to sluggish response and large equilibrium queue. The factor $\tau/(1-\beta)$ in K comes from the queue.

The high gain K_{tcp} in (3.10) is mainly responsible for instability at high delay, high capacity or low load. It makes it difficult for any AQM algorithm to stabilize the current TCP.

3.6 Conclusion

We have presented simulation results to demonstrate that it is protocol stability more than other factors that determine the dynamics of TCP/RED. We have developed a multi-link, multi-source model that can be used to study the stability of general TCP/AQM. We have presented a sufficient stability condition for the case of a single link with heterogeneous sources and illustrated the form of Reno/RED's stability region. It implies that Reno/RED becomes unstable when the network scales up in delay or capacity. Our analysis indicates the role, and the difficulty, of RED in stabilizing Reno.

Chapter 4

Modelling and Dynamics of FAST

4.1 Introduction

Congestion control is a distributed feedback algorithm to allocate network resources among competing users. The algorithms in the current Internet, TCP Reno and its variants, have prevented severe congestion while the Internet underwent explosive growth during the last decade. In the previous chapter, we have shown that TCP Reno is ill suited for the future high-speed networks. It is well known that it does not scale as the bandwidth-delay product as the Internet continues to grow [59, 100]. This has motivated several recent proposals for congestion control of high-speed networks, including HSTCP [39], Scalable TCP [81], FAST TCP [69], and BIC TCP [163] (see [69] for extensive references). We have briefly described the motivation, background theory, and congestion window update functions of FAST TCP in Chapter 2. The details of the architecture, algorithms, extensive experimental evaluations of FAST TCP, and comparison with other TCP variants can be found in [69]. Local stability of FAST TCP in the absence of feedback delay is proved in [69] for the case of a single link. We extend the analysis to local stability with feedback delay and global stability without feedback delay, both for general networks.

Most of the stability analysis in the literature is based on the fluid model introduced in [59] (see surveys in [98, 79, 138] for extensions and related models). Key features of many of these models are that a source controls its sending rate directly[1] and that the queueing

[1] Even when the congestion window size is used as the control variable, sending rate is often taken to be the window normalized by a *constant* round-trip time, and hence a source still controls its rate directly.

delay at a link is proportional to the integral of the excess demand for its bandwidth.

In reality, a source dynamically sets its congestion window rather than its sending rate. These models do not adequately capture the self-clocking effect where a packet is sent only when an old one is acknowledged, except briefly and immediately after the congestion window is changed. This automatically constrains the input *rate* at a link to the link capacity, after a brief transient, no matter how large the congestion windows are set. Recently, a new discrete-time link model was proposed in [160, 69] to capture this effect, and detailed experimental validations have been carried out in [160]. While the traditional continuous-time link model does not consider self-clocking, the new discrete-time link model ignores the fast dynamics at the links. We first present both models of FAST TCP in Section 4.2. Experimental results are provided to show that, despite errors in these models, both of them track queueing delays reasonably well.

In Section 4.3, we prove that FAST TCP is globally stable for arbitrary networks when there is no feedback delay using the continuous-time model. We also derive a sufficient condition for local asymptotic stability for arbitrary networks with feedback delay, using the techniques developed in [125, 152]. This condition is also necessary when the sources are homogeneous with a single bottleneck link. We compare the predictions of stability based on this condition to experiments on the Dummynet Testbed with such topology. Our experiments suggest that FAST TCP is always stable for homogeneous sources with a single link, while the model with delay predicts instability when the delay is large. We conjecture that this conflict maybe due to the self-clocking effect ignored in the continuous-time model.

In Sections 4.4, we analyze the stability of FAST TCP using the discrete-time model. First, we prove that local asymptotic stability of FAST TCP in arbitrary networks in the presence of delay depends on feedback delays only through their heterogeneity. It implies in particular that a network where all sources have the same delay is always stable, no matter how large the delay is. It also confirms the common belief that a slower update enhances stability. Then we restrict ourselves to a single link without feedback delay and prove the global stability of FAST TCP. The techniques developed for this discrete-time model are new and applicable to analyzing other protocols.

Finally, we conclude in Section 4.5 with limitations of this work.

4.2 Model

4.2.1 Notation

A network consists of a set of L links indexed by l with finite capacity c_l. It is shared by a set of N flows identified by their sources indexed by i. Let R be the routing matrix where $R_{li} = 1$ if source i uses link l, and 0 otherwise.

We use t for time in the continuous model, and for time step in the discrete-time model. The meaning of t should be clear from the context. FAST TCP updates its congestion window every fixed time period, which is used as the time unit.

Let d_i denote the round-trip propagation delay of source i, and $q_i(t)$ denote the round-trip queueing delay. The round-trip time is given by $T_i(t) := d_i + q_i(t)$. We denote the forward feedback delay from source i to link l by τ_{li}^f and the backward feedback delay from link l to source i as τ_{li}^b. The sum of forward delay from source i to any link l and the backward delay from link l to source i is fixed, i.e., $\tau_i := \tau_{li}^f + \tau_{li}^b$ for any link l on the path of source i. We make a subtle assumption here. In reality, the feedback delays τ_{li}^f, τ_{li}^b include queueing delay and are time-varying. We assume for simplicity that they are constant, and mathematically unrelated to $T_i(t)$. Later, when we analyze linear stability around the network equilibrium in the presence of feedback delay, we can interpret τ_i as the equilibrium value of T_i.

Let $w_i(t)$ be source i's congestion window at time t (discrete or continuous time). The sending rate of source i at time t is defined as

$$x_i(t) = \frac{w_i(t)}{T_i(t)}, \tag{4.1}$$

where $T_i(t) = d_i + q_i(t)$. The aggregate rate at link l is

$$y_l(t) = \sum_i R_{li} x_i(t - \tau_{li}^f). \tag{4.2}$$

Let $p_l(t)$ be the queueing delay at link l. The end-to-end queueing delay $q_i(t)$ observed by source i is

$$q_i(t) = \sum_l R_{li} p_l(t - \tau_{li}^b). \tag{4.3}$$

4.2.2 Discrete and continuous-time models

FAST TCP source periodically updates its congestion window w based on the average RTT and estimated queueing delay. The pseudo-code is

$$\texttt{w} \leftarrow (1-\gamma)\texttt{w} \;+\; \gamma \left(\frac{\texttt{baseRTT}}{\texttt{RTT}} \texttt{w} + \alpha \right),$$

where $\gamma \in (0, 1]$, `baseRTT` is the minimum RTT observed, and α is a constant. We model this by the following discrete time equation

$$w_i(t+1) = \gamma \left(\frac{d_i w_i(t)}{d_i + q_i(t)} + \alpha_i \right) + (1-\gamma) w_i(t), \tag{4.4}$$

where $w_i(t)$ is the congestion window of the ith source, $\gamma \in (0, 1]$, and α_i is a constant for source i. The corresponding continuous-time model is

$$\dot{w}_i(t) = \gamma \left(\frac{d_i w_i(t)}{d_i + q_i(t)} + \alpha_i - w_i(t) \right), \tag{4.5}$$

where the time is measured in the unit of update period in FAST TCP.

For the continuous-time model, queueing delay has been traditionally modelled with

$$\dot{p}_l(t) = \frac{1}{c_l}(y_l(t) - c_l). \tag{4.6}$$

However, TCP uses self-clocking: the source always tries to maintain that the number of packets in fly equals to the congestion window size. When the congestion window is fixed, the source will send a new packet exactly after it receives an ACK packet. When the congestion window changes, the source sends out bulk traffic in burst, or sends nothing in a short time period. Therefore, one round-trip time after a congestion window is changed, packet transmission will be clocked at the same rate as the throughput the flow receives.

We assume that the disturbance in the queues due to congestion window changes settles down quickly compared with the update period of the discrete-time model; see [160] for detailed justification and validation experiments for these arguments. A consequence of this assumption is that the link queueing delay vector, $p(t) = (p_l(t)$, for all $l)$, is determined implicitly by sources' congestion windows in a static manner

$$\sum_i R_{li} \frac{w_i(t-\tau_{li}^f)}{d_i + q_i(t-\tau_{li}^f)} \begin{cases} = c_l & \text{if } p_l(t) > 0 \\ \leq c_l & \text{if } p_l(t) = 0 \end{cases}, \tag{4.7}$$

where the q_i is the end-to-end queueing delay given by (4.3).

In summary, the continuous-time model is specified by (4.5) and (4.6), and the discrete-time model is specified by (4.4) and (4.7), where the source rates and aggregate rates at links are given by (4.1) and (4.2), and the end-to-end delays are given by (4.3). While the continuous-time model does not take self-clocking into full account, the discrete-time model ignores the fast dynamics at the links. Before comparing these models, we clarify their common equilibrium structure by the following theorem cited from [69].

Theorem 4.1. *Suppose that the routing matrix R has full row rank. A unique equilibrium (x^*, p^*) of the network exists, and x^* is the unique maximizer of*

$$\max_{x \geq 0} \sum_i \alpha_i \log x_i \quad \textit{s.t.} \quad Rx \leq c \tag{4.8}$$

with p^* as the corresponding optimum of its Lagrangian dual. This implies in particular that the equilibrium rate x^* is α_i-weighted proportionally fair.

4.2.3 Validation

The continuous-time link model implies that the queue takes an infinite amount of time to converge after a window change. In the other extreme, the discrete-time link model assumes that the queue settles down in one sampling time. Neither is perfect, but we now present experimental results that suggest both track the queue dynamics well.

All the experiments reported in this paper are carried out on the Dummynet Testbed

[132]. A FreeBSD machine is configured as a Dummynet router that provides different propagation delays for different sources. It can be configured with different capacity and buffer size. In our experiments, the bottleneck link capacity is 800Mbps, and the buffer size is 4000 packets with a fixed packet length of 1500 bytes. A Dummynet monitor records the queue size every 0.4 second. The congestion window size and RTT are recorded at the host every 50ms. TCP traffic is generated using *iperf*. The publicly released code of FAST [33] is used in all experiments involving FAST. We present two experiments to validate the model, one closed-loop and one open-loop.

In the first (closed-loop) experiment, there are 3 FAST TCP sources sharing a Dummynet router with a common propagation delay of 100ms. The measured and predicted queue sizes are given in Figure 4.1. In the beginning of the experiment, the FAST sources are in the slow start phase, and none of the models gives accurate prediction. After the FAST TCP enters the congestion avoidance phase, both models track the queue size well.

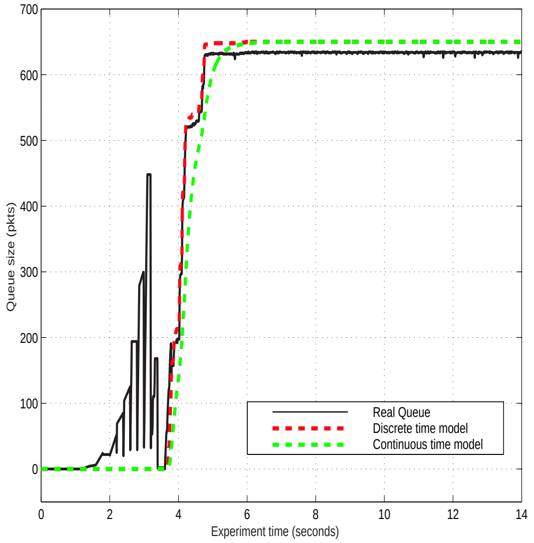
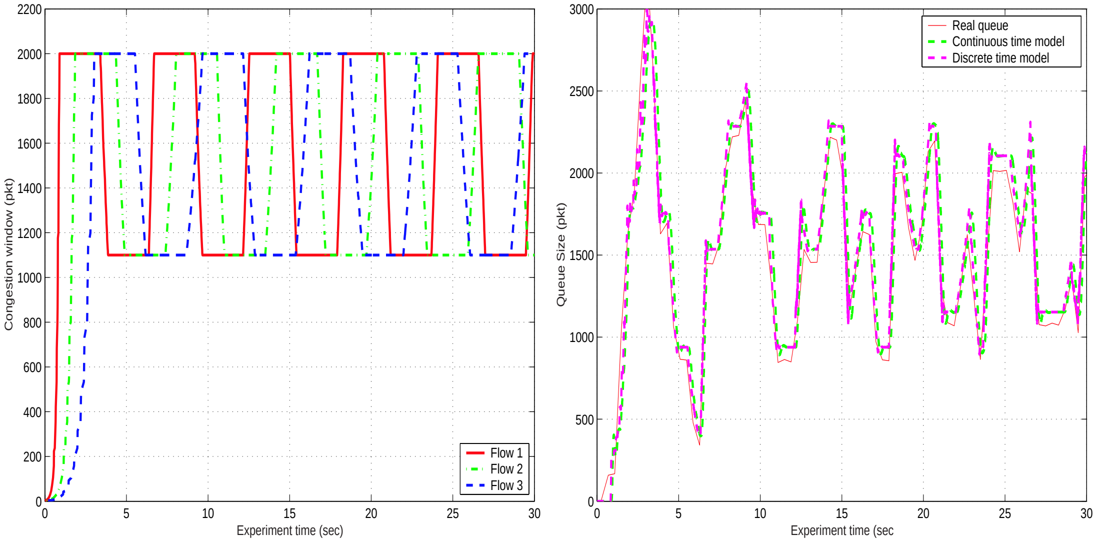


Figure 4.1: Model validation–closed loop

To eliminate the modelling error in the congestion window adjustment algorithm itself while validating the link models, we decouple the TCP and queue dynamics by using open-loop, window control. The second experiment involves three sources with propagation delays 50ms, 100ms, and 150ms sharing the same Dummynet router.

We changed the Linux 2.4.19 kernel so that the sources vary their window sizes according to the schedules shown in Figure 4.2(a). The sequences of congestion window sizes are then used in (4.1)–(4.2) and (4.6) to compute the queueing delay predicted by the continuous-time model. We also use them in (4.1)–(4.2) and (4.7) to compute the predictions of the discrete-time model. The queueing delay measured from the Dummynet and those predicted by these two models are shown in Figure 4.2(b), which indicates that both models track the queue sizes well. We next analyze the stability properties of these two models.



(a) Scheduled congestion windows

(b) Resulting queue size

Figure 4.2: Model validation–open loop.

4.3 Stability analysis with the continuous-time model

We present the stability analysis of the continuous model in general networks with and without feedback delays.

4.3.1 Global stability without feedback delay

In this subsection, we show that FAST is globally stable for general networks by designing a Lyapunov. When there is no feedback delay, the equations (4.2) and (4.3) can be simplified as

$$y_l(t) = \sum_i R_{li} x_i(t) \quad \text{and} \quad q_i(t) = \sum_l R_{li} p_l(t). \tag{4.9}$$

Suppose that R is full row rank, and the system has unique equilibrium source rates and link prices. Let $w_i, p_l, \ldots$ be the equilibrium quantities, and denote $\delta w_i(t) := w_i(t) - w_i, \delta p_l(t) = p_l(t) - p_l, \ldots$. From (4.5) we can formulate the equilibrium window as

$$w_i = \frac{\alpha_i T_i}{q_i}, \tag{4.10}$$

where T_i is the equilibrium round-trip delay $T_i = d_i + q_i$.

Based on (4.5) and (4.10), we can write the derivative of $w_i(t)$ as

$$\begin{aligned} \frac{1}{\gamma}\dot{w}_i(t) &= \alpha_i - \frac{q_i(t) w_i(t)}{T_i(t)}, \\ &= \alpha_i - \frac{q_i(t)}{T_i(t)}(w_i + \delta w_i(t)), \\ &= -\frac{q_i(t)}{T_i(t)}\delta w_i(t) + \alpha_i \frac{T_i(t) q_i - q_i(t) T_i}{T_i(t) q_i}, \\ &= -\frac{q_i(t)}{T_i(t)}\delta w_i(t) - \alpha_i \frac{d_i \delta q_i(t)}{T_i(t) q_i}. \end{aligned}$$

Therefore, we have

$$\frac{1}{\gamma}\delta\dot{w}_i(t) = -\frac{q_i(t)}{T_i(t)}\delta w_i(t) - \frac{\alpha_i d_i}{T_i(t) q_i}\delta q_i(t). \tag{4.11}$$

Based on (4.1) and (4.10) we have

$$\delta x_i(t) = \frac{w_i + \delta w_i(t)}{T_i(t)} - \frac{w_i}{T_i} = \frac{\delta w_i(t)}{T_i(t)} - (\frac{1}{T_i} - \frac{1}{T_i(t)}) w_i = \frac{\delta w_i(t)}{T_i(t)} - \frac{\delta q_i(t)}{T_i(t) T_i}\frac{\alpha_i T_i}{q_i}.$$

Therefore, we have

$$\delta x_i(t) = \frac{1}{T_i(t)}\delta w_i(t) - \frac{\alpha_i}{T_i(t) q_i}\delta q_i(t). \tag{4.12}$$

Based on (4.12) and (4.9), the derivative of link price is

$$\dot{p}_l(t) = \frac{1}{c_l}(\sum_i R_{li}x_i(t) - c_l) = \frac{1}{c_l}\sum_i R_{li}\delta x_i(t). \tag{4.13}$$

From (4.12) and (4.13), we have

$$\delta\dot{p}_l(t) = \frac{1}{c_l}\sum_i R_{li}\left(\frac{1}{T_i(t)}\delta w_i(t) - \frac{\alpha_i}{T_i(t)q_i}\delta q_i(t)\right). \tag{4.14}$$

With the preliminary results, we present and prove the following theorem.

Theorem 4.2. *The continues-time model of FAST TCP is globally asymptotically stable when there is no feedback delay and R is full row rank.*

Proof: Considering the function $V(w(t), p(t))$ defined as

$$V(w(t), p(t)) = \frac{1}{2\gamma}\sum_i \frac{q_i}{\alpha_i d_i}(w_i(t) - w_i)^2 + \frac{1}{2}\sum_l c_l(p_l(t) - p_l)^2. \tag{4.15}$$

Clearly, the function $V(w, p)$ is non-negative for all $(w(t), p(t))$. It is zero if and only if $w(t) = w$ and $p(t) = p$, where the system is at its equilibrium. Differentiating $V(w(t), p(t))$ with respect to the solution trajectory using (4.14) and (4.11) yields

$$\begin{aligned}
\dot{V}(w(t), p(t)) &= \sum_i \frac{q_i}{\gamma\alpha_i d_i}\delta w_i(t)\delta\dot{w}_i(t) + \sum_l c_l\delta p_l(t)\delta\dot{p}_l(t), \\
&= \sum_i \frac{q_i}{\alpha_i d_i}\delta w_i(t)\left(-\frac{q_i(t)}{T_i(t)}\delta w_i(t) - \frac{\alpha_i d_i}{T_i(t)q_i}\delta q_i(t)\right) \\
&\quad + \sum_l\sum_i R_{li}\left(\frac{1}{T_i(t)}\delta w_i(t) - \frac{\alpha_i}{T_i(t)q_i}\delta q_i(t)\right)\delta p_l(t), \\
&= -\sum_i \frac{q_i q_i(t)}{T_i(t)\alpha_i d_i}\delta w_i(t)^2 - \sum_i \frac{1}{T_i(t)}\delta w_i(t)\delta q_i(t) \\
&\quad + \sum_i \frac{1}{T_i(t)}\delta w_i(t)\sum_l R_{li}\delta p_l(t) - \sum_i \frac{\alpha_i}{T_i(t)q_i}\delta q_i(t)\sum_l R_{li}\delta p_l(t), \\
&= -\sum_i \frac{q_i^* q_i(t)}{T_i(t)\alpha_i d_i}\delta w_i(t)^2 - \sum_i \frac{\alpha_i}{T_i(t)q_i}\delta q_i(t)^2
\end{aligned}$$

From the above equation, $\dot{V}(w(t), p(t)) \leq 0$. Since $\alpha > 0, q_i > 0$ and $T_i(t) > d_i > 0$, if the equality holds, $\delta q_i(t) = 0$, which also implies that $\delta w_i(t) = 0$. Therefore $\dot{V}(w(t), p(t)) = 0$, if and only if the source rates are at equilibrium. Therefore, the system is at its unique equilibrium under our assumption that R is full row rank.

From the above argument, $V(w(t), p(t))$ is a system Lyapunov function, and the system is globally asymptotically stable. □

From the proof, it is clear that $\dot{V}(w(t), p(t)) = 0$ only implies that the source rates are in equilibrium. When R is not full rank, the source rates still globally converge to their equilibrium values, but the equilibrium link price p is no longer unique and may not converge to a fixed value.

In our continuous model, we ignored that positive projection in the link price updates (i.e., the link prices have to be non-negative). This is equivalent to assuming that the bottleneck link is unchanged in this dynamical system. We need to consider this saturation problem in our future research.

4.3.2 Local stability with feedback delay

When feedback delays are present, the global stability analysis for FAST TCP in general networks is still open. In this section, we try to provide a condition to ensure local stability.

Since there exists a unique equilibrium as described in Theorem 4.1, we can linearize the model (4.5) and (4.6) around this equilibrium. Define routing matrices with feedback delay in frequency domain as

$$(R_f(s))_{li} := \begin{cases} e^{-\tau_{li}^f s} & \text{if } R_{li} = 1 \\ 0 & \text{if } R_{li} = 0 \end{cases} \qquad (R_b(s))_{li} := \begin{cases} e^{-\tau_{li}^b s} & \text{if } R_{li} = 1 \\ 0 & \text{if } R_{li} = 0 \end{cases} .$$

Let x_i, q_i, and T_i be the corresponding equilibrium values associated with source i. The following Lemma provides the open-loop transfer function.

Lemma 4.1. *The open-loop transfer function of the linearized FAST TCP system is*

$$L(s) = D_3 R_f(s) \Lambda(s) X R_f^T(-s), \tag{4.16}$$

where

$$D_3 := \mathit{diag}\left(\frac{1}{c_l}\right), \quad X := \mathit{diag}(x_i), \ \textit{and} \ \Lambda(s) := \mathit{diag}(\frac{e^{-\tau_i s}}{T_i s}\frac{T_i s + \gamma T_i}{T_i s + \gamma q_i}).$$

Proof. See Appendix 4.6.1. □

The following theorem provides a sufficient condition for local stability.

Theorem 4.3. *The FAST TCP system described by (4.5) and (4.6) is locally stable if*

$$\frac{M}{\phi}\sqrt{\frac{\phi^2 + \gamma^2 T_{\max}^2}{\phi^2 + \gamma^2 q_{\min}^2}} < 1, \tag{4.17}$$

where M is the maximal number of links in the path of any source, $q_{\min} = \min_i q_i$, $T_{\max} = \max_i T_i$ and

$$\phi := \min_i \left(\frac{\pi}{2} - \tan^{-1}\frac{1 - q_i/T_i}{2\sqrt{q_i/T_i}}\right). \tag{4.18}$$

Proof. See the Appendix 4.6.2. □

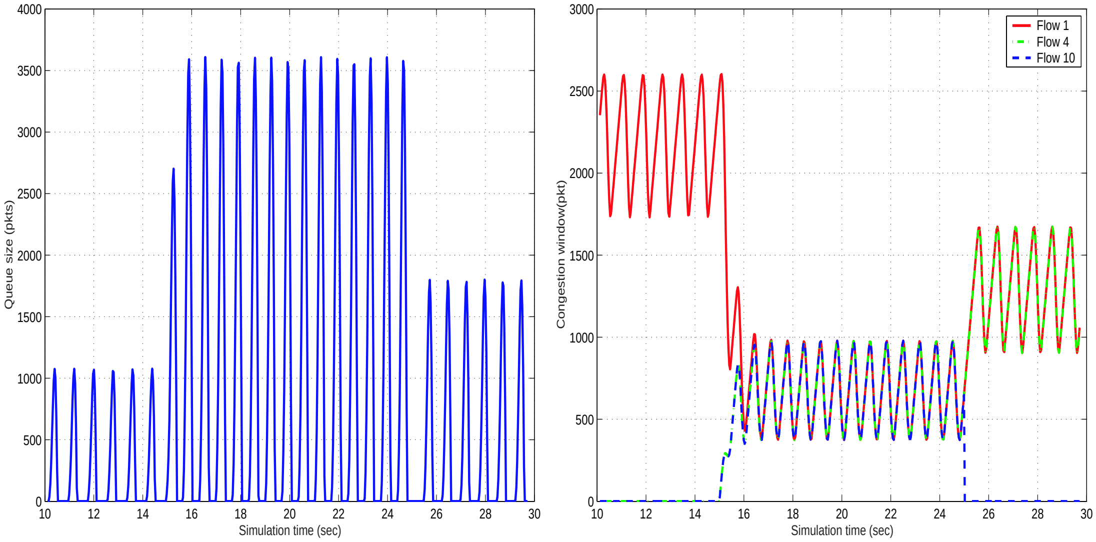
This is actually a very weak theorem. The condition (4.18) can hardly be satisfied when M is large. But it can provide us with some information about the effect of various parameters on the stability. For example, this condition suggests that the equilibrium queueing delay should be large to guarantee stability. In general, this condition is only sufficient. When there is only one link and all sources have the same feedback delays, it becomes necessary as well. Our numerical simulations for this model validate this.

4.3.3 Numerical simulation and experiment

The condition in Theorem 4.3 implies that FAST TCP may become unstable in a single bottleneck network with homogeneous sources. However our experiments with FAST TCP on the Dummynet Testbed have always been stable.

We now present an experiment that violates the local stability condition. Moreover, numerical simulation of the continuous-time model exhibits instability. Yet, the same network on Dummynet is clearly stable. This suggests that the discrepancy is not in the stability theorem but rather in the continuous-time model.

In our experiment, the sources have identical propagation delay of 100ms with a constant α value of 70 packets. They share a bottleneck with capacity of 800Mbps. The simulations and experiments consist of three intervals. The interval length is 10 seconds for the continuous-time model simulation and 100 seconds for the experiment[2]. Three sources are active from the beginning of the experiment, seven additional sources activate in the second interval, and in the last interval, all sources become inactive except five of them. The simulation and experimental results are shown in Figure 4.3 and Figure 4.4, respectively.



(a) Queue size

(b) Window size

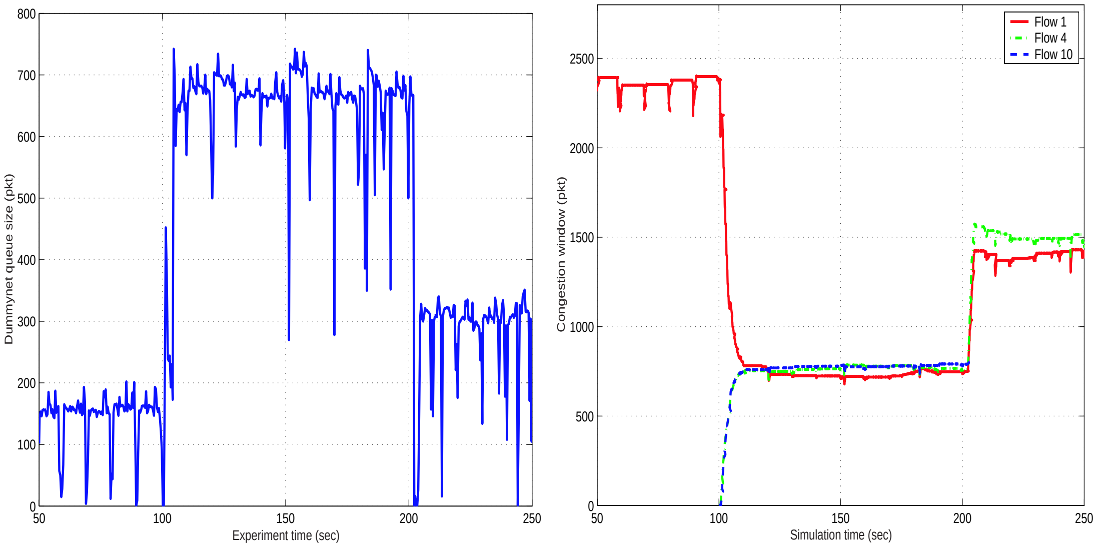
Figure 4.3: Numerical simulations of FAST TCP.

The stability condition in Theorem 4.3 is not satisfied and, as expected, the numerical simulation based on the continuous-time model exhibits periodic oscillation. However, in the Dummynet experiment, FAST TCP is actually stable (see Figure 4.4).[3]

We believe that the discrepancy is largely due to the fact that the continuous-time model does not capture the self-clocking effect accurately. Self-clocking ensures that packets are

[2] We use a long duration in the Dummynet experiment because a FAST TCP source takes longer to converge due to slow-start, which is not included in our model.

[3] The regular spikes every 10 seconds in the queue size are probably due to a certain background task in the sending host.



(a) Queue size

(b) Window size

Figure 4.4: Dummynet experiments of FAST TCP.

sent at the same rate as the goodput the source receives, except briefly when the window size changes. This self-clocking feature can actually help the system approach an equilibrium. Indeed, for the case of one source for one link, a discrete-event model is used in [160] to prove that TCP FAST and Vegas are always stable regardless of the feedback delay. It also provides justification for the discrete-time models in (4.4) and (4.7) based on the self-clocking feature introduced in the last section.

4.4 Stability analysis with the discrete-time model

We now analyze the stability of this model. We will see that the discrete-time model predicts that a network of homogeneous sources with the same feedback delay is locally stable no matter how large the delay is, agreeing with our experimental experience. In the following subsection, we study the local stability of FAST TCP using the discrete-time model for arbitrary networks with feedback delays.

4.4.1 Local stability with feedback delay

A network of FAST TCP sources is modelled by equations (4.3), (4.4), and (4.7). This generalizes the model in [69] by including feedback delay. When local stability is studied, we ignore all un-congested links (links where prices are zero in equilibrium) and assume that equality always holds in (4.7).

The main result of this section provides a sufficient condition for local asymptotic stability in general networks with common feedback delay.

Theorem 4.4. *FAST TCP is locally stable for arbitrary networks if* $\gamma \in (0,1]$ *and if all sources have the same round-trip feedback delay* $\tau_i = \tau$ *for all* i.

The stability condition in the theorem does not depend on the value of the feedback delay, but only on the heterogeneity among them. In particular, when all feedback delays are ignored, $\tau_i = 0$ for all i, then FAST TCP is locally asymptotically. This generalizes the stability result in [69].

Corollary 4.1. *FAST TCP is locally asymptotically stable in the absence of feedback delay for general networks with any* $\gamma \in [0,1)$.

The rest of this subsection is devoted to the proof of Theorem 4.4.

We apply Z-transform to the linearized system and use the generalized Nyquist criterion to derive a sufficient stability condition. Define the forward and backward Z-transformed routing matrices $R_f(z)$ and $R_b(z)$ as

$$(R_f(z))_{li} := \begin{cases} z^{-\tau^f_{li}} & \text{if } R_{li} = 1 \\ 0 & \text{if } R_{li} = 0 \end{cases} \quad \text{and} \quad (R_b(z))_{li} := \begin{cases} z^{-\tau^b_{li}} & \text{if } R_{li} = 1 \\ 0 & \text{if } R_{li} = 0 \end{cases} .$$

The relation $\tau^f_{li} + \tau^b_{li} = \tau_i$ gives

$$R_b(z) = R_f(z^{-1}) \cdot \text{diag}(z^{-\tau_i}). \tag{4.19}$$

Denote $W(z)$, $Q(z)$, and $P(z)$ as the corresponding Z-transforms of $\delta w(t)$, $\delta q(t)$, and $\delta p(t)$ for the linearized system. Let q and w be the end-to-end queueing delay and congestion window at equilibrium. Linearizing (4.7) yields

$$\sum_i R_{li} \left(\frac{\delta w_i(t-\tau_{li}^f)}{d_i+q_i} - w_i \frac{\delta q_i(t-\tau_{li}^f)}{(d_i+q_i)^2} \right) = 0,$$

where the equality is used in (4.7). The corresponding Z-transform in matrix form is

$$R_f(z)D^{-1}MW(z) - R_f(z)BQ(z) = 0, \tag{4.20}$$

where the diagonal matrices B, D, and M are

$$B := \text{diag}\left(\frac{w_i}{(d_i+q_i)^2}\right), M := \text{diag}\left(\frac{d_i}{d_i+q_i}\right), \text{ and } D := \text{diag}(d_i).$$

Since $R_f(z)$ is generally not a square matrix, we cannot cancel it in (4.20).

Equation (4.3) is already linear, and the corresponding Z-transform in matrix form is

$$Q(z) = R_b(z)^T P(z). \tag{4.21}$$

By combining (4.20) and (4.21), we obtain

$$\begin{pmatrix} I & -R_b^T(z) \\ R_f(z)B & 0 \end{pmatrix} \begin{pmatrix} Q(z) \\ P(z) \end{pmatrix} = \begin{pmatrix} 0 \\ R_f(z)D^{-1}M \end{pmatrix} W(z).$$

Solving this equation with block matrix inverse gives the transfer function from $W(z)$ to $Q(z)$

$$\frac{Q(z)}{W(z)} = R_b^T(z)(R_f(z)BR_b^T(z))^{-1}R_f(z)D^{-1}M.$$

The Z-transform of the linearized, congestion window update algorithm is

$$zW(z) = \gamma\left(MW(z) - DBQ(z)\right) + (1-\gamma)W(z).$$

By combining the above equations, we get the open-loop transfer function $L(z)$ from $W(z)$

to $W(z)$ as

$$L(z) \;=\; -\left(\gamma\left(M - DBR_b^T(z)(R_f(z)BR_b^T(z))^{-1}R_f(z)D^{-1}M\right) + (1-\gamma)I\right)z^{-1}.$$

A sufficient condition for local stability can be developed based on the generalized Nyquist criterion [23, 31]. Since the open-loop system is stable, if we can show that the eigenvalue loci of $L(e^{jw})$ does not enclose -1 for $\omega \in [0, 2\pi)$, the closed-loop system is stable. Therefore, if the spectral radius of $L(e^{jw})$ is strictly less than 1 for $\omega \in [0, 2\pi)$, the system will be stable.

When $z = e^{jw}$, the spectral radii of $L(z)$ and $-zL(z)$ are the same. Hence, we only need to study the spectral radius of

$$J(z) : \;=\; \gamma(M - DBR_b^T(z)\left(R_f(z)BR_b^T(z)\right)^{-1}R_f(z)D^{-1}M + (1-\gamma)I.$$

Clearly, the eigenvalues of $J(z)$ are dependent on γ. For any given $z = e^{j\omega}$, let the eigenvalues of $J(z)$ be denoted by $\lambda_i(\gamma)$, $i = 1 \ldots N$, as functions of $\gamma \in (0, 1]$. It is clear that

$$|\lambda_i(\gamma)| \;=\; |\gamma\lambda_i(1) + (1-\gamma)| \le \gamma|\lambda_i(1)| + (1-\gamma).$$

Hence if $\rho(J(z)) < 1$ for any $z = e^{j\omega}$ for $\gamma = 1$, it will also hold for all $\gamma \in (0, 1]$. Therefore, it suffices to study the stability condition for $\gamma = 1$.

Let μ_i be the ith diagonal entry of matrix M with $\mu_i = d_i/(d_i + q_i)$. Denote $\mu_{\max} := \max_i \mu_i$. Since the end-to-end queueing delay q_i cannot be zero at equilibrium (otherwise the rate will be infinitely large), we have $q_i > 0$ and $\mu_{\max} < 1$. The following lemma characterizes the eigenvalues of $J(z)$ with $\gamma = 1$.

Lemma 4.2. *When $z = e^{j\omega}$ with $\omega \in [0, 2\pi)$ and $\gamma = 1$, the eigenvalues of $J(z)$ have the following properties:*

1. *There are L zero eigenvalues with the corresponding eigenvectors as the columns of matrix $M^{-1}DBR_b^T(z)$.*

2. *The nonzero eigenvalues have moduli less than* 1 *if* $\tau_{\max} - \tau_{\min} < 1/4$*, where* $\tau_{\max} = \max_i \tau_i$ *and* $\tau_{\min} = \min_i \tau_i$ *.*

Proof: At $\gamma = 1$, the matrix $J(z)$ is

$$M - DBR_b^T(z)(R_f(z)BR_b^T(z))^{-1}R_f(z)D^{-1}M.$$

It is easy to check that

$$J(z)M^{-1}DBR_b^T(z) = DBR_b^T(z) - DBR_b^T(z) = 0.$$

Since $M^{-1}DBR_b^T(z)$ has full column rank, it consists of L linearly independent eigenvectors of $J(z)$ with corresponding eigenvalue 0. This proves the first assertion.

For the second assertion, suppose that λ is an eigenvalue of $J(z)$ for a given z. Define matrix A as

$$A : \;=\; J(z) - \lambda I = (M - \lambda I) - DBR_b^T(z)(R_f(z)BR_b^T(z))^{-1}R_f(z)D^{-1}M,$$

which is singular by definition. Based on the matrix inversion formula (see, e.g., [62])

$$(J \;+\; EHS)^{-1} = J^{-1} - J^{-1}E(H^{-1} + SJ^{-1}E)^{-1}SJ^{-1},$$

if $J + EHS$ is singular, then either J or $H^{-1} + SJ^{-1}E$ is singular. We can let

$$J := M - \lambda I, \;\; E := -DBR_b^T(z), \;\; H := (R_f(z)BR_b^T(z))^{-1}, \;\; \text{and } S := R_f(z)D^{-1}M.$$

Since $A = J + EHS$ is singular, either $J = M - \lambda I$ or $H^{-1} + SJ^{-1}E$ is singular. The second term can be reformulated into $R_f(z)(B - M(M - \lambda I)^{-1}B)R_b^T(z)$.

Case 1: $M - \lambda I$ is singular. Since M is diagonal, then

$$0 < \lambda = \frac{d_i}{d_i + q_i} = \mu_i \leq \mu_{\max} < 1.$$

Case 2: $R_f(z)(B - M(M - \lambda I)^{-1}B)R_b^T(z)$ is singular.

It is clear that

$$B - M(M - \lambda I)^{-1}B = \text{diag}\left(1 - \mu_i(\mu_i - \lambda)^{-1}\beta_i\right) = -\lambda\text{diag}\left(\frac{\beta_i}{\mu_i - \lambda}\right),$$

where β_i is the ith diagonal entry of matrix B. Hence, $\lambda = 0$ is always an eigenvalue, which is claimed before. If λ is nonzero, it has to be true that

$$\det\left(R_f(z)\text{diag}\left(\frac{\beta_i}{\mu_i - \lambda}\right)R_b^T(z)\right) = 0. \tag{4.22}$$

When $z = e^{j\omega}$, we have $z^{-1} = \overline{z}$. Hence, equation (4.19) can be rewritten as

$$R_b^T(z) = \text{diag}(z^{-\tau_i})R_f^T(\overline{z}) = \text{diag}(z^{-\tau_i})R_f^*(z).$$

Substituting the above equation into (4.22) with $z = e^{j\omega}$ yields

$$\det\left(R_f(z)\text{diag}\left(\frac{e^{-j\omega\tau_i}\beta_i}{\mu_i - \lambda}\right)R_f^*(z)\right) = 0. \tag{4.23}$$

Therefore, the following formula is also zero

$$e^{-j(\omega\tau_{\max}+\psi)}\det\left(R_f(z)\text{diag}\left(\frac{e^{j(\theta_i+\psi)}\beta_i}{\mu_i - \lambda}\right)R_f^*(z)\right) = 0.$$

where $\theta_i = (\tau_{\max} - \tau_i)\omega$, and ψ can be any value. When $\tau_{\max} - \tau_{\min} < 1/4$, we have

$$0 \le \theta_i = (\tau_{\max} - \tau_i)\omega \le \pi/2.$$

Suppose that there is a solution such that $|\lambda| \ge 1$. Based on Lemma 4.3, which will be presented later, there exists a ψ s.t. Im$\left(\text{diag}\left(e^{j(\theta_i+\psi)}\beta_i/(\mu_i - \lambda)\right)\right)$ is a positive diagonal matrix. Therefore the imaginary part of matrix $R_f(z)\text{diag}\left(e^{j(\theta_i+\psi)}\beta_i/(\mu_i - \lambda)\right)\right) R_f^*(z)$ is positive definite, and the real part is symmetric. From Lemma 4.4 below, it has to be nonsingular. This contradicts the equation

$$\det\left(R_f(z)\text{diag}\left(\frac{e^{j(\theta_i+\psi)}\beta_i}{\mu_i - \lambda}\right)R_f^*(z)\right) = 0.$$

Hence, we have $|\lambda| < 1$. □

The proof of Theorem 4.4 will be complete after the next two lemmas.

Lemma 4.3. *Suppose that* $0 < \mu_i < 1$ *and* $0 \leq \theta_i < \pi/2$. *If* $|\lambda| \geq 1$ *, there exists a* ψ *such that*

$$\mathit{Im}\left(\frac{e^{j(\theta_i+\psi)}\beta_i}{\mu_i - \lambda}\right) > 0 \ \ \mathit{for} \quad i = 1 \ldots N.$$

Proof: See Appendix 4.6.3. □

Lemma 4.4. *If the real part of a complex matrix is symmetric, and the imaginary part is positive definite, then the matrix is nonsingular.*

Proof: See Appendix 4.6.4. □

4.4.2 Global stability for one link without feedback delay

In the absence of feedback delay, when there is only one link, the FAST TCP model can be simplified into

$$w_i(t+1) = \gamma\left(\frac{d_i w_i(t)}{d_i + q(t)} + \alpha_i\right) + (1-\gamma)w_i(t), \tag{4.24}$$

$$\sum_i \frac{w_i(t)}{d_i + q(t)} \leq c \quad \text{with equality if } q(t) > 0, \tag{4.25}$$

where $q(t)$ is the queueing delay at the link (subscript is omitted). The main result of this section proves that the above system (4.24)–(4.25) is globally asymptotically stable and converges to the equilibrium exponentially fast starting from any initial value.

Theorem 4.5. *On a single link, FAST TCP converges exponentially to the equilibrium, in the absence of feedback delay.*

In the rest of this section, we prove the theorem in several steps. The first result is that after finite steps K_1, equality always holds in (4.25) and $q(t) > 0$ for any $t > K_1$. Define the normalized congestion window sum as $Y(t) := \sum_i w_i(t)/d_i$. From (4.25), it is clear that $q(t) > 0$ if and only if $Y(t) > c$.

Lemma 4.5. *There exists $K_1 > 0$ such that the following claims are true for all $t > K_1$:*

1. $q(t) > 0$.
2. $\nu(t+1) = (1-\gamma)\nu(t)$ *where* $\nu(t) := Y(t) - c - \sum_i \alpha_i/d_i$.

Proof: If initially $q(t) = 0$, which also means $Y(t) \le c$, from (4.24) we have $Y(t+1) = Y(t) + \gamma \sum_i \alpha_i/d_i$, which linearly increases with t. Then $Y(t) > c$ after some finite steps. Therefore, there exists a K_1 such that $Y(t) > c$ and $q(t) > 0$ at $t = K_1$.

We will show that $Y(t) > c$ implies $Y(t+1) > c$. Hence $q(t) > 0$ for all $t > K_1$. Moreover, $\nu(t)$ converges exponentially to 0.

Suppose $Y(t) > c$. From $\sum_i w_i(t)/(d_i + q_i(t)) = c$, we have

$$\begin{aligned}
\nu(t+1) &= \sum_i \frac{w_i(t+1)}{d_i} - \sum_i \frac{\alpha_i}{d_i} - c, \\
&= (1-\gamma)\sum_i \frac{w_i(t) - \alpha_i}{d_i} + \gamma \sum_i \frac{w_i(t)}{d_i + q(t)} - c, \\
&= (1-\gamma)\left(\sum_i \frac{w_i}{d_i} - c - \sum_i \frac{\alpha_i}{d_i}\right) = (1-\gamma)\,\nu(t).
\end{aligned}$$

This proves the second assertion. Moreover it implies

$$Y(t+1) = (1-\gamma)Y(t) + \gamma\left(\sum_i \frac{\alpha_i}{d_i} + c\right).$$

Hence, $Y(t) > c$ implies $Y(t+1) > c$ and $q(t+1) > 0$. This completes the proof. □

For the rest of this subsection, we pick a fixed ϵ with $0 < \epsilon < \sum_i \alpha_i/d_i$. Define

$$q_{\min} := \frac{d_{\min}}{c}\left(\sum_i \frac{\alpha_i}{d_i} - \epsilon\right), \quad \text{and} \quad q_{\max} := \frac{d_{\max}}{c}\left(\sum_i \frac{\alpha_i}{d_i} + \epsilon\right),$$

where $d_{\min} := \min_i d_i$ and $d_{\max} := \max_i d_i$.

Then $q(t)$ is bounded by these two values after finite steps.

Lemma 4.6. *There exists a positive K_2 such that $q_{\min} \leq q(t) \leq q_{\max}$ for any $t \geq K_2$.*

Proof: From Lemma 4.5, after finite steps K_1, $\nu(t+1) = (1-\gamma)\nu(t)$. Therefore, there exists a K_2 such that $|\nu(t)| < \epsilon$ for all $t \geq K_2$. It implies

$$\begin{aligned}\sum_i \frac{\alpha_i}{d_i} &< \sum_i \frac{w_i(t)}{d_i} - c + \epsilon = \sum_i \left(\frac{w_i(t)}{d_i} - \frac{w_i(t)}{d_i + q(t)} \right) + \epsilon, \\ &\leq \sum_i \frac{q(t) w_i(t)}{d_{\min}(d_i + q(t))} + \epsilon = \frac{q(t)c}{d_{\min}} + \epsilon.\end{aligned}$$

.

Therefore,

$$q(t) \geq \frac{d_{\min}}{c} \left(\sum_i \frac{\alpha_i}{d_i} - \epsilon \right) = q_{\min}.$$

The proof for $q_{\max}$ is the same. □

Define $\mu_i(t) := d_i/(d_i + q(t))$, and denote $\mu_{\max} := \max_i d_i/(d_i + q_{\min})$, $\mu_{\min} := \min_i d_i/(d_i + q_{\max})$. Based on Lemma 4.6, we have $1 > \mu_{\max} \geq \mu_i(t) > \mu_{\min} > 0$ for any $t \geq K_2$. Define

$$\eta_i(t) := \frac{w_i(t) - \alpha_i}{\alpha_i d_i} - \frac{1}{q(t)}, \tag{4.26}$$

and denote $\eta_{\max}(t) := \max_i \eta_i(t)$, $\eta_{\min}(t) := \min_i \eta_i(t)$. We will show that the window update for source i is proportional to $\eta_i(t)$, and the system is at equilibrium if and only if all $\eta_i(t)$ are zero. The next lemma gives bounds on $\eta_i(t)$.

Lemma 4.7. *There exist two positive numbers δ_1 and δ_2 such that for all $t \geq K_2$*

$$\eta_{\max}(t) > -\delta_1(1-\gamma)^t \quad \textit{and} \quad \eta_{\min}(t) < \delta_2(1-\gamma)^t.$$

Proof: From (4.26), it is easy to check that $Y(t+1) - Y(t) = -\gamma\nu(t)$. By Lemma 4.5, when $t \geq K_2$ we have

$$Y(t+1) - Y(t) = -\gamma\nu(t) \leq \gamma(1-\gamma)^{t-K_2}|\nu(K_2)| = \kappa(1-\gamma)^t, \tag{4.27}$$

where $\kappa := \gamma(1-\gamma)^{-K_2}|\nu(K_2)|$.

The update of source i's congestion window is

$$\begin{aligned} w_i(t+1) - w_i(t) &= \gamma\left(\frac{d_i w_i(t)}{d_i + q(t)} + \alpha_i - w_i(t)\right) = \frac{\gamma q(t)}{d_i + q(t)}\left(\frac{\alpha_i d_i}{q(t)} - (w_i(t) - \alpha_i)\right), \\ &= -\frac{\gamma \alpha_i d_i q(t)}{d_i + q(t)}\left(\frac{w_i(t) - \alpha_i}{\alpha_i d_i} - \frac{1}{q(t)}\right) = -\gamma \alpha_i q(t) \mu_i(t) \eta_i(t). \end{aligned}$$

Choose δ_1 large enough such that $\delta_1 N \gamma \alpha_{\min} q_{\min} \mu_{\min} / d_{\max} > \kappa$ where $\alpha_{\min} := \min_i \alpha_i$.

We now prove $\eta_{\max}(t) > -\delta_1(1-\gamma)^t$ for all $t \geq K_2$ by contradiction. Suppose that there is a time $t \geq K_2$ such that $\eta_{\max}(t) \leq -\delta_1(1-\gamma)^t$. Then all the $\eta_i(t)$ are negative, which implies

$$\begin{aligned} Y(t+1) - Y(t) &= \sum_i (w_i(t+1) - w_i(t))/d_i = \sum_i -\gamma \alpha_i q(t) \mu_i(t) \eta_i(t)/d_i, \\ &\geq N(-\eta_{\max}) \gamma \alpha_{\min} q_{\min} \mu_{\min} / d_{\max}, \\ &\geq \delta_1 N (1-\gamma)^t \gamma \alpha_{\min} q_{\min} \mu_{\min} / d_{\max} > \kappa (1-\gamma)^t. \end{aligned}$$

This contradicts equation (4.27), which proves the claim. The proof for $\eta_{\min}(t)$ is similar.

□

Define $L(t)$ as:

$$L(t) := \eta_{\max}(t) - \eta_{\min}(t). \tag{4.28}$$

The following lemma implies that the difference between different $\eta_i(t)$ goes to zero exponentially fast.

Lemma 4.8. *There are two positive numbers δ_3 and δ_4, such that for $t \geq K_2$ we have*

1. $L(t) \geq 0$.
2. $L(t+1) \leq (1 - \gamma + \gamma \mu_{\max}) L(t) + \delta_3 (1-\gamma)^t$.
3. $L(t) \leq \delta_4 (1 - \gamma + \gamma \mu_{\max})^t$.

Proof: See Appendix 4.6.5. □

Lemma 4.9. *Both $\eta_{\max}(t)$ and $\eta_{\min}(t)$ exponentially converge to zero.*

Proof: When $t \geq K_2$, combining Lemma 4.7 and Lemma 4.8 yields an upper bound for $\eta_{\max}(t)$,

$$\eta_{\max}(t) = L(t) + \eta_{min}(t) \leq \delta_4(1 - \gamma + \gamma\mu_{\max})^t + \delta_2(1 - \gamma)^t.$$

The lower bound of $\eta_{\max}$ is $-\delta_1(1-\gamma)^t \leq \eta_{max}(t)$. Since both the upper and lower bounds of $\eta_{\max}(t)$ converge to zero exponentially fast, it exponentially goes to zero. The proof for $\eta_{\min}(t)$ is similar. □

Proof of Theorem 4.5: The system is at equilibrium if and only if $w_i(t) = w_i(t+1)$ for all i. This is equivalent to $\eta_i(t) = 0$ for all i because of the equation

$$w_i(t+1) - w_i(t) = -\gamma\alpha_i q(t)\mu_i(t)\eta_i(t).$$

Since both $\eta_{\max}(t)$ and $\eta_{\min}(t)$ converge to zero exponentially from any initial value, the system converges to the equilibrium defined by $\eta_i(t) = 0$ globally. □

4.5 Conclusion

We have introduced the traditional continuous-time model for FAST TCP. We analyze its stability for general networks. We prove that the FAST TCP system is globally stable without feedback delay. When the feedback delays are present, a sufficient condition is provided for local stability. However, there are certain inconsistencies between this model and our experiments, which maybe due to the self-clocking effects.

We present a new discrete-time link model that fully captures the effect of self-clocking. Using this discrete-time model, we have derived a sufficient condition for local asymptotic stability for general networks in the presence of feedback delay. The condition states that the system is stable if the difference among delays of the sources is small. This implies, in particular, that a network with homogeneous sources is always stable, consistent with our experimental experience so far. We also prove that FAST TCP is globally stable on a single link in the absence of feedback delay.

This work can be extended in several ways. First, the condition for local asymptotic stability derived appears more restrictive than our experiments suggest. Moreover, we have also found scenarios where predictions of the discrete-time model disagree with experiment. These discrepancies should be clarified. Second, it will be interesting to extend the global stability analysis to general networks with feedback delays. Finally, the new model and the analysis techniques here can be applied to analyze other congestion control algorithms.

4.6 Appendix

4.6.1 Proof of Lemma 4.1

The FAST TCP model (4.1, 4.3, 4.5, 4.2, and 4.6) can be linearized into

$$\begin{aligned} \delta q_i(t) &= \sum_l R_{li} \delta p_l(t - \tau^b_{li}), &\qquad \delta x_i(t) &= \frac{\delta w_i(t)}{d_i + q_i} - \frac{w_i \delta q_i(t)}{(d_i + q_i)^2}, \\ \delta \dot{w}_i(t) &= \gamma \left(\frac{-q_i \delta w_i(t)}{d_i + q_i} - \frac{d_i w_i \delta q_i(t)}{(d_i + q_i)^2} \right), &\qquad \delta y_l(t) &= \sum_i R_{li} \delta x_i(t - \tau^f_{li}), \\ & & \delta \dot{p}_l(t) = \delta y_l(t)/c_l, \end{aligned}$$

where w_i and q_i are the equilibrium values. Since$\tau_i = \tau^f_{li} + \tau^b_{li}$ for all link l on the path of source i, the following equation holds

$$R^T_b(s) = \text{diag}(e^{-\tau_i s}) R^T_f(-s). \tag{4.29}$$

The Laplace transform of the linearized system in matrix form is

$$\begin{cases} Q(s) &= R_b(s)^T P(s), \\ X(s) &= D_1 W(s) - BQ(s,) \\ sW(s) &= \gamma\left((DD_1 - I)W(s) - DBQ(s)\right), \\ Y(s) &= R_f(s)X(s), \\ sP(s) &= D_3 Y(s), \end{cases}$$

where the diagonal matrices are

$$D := \text{diag}\left(d_i\right), \qquad D_1 := \text{diag}\left(\frac{1}{d_i + q_i}\right),$$
$$B := \text{diag}\left(\frac{w_i}{(d_i + q_i)^2}\right), \qquad D_3 := \text{diag}\left(\frac{1}{c_l}\right).$$

The open-loop transfer function from $P(s)$ to $P(s)$ can be derived based on the above equations as

$$\frac{1}{s} D_3 R_f(s) \left(\gamma D D_1 (sI - \gamma(DD_1 - I))^{-1} + 1\right) D_2 R_b^T(s).$$

By using the fact that $T_i = d_i + q_i$, $x_i = w_i/T_i$ and (4.29), we can simplify the open loop transfer function $L(s)$ into (4.16). □

4.6.2 Proof of Theorem 4.3

It is sufficient to show that the eigenvalues of the open-loop transfer function do not encircle -1 in the complex plain for $s = j\omega$, $\omega \geq 0$ when the condition is satisfied. Since both X and $\Lambda(s)$ are diagonal matrices, by using the similar technique in [24], it is not difficult to check that when $s = j\omega$, the set of eigenvalues of $L(s)$ is same as that of $L(j\omega) = \Lambda(s)\hat{R}^T(-j\omega)\hat{R}(j\omega)$ except for some zero eigenvalues where $\hat{R}(j\omega)$ is defined as

$$\hat{R}(j\omega) := \text{diag}(\frac{1}{\sqrt{c_l}}) R_f(j\omega) \text{diag}(\sqrt{x_i}).$$

Following the argument of [152], we study the convex hull as a function of $j\omega$ formed by N Nyquist trajectories. More specifically, the spectrum of $L(j\omega)$ satisfies

$$\sigma(L(j\omega)) = \sigma\left(\Lambda(s)\hat{R}^T(-j\omega)\hat{R}(j\omega)\right) \subseteq \rho\left(\hat{R}^T(-j\omega)\hat{R}(j\omega)\right) \cdot \text{co}\left(0 \cup \{\Lambda_i(j\omega)\}\right),$$

where $i = 1 \dots N$, co$(\cdot)$ denotes the convex hull, and

$$\Lambda_i(j\omega) := \frac{e^{-j\omega T_i}}{j\omega T_i}\frac{j\omega T_i + \gamma T_i}{j\omega T_i + \gamma q_i},$$

where the τ_i is replaced with T_i since $\tau_i = T_i$ at equilibrium. Similar to [24], the spectral radius of $\hat{R}^T(-j\omega)\hat{R}(j\omega)$ is less than M, which is the maximal number of links in the path of any source, $M = \max_i \sum_l R_{li}$. It implies

$$\sigma(L(j\omega)) \subseteq M \cdot \text{co}\left(0 \cup \{\Lambda_i(j\omega),\ \ i = 1 \dots N\}\right).$$

Therefore a sufficient condition for local stability is that $M\Lambda_i(j\omega)$ does not encircle -1 for any i.

It is a standard control theory result that the largest phase lag of $(j\omega T_i + \gamma T_i)/(j\omega T_i + \gamma q_i)$ is produced when $\omega T_i = \sqrt{\gamma T_i \cdot \gamma q_i}$, which is

$$\angle\frac{j\sqrt{\gamma T_i \cdot \gamma q_i} + \gamma T_i}{j\sqrt{\gamma T_i \cdot \gamma q_i} + \gamma q_i} \quad = \quad -\tan^{-1}\frac{1 - q_i/T_i}{2\sqrt{q_i/T_i}}.$$

The above equation yields

$$\angle\Lambda_i(j\omega) \quad \geq \quad -\omega T_i - \frac{\pi}{2} - \tan^{-1}\frac{1 - q_i/T_i}{2\sqrt{q_i/T_i}}.$$

Suppose that at frequency ω_i the phase lag of $\Lambda_i(j\omega)$ is $-\pi$. Hence,

$$-\pi = \Lambda_i(j\omega_i) \geq -\omega_i T_i - \frac{\pi}{2} - \tan^{-1}\frac{1 - q_i/T_i}{2\sqrt{q_i/T_i}}.$$

Based on (4.18) we have

$$\omega_i T_i \geq \phi \text{ for } i = 1 \ldots N.$$

It is easy to check that the magnitude of $\Lambda_i(j\omega)$ is a decreasing function of ω. Therefore,

$$M|\Lambda_i(j\omega_i)| \quad \leq M \quad |\Lambda_i(j\frac{\phi}{T_i})| = \frac{M}{\phi}\sqrt{\frac{\phi^2 + \gamma^2 T_i^2}{\phi^2 + \gamma^2 q_i^2}} \leq \frac{M}{\phi}\sqrt{\frac{\phi^2 + \gamma^2 T_{max}^2}{\phi^2 + \gamma^2 q_{min}^2}} < 1,$$

and $M\Lambda(j\omega_i)$ can not encircle -1. Based on the above argument, the system is locally stable if (4.17) is satisfied. □

4.6.3 Proof of Lemma 4.3

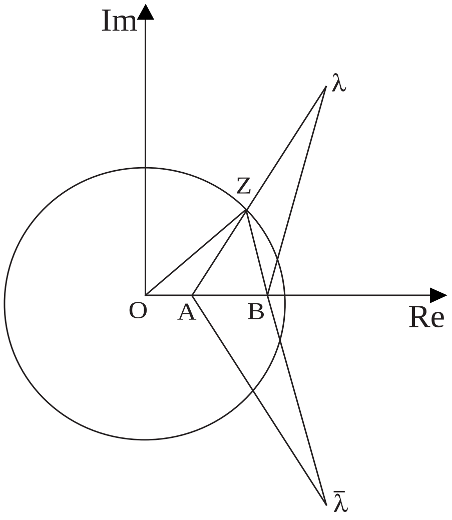


Figure 4.5: Illustration of Lemma 4.3.

Proof: There is a complex plane in Figure 4.5. Let the points A, B, and λ represent the value of $\mu_{\min}$, $\mu_{\max}$, and λ, respectively. Z is the intersection of segment $A\lambda$ and the unit circle, and $\bar{\lambda}$ stands for the complex conjugate of λ.

Let $\phi_i \in [0, 2\pi)$ be the phase of $1/(\mu_i - \lambda)$. Clearly, $\phi_i \in [0, \pi)$ if $\text{Im}(\lambda) \leq 0$, and $\phi_i \in (\pi, 2\pi)$ otherwise. Denote $\phi_{\max} := \max_i \phi_i$ and $\phi_{\min} := \min_i \phi_i$, then $0 \leq \phi_{\max} - \phi_{\min} \leq \pi$. Since every μ_i is in the range $[\mu_{\min}, \mu_{\max}]$, it is easy to check that every

ϕ_i is in the range formed by the phases of $1/(\mu_{\min} - \lambda)$ and $1/(\mu_{\max} - \lambda)$. This implies

$$\phi_{\max} - \phi_{\min} \leq |\angle \frac{1}{\mu_{\min} - \lambda} - \angle \frac{1}{\mu_{\max} - \lambda}| = \angle A\bar{\lambda}B = \angle A\lambda B < \angle OZB < \pi/2.$$

Let $\epsilon > 0$ be small enough such that $\phi_{\max} - \phi_{\min} < \pi/2 - \epsilon$. Choosing $\psi = -\phi_{\min} + \epsilon$ gives us

$$\begin{aligned} \angle \frac{e^{j(\psi+\theta_i)}\beta_i}{\mu_i - \lambda} &= \phi_i + \psi + \theta_i, \\ &= \phi_i - \phi_{\min} + \epsilon + \theta_i, \ \text{(greater than 0)} \\ &< \phi_{\max} - \phi_{\min} + \epsilon + \pi/2 < \pi. \end{aligned}$$

The fact that its phase is in $(0, \pi)$ implies that

$$Im\left(\frac{e^{j(\psi+\theta_i)}\beta_i}{\mu_i - \lambda}\right) > 0.$$

□

4.6.4 Proof of Lemma 4.4

Suppose that $A = A_r + jA_i$ where $A_r = A_r^T$ and A_i is positive definite. If A is singular, there exists a nonzero vector v such that $Av = 0$. Suppose that $v = \alpha + j\beta$. Then $Av = 0$ gives

$$A_r\alpha - A_i\beta = 0, \quad (4.30)$$

$$A_r\beta + A_i\alpha = 0. \quad (4.31)$$

Multiplying β^T to equation (4.30) yields

$$\beta^T A_r\alpha = \beta^T A_i\beta \geq 0. \quad (4.32)$$

Multiplying α^T to equation (4.31) gives us

$$\alpha^T A_r \beta = -\alpha^T A_i \alpha \leq 0. \tag{4.33}$$

Since $\beta^T A_r \alpha = \alpha^T A_r^T \beta = \alpha^T A_r \beta$, both (4.32) and (4.33) must hold with equality. This means that both α and β are zero. It contradicts the assumption that v is nonzero. □

4.6.5 Proof of Lemma 4.8

It is obvious that $L(t) \geq 0$ because of its definition in (4.28). We start with the update of $\eta_i(t)$

$$\begin{aligned} \eta_i(t+1) - \eta_i(t) &= \frac{w_i(t+1) - w_i(t)}{\alpha_i d_i} - \frac{1}{q(t+1)} + \frac{1}{q(t)}, \\ &= -\frac{\gamma \alpha_i q(t) \mu_i(t) \eta_i(t)}{\alpha_i d_i} - \frac{1}{q(t+1)} + \frac{1}{q(t)}, \\ &= -\frac{\gamma q(t) \eta_i(t)}{d_i + q(t)} - \frac{1}{q(t+1)} + \frac{1}{q(t)}, \\ &= -\gamma(1 - \mu_i(t))\eta_i(t) - \frac{1}{q(t+1)} + \frac{1}{q(t)}. \end{aligned}$$

For simplicity, we let $a_i(t) := 1 - \gamma + \gamma\mu_i(t)$ and denote $a_{\max} := 1 - \gamma + \gamma\mu_{\max}$, then $a_i(t) \leq a_{\max}$. This definition simplifies the above equation into

$$\eta_i(t+1) = a_i(t)\eta_i(t) - \frac{1}{q(t+1)} + \frac{1}{q(t)}. \tag{4.34}$$

By comparing equation (4.34) for source i and j, we obtain

$$\eta_i(t+1) - \eta_j(t+1) = a_i(t)\eta_i(t) - a_j(t)\eta_j(t). \tag{4.35}$$

Without loss of generality, suppose that at time $t+1$, the largest and smallest values of η are achieved at sources i and j, respectively. This assumption implies

$$L(t+1) = \eta_i(t+1) - \eta_j(t+1).$$

The upper bound of $L(t+1)$ is derived by considering the following three cases separately.

Case 1: $\eta_i(t)$ and $\eta_j(t)$ have different signs. It is easy to see that

$$\begin{aligned} L(t+1) &= a_i(t)\eta_i(t) - a_j(t)\eta_j(t) \le a_{\max}(\eta_i(t) - \eta_j(t)), \\ &\le a_{\max}(\eta_{\max}(t) - \eta_{\min}(t)) = a_{\max}L(t). \end{aligned}$$

Case 2: Both $\eta_i(t)$ and $\eta_j(t)$ are positive. It yields

$$\begin{aligned} L(t+1) &= a_i(t)\mu_i(t)\eta_i(t) - a_j(t)\eta_j(t) \le a_{\max}\eta_{\max}(t), \\ &= a_{\max}L(t) + a_{\max}\eta_{\min}(t) \le a_{\max}L(t) + a_{\max}\delta_2(1-\gamma)^t, \\ &\le a_{\max}L(t) + \delta_3(1-\gamma)^t, \end{aligned}$$

where the last step is choosing δ_3 larger than $a_{\max}\delta_2$.

Case 3: Both $\eta_i(t)$ and $\eta_j(t)$ are negative. The proof is similar to that for Case 2.

Summarizing all the above cases, we have proved $L(t+1) \le a_{\max}L(t) + \delta_3(1-\gamma)^t$ for all $t \ge K_2$. Denote $b := 1-\gamma$, then $1 > a_{\max} > b \ge 0$. For any $t \ge K_2$, an upper bound of $L(t)$ is

$$\begin{aligned} L(t) &\le a_{\max}L(t-1) + \delta_3 b^{t-1} \le a_{\max}^{t-K_2}L(K_2) + \delta_3(b^{t-1} + b^{t-2}a_{\max} + \cdots + b^{K_2}a_{\max}^{t-K_2-1}), \\ &= \left(a_{\max}^{-K_2}L(K_2) - \delta_3\frac{b^{K_2}a_{\max}^{-K_2}}{b-a_{\max}}\right)a_{\max}^t + \frac{\delta_3}{b-a_{\max}}b^t. \end{aligned}$$

Note that the coefficient of b^t is negative. By choosing δ_4 as the coefficient of $a_{\max}^t$, we get

$$L(t) \le \delta_4 a_{\max}^t = \delta_4(1-\gamma+\gamma\mu_{\max})^t.$$

□

Chapter 5

Cross-Layer Optimization in TCP/IP Networks

5.1 Introduction

Recent studies have shown that any TCP congestion control algorithm can be interpreted as carrying out a distributed primal-dual algorithm over the Internet to maximize aggregate utility, and a user's utility function is defined by its TCP algorithm, see, e.g., [80, 97, 116, 107, 101, 88, 96] for unicast, [75, 30] for multi-cast, and [98, 79, 138] for recent surveys and further references. All of these works assume that routing is given and fixed at the timescale of interest, and TCP, together with active queue management (AQM), attempts to maximize aggregate utility over source rates. In this chapter, we study the cross-layer utility maximization at the timescale of route changes.

We focus on the situation where a single minimum-cost route (shortest path) is selected for each source-destination pair. This models IP routing in the current Internet within an Autonomous System using common routing protocols such as OSPF [119][1] or RIP [56]. Routing is typically updated at a much slower timescale than TCP–AQM. We model this by assuming that TCP and AQM converge instantly to equilibrium after each route update to produce source rates and "congestion prices" for that update period. These congestion prices may represent delays or loss probabilities across network links. They determine the next routing update in the case of dynamic routing, similar to the system analyzed in

[1] Even though OSPF implements a shortest-path algorithm, it allows multiple equal-cost paths to be utilized. Our model ignores this feature.

[48]. Thus TCP–AQM/IP forms a feedback system where routing interacts with congestion control in an iterative process. We are interested in the equilibrium and stability properties of this iterative process. To simplify notation, we will henceforth use TCP–AQM/IP and TCP/IP interchangeably.

Here are our main results. In the case of pure dynamic routing, i.e., when link costs are the congestion prices generated by TCP–AQM, it turns out that we can interpret TCP/IP as a distributed primal-dual algorithm to maximize aggregate utility over *both* source rates (by TCP–AQM) and routes (by IP) if TCP/IP converges. We consider the problem, and its Lagrangian dual, of maximizing utility over source rates and over routing that use only a *single* path for each source-destination pair. Unlike the TCP-AQM problem or the multi-path routing problem that are convex optimizations with no duality gap, the single path TCP/IP problem is non-convex and generally has a duality gap. Equilibrium of the TCP/IP system exists if and only if this problem has no duality gap. In this case, TCP/IP equilibrium solves both the primal and the dual problem. Moreover, it incurs no penalty for not splitting traffic across multiple paths: optimal single-path routing achieves the same aggregate utility as optimal multi-path routing. Multi-path routing can achieve a strictly higher utility only when there is a duality gap between the single-path primal and dual problems, but in this case, the TCP/IP iteration does not even have an equilibrium, let alone solving the utility maximization problem.

Even when the single-path problem has no duality gap and TCP/IP has an equilibrium, the equilibrium is generally unstable under pure dynamic routing. It can be stabilized by adding a sufficiently large static component to the definition of link cost. The existence and characterization of TCP/IP equilibrium when the link costs are not pure congestion prices, however, are open problems. To proceed, we specialize to a ring network with a common destination and demonstrate an inevitable tradeoff between utility maximization and routing stability (Section 5.5). Specifically, we show that the TCP/IP system over the special ring network is indeed unstable when link costs are pure prices. It can be stabilized by adding a static component to the link cost, but at the expense of a reduced utility in equilibrium. The loss in utility increases with the weight on the static component. Hence, while stability requires a small weight on prices, utility maximization favors a large weight.

We present numerical results to validate these qualitative conclusions in a general network topology. These results also suggest that routing instability can reduce aggregate utility to less than that achievable by (the necessarily stable) pure static routing.

Indeed we show that if the link capacities are optimally provisioned, then *pure static* routing is enough to maximize utility even for general networks (Section 5.6). Moreover, it is optimal within the class of multi-path routing: again, there is no penalty at optimality in not splitting traffic across multiple paths.

Finally, we discuss some implications and limitations of these results (Section 5.7).

5.2 Related work

Our goal is to understand equilibrium and stability issues in cross-layer optimization of TCP/IP networks. Another approach to joint routing and congestion control is to allow multi-path routing, i.e., a source can transmit its data along multiple paths to its destination. In this formulation, a source's decision is decomposed into two: how much traffic to send (congestion control) and how to distribute it over the available paths (multi-path routing or load balancing) in order to maximize aggregate utility. This has been analyzed in, e.g., [46, 80, 74]. The general intuition is that, for each source-destination pair, only paths with the minimum, and hence equal, "congestion prices" will be used, and this minimum price determines the total source rate as in the single-path case. In contrast to TCP/IP networks, this formulation assumes that both decisions operate on the same timescale. However, it provides an upper bound to the problem TCP/IP attempts to solve (see Section 5.4).

The multi-path problem is equivalent to the multicommodity flow problem which has been extensively studied; see, e.g., [1, 13]. The standard formulation is to maximize aggregate throughput, corresponding to a common and linear utility function. It is then a linear program and therefore can be solved in polynomial time, though there are combinatorial algorithms for this class of problems that are more efficient. Recently, fast approximation algorithms and their competitive ratios have been developed for network flow, and other, problems, e.g., [130, 48, 7]. Since the multi-path problem upper bounds the TCP/IP problem, the work on network flow problems provides insight to the performance limit of

TCP/IP. There are however differences. First, our single-path routing problem is NP-hard (see Section 5.4) and generally has a duality gap, whereas the network flow problem is generally a linear program that is in P and has no duality gap. Second, the utility functions that correspond to common TCP algorithms are strictly concave whereas they are typically linear, in fact, identity, functions in network flow problems. Third, the algorithms developed for network flow problems are typically centralized and therefore cannot model TCP/IP iterations or be carried out in a large network where they must be decentralized.

Instability of single-path routing is not surprising as it is well known that stability generally requires that the relative weight on the dynamic (traffic-sensitive) component of the link cost be small. Indeed, our conclusions are similar to those reached in [12, 103] that study the same ring network for routing stability using different link costs. Here, since the dynamic component is the dual-optimal price for the utility maximization problem computed by TCP–AQM, this implies a tradeoff between routing stability and utility maximization.

5.3 Model

In general, we use small letters to denote vectors, e.g., x with x_i as its ith component; capital letters to denote matrices, e.g., H, W, R, or constants; e.g., L, N, K^i; and script letters to denote sets of vectors or matrices, e.g., $\mathcal{W}_s$, $\mathcal{W}_m$, $\mathcal{R}_s$, $\mathcal{R}_m$. Superscript is used to denote vectors, matrices, or constants pertaining to source i, e.g., y^i, w^i, H^i, K^i.

A network is modelled as a set of L uni-directional links with finite capacities $c = (c_l, l = 1, \ldots, L)$, shared by a set of N source-destination pairs, indexed by i (we will also refer to the pair simply as "source i"). There are K^i acyclic paths for source i represented by a $L \times K^i$ 0-1 matrix H^i where

$$H^i_{lj} = \begin{cases} 1, & \text{if path } j \text{of source } i \text{ uses link } l \\ 0, & \text{otherwise.} \end{cases}$$

Let $\mathcal{H}^i$ be the set of all columns of H^i that represents all the available paths to i under single-path routing. Define the $L \times K$ matrix H as

$$H \;\;=\;\; [H^1 \;\ldots\; H^N],$$

where $K := \sum_i K^i$, and H defines the topology of the network.

Let w^i be a $K^i \times 1$ vector where the jth entry represents the fraction of source i on its jth path such that

$$w^i_j \geq 0 \;\; \forall j, \quad \text{and} \quad \mathbf{1}^T w^i = 1,$$

where **1** is a vector of an appropriate dimension with the value 1 in every entry. We require $w^i_j \in \{0, 1\}$ for single-path routing and allow $w^i_j \in [0, 1]$ for multi-path routing. Collect the vectors w^i, $i = 1, \ldots, N$, into a $K \times N$ block-diagonal matrix W. Let $\mathcal{W}_s$ be the set of all such matrices corresponding to single-path routing defined as

$$\{W | W = \text{diag}(w^1, \ldots, w^N) \in \{0, 1\}^{K \times N}, \mathbf{1}^T w^i = 1 \;\}.$$

Define the corresponding set $\mathcal{W}_m$ for multi-path routing as

$$\{W | \; W = \text{diag}(w^1, \ldots, w^N) \in [0, 1]^{K \times N}, \; \mathbf{1}^T w^i = 1 \;\}. \tag{5.1}$$

As mentioned above, H defines the set of acyclic paths available to each source and represents the network topology. W defines how the sources load balance across these paths. Their product defines a $L \times N$ routing matrix $R = HW$ that specifies the fraction of i's flow at each link l. The set of all single-path routing matrices is

$$\mathcal{R}_s \;\;:=\;\; \{ \; R \mid R = HW, W \in \mathcal{W}_s \; \}, \tag{5.2}$$

and the set of all multi-path routing matrices is

$$\mathcal{R}_m \;\;:=\;\; \{ \; R \mid R = HW, W \in \mathcal{W}_m \; \}. \tag{5.3}$$

The difference between single-path routing and multi-path routing is the integer constraint on W and R. A single-path routing matrix in $\mathcal{R}_s$ is an 0-1 matrix

$$R_{li} \;=\; \begin{cases} 1, & \text{if link } \; l \text{ is in a path of source } i \\ 0, & \text{otherwise.} \end{cases}$$

A multi-path routing matrix in $\mathcal{R}_m$ is one whose entries are in the range $[0, 1]$

$$R_{li} \begin{cases} > \; 0, & \text{if link } \; l \text{ is in a path of source } i \\ = \; 0, & \text{otherwise.} \end{cases}$$

The path of source i is denoted by $r^i = [R_{1i} \;\; \ldots \;\; R_{Li}]^T$, the ith column of the routing matrix R.

We consider the situation where TCP–AQM operates at a faster timescale than routing updates. We assume a *single* path is selected for each source-destination pair that minimizes the sum of the link costs in the path, for some appropriate definition of link cost. In particular, traffic is not split across multiple paths from the source to the destination even if it is available. This models, for example, IP routing within an Autonomous System. We focus on the timescale of the route changes and assume TCP–AQM is stable and converges instantly to equilibrium after a route change. As in [96], we will interpret the equilibria of various TCP and AQM algorithms as solutions of a utility maximization problem defined in [80]. Different TCP algorithms solve the same prototypical problem (5.4) with different utility functions [96, 101].

Specifically, suppose each source i has a utility function $U_i(x_i)$ as a function of its total transmission rate x_i. We assume U_i is strictly concave increasing (which is the case for common TCP algorithms [96]). Let $R(t) \in \mathcal{R}_s$ be the (single-path) routing in period t. Let the equilibrium rates $x(t) = x(R(t))$ and prices $p(t) = p(R(t))$ generated by TCP–AQM in period t, respectively, be the optimal solutions of the constrained maximization problem

$$\max_{x \geq 0} \; \sum_i U_i(x_i) \qquad \text{s. t. } \; R(t)x \; \leq \; c, \tag{5.4}$$

and its Lagrangian dual

$$\min_{p\geq 0} \sum_i \max_{x_i\geq 0} \left(U_i(x_i) - x_i \sum_l R_{li}(t) p_l \right) \ + \ \sum_l c_l p_l. \tag{5.5}$$

The prices, $p_l(t)$, $l = 1, \ldots, L$, are measures of congestion, such as queueing delays or loss probabilities [96, 101]. We assume the link costs in period t are

$$d_l(t) \ \ = \ \ ap_l(t) + b\tau_l, \tag{5.6}$$

where $a \geq 0$, $b \geq 0$, and $\tau_l > 0$ are constants. Based on these costs, each source computes its new route $r^i(t+1) \in \mathcal{H}^i$ individually that minimizes the total cost on its path

$$r^i(t+1) \ \ = \ \ \arg\min_{r^i\in\mathcal{H}^i} \sum_l d_l(t) r^i_l. \tag{5.7}$$

In (5.6), τ_l are traffic insensitive components of the link cost $d_l(t)$, e.g., $\tau_l = 1/c_l$. If τ_l represent the fixed propagation delays across links l and $p_l(t)$ the queueing delays at link l, then $d_l(t)$ represent total delays across link l. The protocol parameters a and b determine the responsiveness of routing to network traffic: $a = 0$ corresponds to static routing, $b = 0$ corresponds to purely dynamic routing, and the larger the ratio of a/b, the more responsive routing is to network traffic. As we will see below, they determine whether an equilibrium exists and whether it is stable, and the achievable utility at equilibrium.

An equivalent way to specify the TCP–AQM/IP system as a dynamical system, at the timescale of route changes, is to replace (5.4)–(5.5) by their optimality conditions. The routing is updated according to

$$r^i(t+1) = \arg\min_{r^i\in\mathcal{H}^i} \sum_l (ap_l(t) + b\tau_l) r^i_l, \ \ \text{for all } i, \tag{5.8}$$

where $p(t)$ and $x(t)$ are given by

$$\begin{aligned}\sum_l R_{li}(t)p_l(t) &= U_i'(x_i(t)) \quad \text{for all } i & (5.9)\\ \sum_i R_{li}(t)x_i(t) &\begin{cases} \leq c_l & \text{if } p_l(t) \geq 0 \\ = c_l & \text{if } p_l(t) > 0 \end{cases} \quad \text{for all } l & (5.10)\\ x(t) \geq 0, &\quad p(t) \geq 0. & (5.11)\end{aligned}$$

This set of equations describe how the routing $R(t)$, rates $x(t)$, and prices $p(t)$ evolve. Note that $x(t)$ and $p(t)$ depend only on $R(t)$ through (5.9)–(5.11), implicitly assuming that TCP–AQM converges instantly to an equilibrium given the new routing $R(t)$.

We say that (R^*, x^*, p^*) is an *equilibrium of TCP/IP* if it is a fixed point of (5.4)–(5.7), or equivalently, (5.8)–(5.11), i.e., starting from routing R^* and associated (x^*, p^*), the above iterations yield (R^*, x^*, p^*) in the subsequent periods.

5.4 Equilibrium of TCP/IP

In this section, we study the condition under which TCP/IP as modelled by (5.4)–(5.7) or (5.8)–(5.11) has an equilibrium and characterize the equilibrium as the optimal solution of an optimization problem. Since (5.8)–(5.11) is a system of mixed integer nonlinear inequalities, characterization of its equilibrium, even the basic question of existence and uniqueness, is in general difficult to determine. The case of pure dynamic routing, $a > 0$ and $b = 0$, is the simplest and most instructive.

We completely characterize the case of pure dynamic routing, $a > 0$ and $b = 0$ in this section. Without loss of generality, we set $a = 1$ in (5.7) and (5.8) when $b = 0$.

Consider the joint optimization problem

$$\max_{R \in \mathcal{R}_s} \max_{x \geq 0} \sum_i U_i(x_i) \quad \text{s. t. } Rx \leq c, \tag{5.12}$$

and its Lagrangian dual

$$\min_{p\geq 0} \sum_i \max_{x_i \geq 0} \left(U_i(x_i) - x_i \min_{r^i \in \mathcal{H}^i} \sum_l R_{li} p_l \right) + \sum_l c_l p_l, \tag{5.13}$$

where r^i is the ith column of R with $r^i_l = R_{li}$. While (5.4)–(5.5) maximize utility over source rates only, problem (5.12) maximizes utility over both rates and routes. While (5.4) is a convex program without duality gap, problem (5.12) is non-convex because the variable R is discrete and generally has a duality gap.[2] The interesting feature of the dual problem (5.13) is that the maximization over R takes the form of minimum-cost routing with prices p generated by TCP–AQM as link costs. This suggests that TCP/IP might turn out to be a distributed algorithm that attempts to maximize utility, with proper choice of link costs. This is indeed true when equilibrium of TCP/IP exists.

Theorem 5.1. *Under pure dynamic routing, that is, $a = 1$ and $b = 0$.*

1. *An equilibrium (R^*, x^*, p^*) of TCP/IP exists if and only if there is no duality gap between (5.12) and (5.13).*

2. *In this case, the equilibrium (R^*, x^*, p^*) is a solution of (5.12) and (5.13).*

Hence, one can regard the layering of TCP and IP as a decomposition of the utility maximization problem over source rates and routes into a distributed and decentralized algorithm, carried out on two different timescales, in the sense that an equilibrium of the TCP/IP iteration (5.8)–(5.11), if it exists, solves (5.12) and (5.13). An equilibrium may not exist. Even if it does, it may not be stable–an issue we address in Section 5.5.

Example 1: Duality gap

A simple example, where there is a duality gap and equilibrium of TCP/IP does not exist, consists of a single source-destination pair connected by two parallel links each of capacity 1, as shown in Figure 5.2 (take $N = 1$). Clearly, under pure dynamic single-path routing, equilibrium of TCP/IP does not exist, because the TCP/IP iteration (5.8)–(5.11) will choose

[2] The nonlinear constraint $Rx \leq c$ can be converted into a linear constraint–see proof of Theorem 5.2–so integer constraint on R is the real source of difficulty.

one of the two routes in each period to carry all traffic. TCP–AQM will generate positive price for the chosen route and zero price for the other route, so that in the next period, the other route will be selected, and the cycle repeats. The proof that there is a duality gap between the primal problem (5.12) and the dual problem (5.13) is given in Appendix 5.8.1 (take $N = 1$). Intuitively, either path is optimal (both for primal and for dual problem). For the primal problem the optimal rate is $x^* = 1$, constrained by link capacity, whereas for the dual problem, the optimal rate is $x^* = 2$, primal infeasible. Hence the primal optimal value is $U(1)$, strictly less than the dual optimal value of $U(2)$. □

The duality gap is a measure of "cost of not splitting". To elaborate, define the Lagrangian [14, 104]

$$L(R, x, p) \;=\; \sum_i \left(U_i(x_i) - x_i \sum_l R_{li} p_l \right) \;+\; \sum_l c_l p_l .$$

The primal (5.12) and dual (5.13) can then be expressed respectively as

$$\begin{aligned} V_{sp} &= \max_{R \in \mathcal{R}_s, x \geq 0} \; \min_{p \geq 0} \; L(R, x, p) \\ V_{sd} &= \min_{p \geq 0} \; \max_{R \in \mathcal{R}_s, x \geq 0} \; L(R, x, p). \end{aligned}$$

If we allow sources to distribute their traffic among multiple paths available to them, then the corresponding problems for multi-path routing are

$$\begin{aligned} V_{mp} &= \max_{R \in \mathcal{R}_m, x \geq 0} \; \min_{p \geq 0} \; L(R, x, p) \\ V_{md} &= \min_{p \geq 0} \; \max_{R \in \mathcal{R}_m, x \geq 0} \; L(R, x, p). \end{aligned} \tag{5.14}$$

Theorem 5.2. *The relations among these four problems are* $V_{sp} \leq V_{sd} = V_{mp} = V_{md}$.

According to Theorem 5.1, TCP/IP has an equilibrium exactly when there is no duality gap in the single-path utility maximization, i.e., when $V_{sp} = V_{sd}$. Theorem 5.2 then says that in this case, there is no penalty in not splitting the traffic, i.e., single-path routing performs as well as multi-path routing, $V_{sp} = V_{mp}$. Multi-path routing achieves a strictly higher utility V_{mp} precisely when TCP/IP has no equilibrium, in which case the TCP/IP iteration (5.8)–(5.11) cannot converge, let alone solving the single-path utility maximization problem (5.12) or (5.13). In this case the problem (5.12) and its dual (5.13) do not characterize TCP/IP, but their gap measures the loss in utility in restricting routing to single-path and is of independent interest.

Even though minimum-cost routing is polynomial, it is shown in [153] that single-path utility maximization is NP-hard. This is not surprising since, e.g., a related problem on load balancing on a ring has been proved to be NP-hard in [29].

Theorem 5.3. *The primal problem (5.12) is NP-hard.*

Theorem 5.3 shows that the general problem (5.12) is NP-hard by reducing all instances of the integer partition problem to some instances of the primal problem (5.12). Theorem 5.2 however implies that the sub-class of the utility maximization problems with no duality gap are polynomial-time solvable, since they are equivalent to multi-path problems that are concave programs and hence polynomial-time solvable. It is a common phenomenon for sub-classes of NP-hard problems to have polynomial-time algorithms. Informally, the hard problems are those with nonzero duality gap.

The rest of this section is devoted to the proofs for Theorems 5.1–5.3. We will first prove Theorem 5.2. Then we show that an equilibrium of TCP/IP must solve the dual problem (5.13). This together with Theorem 5.2 implies Theorem 5.1. Finally, we present a proof for Theorem 5.3.

Proof of Theorem 5.2. The inequality follows from the weak duality theorem [14]. We now prove $V_{sd} = V_{md}$ and $V_{mp} = V_{md}$. We formulate V_{sd} as

$$\begin{aligned} V_{sd} &= \min_{p \geq 0} \max_{R \in \mathcal{R}_s, x \geq 0} \left(\sum_i U_i(x_i) - p^T R x \right) + p^T c, \\ &= \min_{p \geq 0} \max_{x \geq 0} \left(\sum_i U_i(x_i) - \min_{W \in \mathcal{W}_s} p^T H W x \right) + p^T c, \end{aligned}$$

where $R = HW$ with $W \in \mathcal{W}_s$ from (5.2). Similarly, from (5.3) we have

$$V_{md} \;=\; \min_{p\geq 0}\max_{x\geq 0}\left(\sum_i U_i(x_i) - \min_{W\in\mathcal{W}_m} p^T HWx\right) + p^T c.$$

Define functions $f_s(x,p)$ and $f_m(x,p)$ as

$$f_s(x,p) := \min_{W\in\mathcal{W}_s} p^T HWx, \qquad f_m(x,p) := \min_{W\in\mathcal{W}_m} p^T HWx.$$

In order to show that $V_{sd} = V_{md}$, we only need to show that $f_s(x,p) = f_m(x,p)$. Clearly $f_s(x,p) \geq f_m(x,p)$ since $\mathcal{W}_s \subseteq \mathcal{W}_m$. From (5.1), noting that $W = \text{diag}(w^i)$, we have

$$\begin{array}{rcl} f_m(x,p) & = & \min\limits_W p^T HWx \\ \text{s. t.} & & \mathbf{1}^T w^i = 1, \quad 0 \leq w^i_j \leq 1. \end{array}$$

Since this is a linear programming for given x and p, at least one of the optimal points lies on the boundary, i.e., $w^i_j = 0$ or 1 for all i and j, and hence is in $\mathcal{W}_s \subseteq \mathcal{W}_m$. Such a point solves both $f_s(x,p)$ and $f_m(x,p)$, i.e., $f_s(x,p) = f_m(x,p)$.

To show $V_{md} = V_{mp}$, transform V_{mp} into a convex optimization with linear constraints, which hence has no duality gap; see, e.g., [14]. Now, V_{mp} is equivalent to the problem

$$\max_{R\in\mathcal{R}_m, x\geq 0} \sum_i U_i(x_i) \quad \text{s.t. } Rx \leq c. \tag{5.15}$$

Note that this is not a convex program since the feasible set specified by $Rx \leq c$ is generally not convex. Define the $K_i \times 1$ vectors y^i in terms of the scalar x_i and the $K_i \times 1$ vectors w^i as the new variables

$$y^i \;=\; x_i w^i. \tag{5.16}$$

The mapping from (x_i, w^i) to y^i is one-to-one: the inverse of (5.16) is $x_i = \mathbf{1}^T y^i$ and $w^i = y^i/x_i$. Now change the variables in (5.15) and (5.14) from (W, x) to y by substituting

$x_i = \mathbf{1}^T y^i$ and $Rx = HWx = Hy$ into (5.15) and (5.14). We obtain an equivalent problem

$$\max_{y \geq 0} \sum_i U_i(\mathbf{1}^T y^i) \quad \text{s.t.} \;\; Hy \leq c$$

and its Lagrangian dual. This is a convex program with linear constraint and hence has no duality gap. This proves $V_{mp} = V_{md}$. □

Proof of Theorem 5.1. It is easy to show that optimal solutions exist for both the primal problem (5.12) and its dual (5.13), so the issue is whether there is a duality gap. We will prove the theorem in two steps. First, given an equilibrium $(\tilde{R}, \tilde{x}, \tilde{p})$ of TCP/IP, we will show that it solves both the primal (5.12) and the dual (5.13), and hence there is no duality gap. Then, given a solution (R^*, x^*, p^*) of the primal and the dual problems, we will show that it is an equilibrium of TCP/IP.

Step 1: Necessity. Let $(\tilde{R}, \tilde{x}, \tilde{p})$ be an equilibrium of TCP/IP, i.e., a fixed point of (5.4)–(5.7) with $a = 1$, $b = 0$. Then

$$\tilde{p}^T \tilde{r}^i \;=\; \min_{r^i \in \mathcal{H}^i} \tilde{p}^T r^i \qquad \text{for all } i, \tag{5.17}$$

$$(\tilde{p}, \tilde{x}) \;=\; \arg\min_{p \geq 0} \max_{x \geq 0} \left(\sum_i U(x_i) - p^T \tilde{R} x \right) + p^T c, \tag{5.18}$$

where r^i are the ith columns of routing matrix $R \in \mathcal{R}_s$.[3] We will show that $(\tilde{R}, \tilde{x}, \tilde{p})$ solves the dual problem (5.13). Then, since the dual problem (5.13) upper bounds the primal problem (5.12) by Theorem 5.2, and $\tilde{R} \in \mathcal{R}_s$ is a single-path routing and hence primal feasible, $(\tilde{R}, \tilde{x}, \tilde{p})$ also solves the primal (5.12).

To show that $(\tilde{R}, \tilde{x}, \tilde{p})$ solves the dual problem, we use the fact that the dual problem has an optimal solution, denoted by (R^*, x^*, p^*) and show that both achieve the same dual

[3] One can exchange the order of min and max in (5.18) since given $\tilde{R}$, there is no duality gap in $\max_{x \geq 0} \sum_i U_i(x_i)$ s. t. $\tilde{R}x \leq c$.

objective value, i.e., $L(\tilde{R}, \tilde{x}, \tilde{p}) = L(R^*, x^*, p^*)$. Now

$$(p^*, x^*, R^*) = \arg\min_{p \geq 0} \max_{x \geq 0} \left(\sum_i U(x_i) - \min_{R \in \mathcal{R}_s} p^T R x + p^T c \right). \tag{5.19}$$

Let

$$\begin{aligned} f(p) &:= \max_{x \geq 0} \left(\sum_i U(x_i) - p^T \tilde{R} x \right) + p^T c, \\ g(p) &:= \max_{x \geq 0} \left(\sum_i U(x_i) - \min_{R \in \mathcal{R}_s} p^T R x \right) + p^T c. \end{aligned}$$

Then (5.18) implies $f(\tilde{p}) = \min_{p \geq 0} f(p)$ and (5.19) implies $g(p^*) = \min_{p \geq 0} g(p)$. Since $\tilde{R} \in \mathcal{R}_s$, we have

$$f(p) \quad \leq \quad g(p) \qquad \text{for all } p \geq 0,$$

and hence

$$f(\tilde{p}) = \min_{p \geq 0} f(p) \quad \leq \quad \min_{p \geq 0} g(p) = g(p^*).$$

On the other hand

$$\begin{aligned} f(\tilde{p}) &= \max_{x \geq 0} \ \sum_i U(x_i) - \tilde{p}^T \tilde{R} x + \tilde{p}^T c \\ &= \max_{x \geq 0} \sum_i U(x_i) - \sum_i x_i \left(\tilde{p}^T \tilde{r}^i \right) + \tilde{p}^T c \\ &= \max_{x \geq 0} \sum_i U(x_i) - \sum_i x_i \left(\min_{r^i \in \mathcal{H}^i} \tilde{p}^T r^i \right) + \tilde{p}^T c \\ &= g(\tilde{p}) \\ &\geq g(p^*), \end{aligned}$$

where the third equality follows from (5.17). Therefore, $f(\tilde{p}) = g(p^*) = g(\tilde{p})$ and $L(\tilde{R}, \tilde{x}, \tilde{p}) = L(R^*, x^*, p^*)$. Moreover, $(\tilde{R}, \tilde{x}, \tilde{p})$ is an optimal solution of the dual problem.

Step 2: Sufficiency. Assume that there is no duality gap and (R^*, x^*, p^*) is an optimal solution for both the primal problem (5.12) and its dual (5.13). We claim that it is also an equilibrium of (5.4)–(5.7) with $a = 1$ and $b = 0$, i.e., we need to show that

$$\begin{aligned} (p^*)^T (r^i)^* &= \min_{r^i \in \mathcal{H}^i} (p^*)^T r^i, && (5.20) \\ (p^*, x^*) &= \arg\min_{p \geq 0} \max_{x \geq 0} L(R^*, x, p) = \arg\max_{x \geq 0} \min_{p \geq 0} L(R^*, x, p), && (5.21) \end{aligned}$$

where $(r^i)^*$ are the ith columns of R^*. The second equality in (5.21) follows from the fact that there is no duality gap for the TCP–AQM problem.

Since (R^*, x^*, p^*) solves the dual problem (5.13), the optimal routing matrix R^* satisfies (5.20) by the saddle point theorem [14]. But (R^*, x^*, p^*) also solves the primal problem (5.12). In particular, (x^*, p^*) solves the utility maximization problem over source rates and its Lagrangian dual, with R^* as the routing matrix, i.e., (x^*, p^*) satisfies (5.21). □

Proof of Theorem 5.3. We describe a polynomial time procedure that reduces an instance of integer partition problem [47, pp. 47] to a special case of the primal problem. Given a set of integers $c_1, \ldots, c_N$, the integer partition problem is to find a subset $A \subset \{1, \ldots, N\}$ such that

$$\sum_{i \in A} c_i = \sum_{i \notin A} c_i.$$

Given an instance of the integer partition problem, consider the network in Figure 5.1, with N sources at the root, two relay nodes, and N receivers, one at each of the N leaves. The two links from the root to the relay nodes have a capacity of $\sum_i c_i/2$ each, and the two links from each relay node to receiver i have a capacity of c_i. All receivers have the same utility function that is increasing. The routing decision for each source is to decide which relay node to traverse. Clearly, maximum utility of $\sum_i U_i(c_i)$ is attained when each receiver i receives at rate c_i, from exactly one of the relay nodes, and the links from the root to the two relay nodes are both saturated. Such a routing exists if and only if there is a solution to the integer partition problem. □

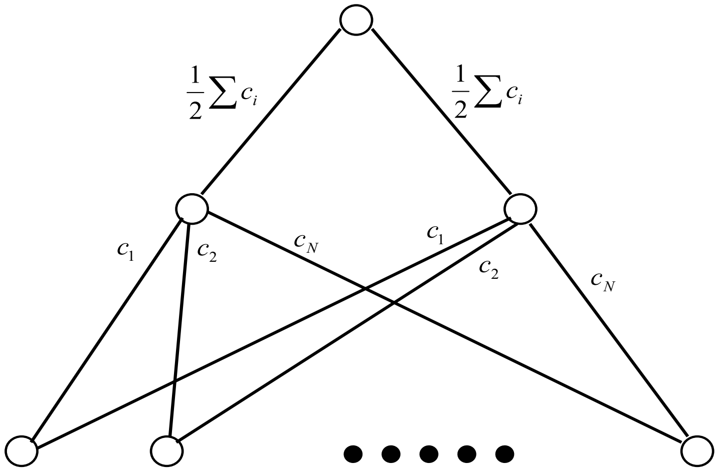


Figure 5.1: Network to which integer partition problem can be reduced.

Comment: The case of $b > 0$ for general network is completely open. If $a = 0$ and $b > 0$, routing $R(t) = R$, for all t, is the static minimum-cost routing with τ_l as the link costs. An equilibrium $(R, x(R), p(R))$ always exists in this case. Even though R minimizes routing cost and $(x(R), p(R))$ solves (5.4)–(5.5), it is not known if $(R, x(R), p(R))$ *jointly* solves any optimization problem.

For the case of $a > 0$ and $b > 0$, even the existence of equilibrium is unknown for general networks. In the following section, we will study the dynamics of TCP/IP under the assumption that such an equilibrium of TCP/IP exists.

5.5 Dynamics of TCP/IP

Theorem 5.1 suggests using pure congestion prices $p(t)$ generated by TCP–AQM as link costs. In this case, an equilibrium of TCP/IP, when it exists, maximizes aggregate utility over both rates and routes. We show in this section however that the equilibrium may not be unstable. Routing can be stabilized by including a strictly positive traffic-insensitive (static) component in link costs ($b > 0$), but at a reduced achievable utility. There thus seems to be an inevitable tradeoff between achievable utility and routing stability.

To make this precise, we start with analysis of a special ring network with a common destination. As remarked in the last section, for a general network, we do not even know if an equilibrium exists when $b > 0$, let alone characterizing it. For the ring network, however, not only does equilibrium always exists, but we can also study rigorously its stability and achievable utility, as well as their tradeoff under minimum-cost routing. We also illustrate through a numerical example that the qualitative conclusions derived from this network seem to generalize to a general network.

5.5.1 Simple ring network

Consider a ring network with $N+1$ nodes, indexed by $i = 0, 1, \ldots, N$. Nodes $i \geq 1$ are sources and their common destination is node 0; see Figure 5.2. For notational convenience we will also refer to node 0 as node $N+1$. Each pair of nodes is connected by two links, one in each direction. We will refer to the two uni-directional links between node $i-1$ and i as link i; the direction should be clear from the context. The fixed delay on link i is denoted as $\tau_i > 0$, $i = 1, \ldots, N+1$, in each direction. We construct the cost on link i in period t as $d_i(t) = ap_i(t) + b\tau_i$, where $p_i(t)$ is the price on link i. At time t, source i routes all of its traffic in the direction, counterclockwise or clockwise, with the smaller cost. The ring network is particularly simple because the routing of the whole network can be represented by a single number r. Note that under minimum-cost routing, if node i sends in the counterclockwise direction, so must node $i-1$, and if node i sends in the clockwise direction, so must node $i+1$. Hence, we can represent routing on the network

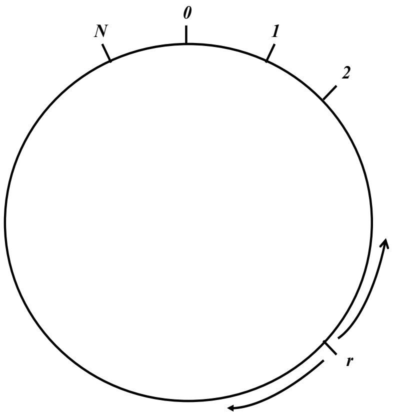


Figure 5.2: A ring network.

by $r \in \{0, \ldots, N\}$ with the interpretation that nodes $1, \ldots, r$ send in the counterclockwise direction and nodes $r+1, \ldots, N$ send in the clockwise direction.

For this special case, we now show that the duality gap is trivial, that minimum-cost routing based just on prices ($b = 0$) indeed solves the primal and dual problems as Theorem 5.1 guarantees, but that the equilibrium is unstable. Using a continuous model, we then show that routing can be stabilized if the weight b on the fixed delay is nonzero and the weight a on price is small enough. The maximum achievable utility however decreases with smaller a. There is thus an inevitable tradeoff between utility maximization and routing stability.

5.5.2 Utility and stability of pure dynamic routing

Suppose all sources i have the same utility function $U(x_i)$, and all links have the same capacity of $c = 1$ unit. We assume that U is *strictly* concave increasing and differentiable. Then at any time, only link 1, in the counterclockwise direction, and link $N+1$, in the clockwise direction, can be saturated and have strictly positive price. The utility maximization problem (5.12) reduces to the following simple form

$$\max_{r \in \{0,\ldots,N\}} \ \max_{x_i} \quad \sum_i U(x_i) \tag{5.22}$$

$$\text{subject to} \quad \sum_{i=1}^{r} x_i \le 1, \ \text{and} \ \sum_{i=r+1}^{N} x_i \le 1. \tag{5.23}$$

When routing is r, nodes $i = 1, \ldots, r$ see price $p_1(r)$ on their paths while nodes $i = r+1, \ldots, N$ see price $p_{N+1}(r)$ on their paths. Since these rates $x_i(r)$ and prices $p_i(r)$ are primal and dual optimal, they satisfy [97]

$$U'(x_i(r)) \ = \ p_1(r) \qquad \text{for } i = 1, \ldots, r, \tag{5.24}$$

$$U'(x_i(r)) \ = \ p_{N+1}(r) \qquad \text{for } i = r+1, \ldots, N. \tag{5.25}$$

This implies that $x_1(r) = \cdots = x_r(r)$ and $x_{r+1}(r) = \cdots = x_N(r)$.

It is easy to see that the optimal routing $r^* \neq 0$ or N. Hence both constraints are active at optimality, implying that (from (5.23))

$$x_1(r) = \cdots = x_r(r) = \tfrac{1}{r}, \tag{5.26}$$

$$x_{r+1}(r) = \cdots = x_N(r) = \tfrac{1}{N-r}. \tag{5.27}$$

The problem (5.22)–(5.23) thus becomes

$$\max_{r\in\{1,\ldots,N-1\}} r\,U\left(\frac{1}{r}\right) \;+\; (N-r)\,U\left(\frac{1}{N-r}\right).$$

Dividing the objective function by N and using the strict concavity of U, we have

$$\frac{r}{N}\,U\left(\frac{1}{r}\right) \;+\; \frac{N-r}{N}\,U\left(\frac{1}{N-r}\right) \;\le\; U\left(\frac{2}{N}\right),$$

with equality if and only if $r = N/2$. This implies that the optimal routing is

$$r^* \;:=\; \lfloor N/2 \rfloor \tag{5.28}$$

and the maximum utility is

$$V^* := \left\lfloor \frac{N}{2} \right\rfloor\, U\left(\frac{1}{\lfloor N/2 \rfloor}\right) \;+\; \left\lceil \frac{N}{2} \right\rceil\, U\left(\frac{1}{\lceil N/2 \rceil}\right), \tag{5.29}$$

where $\lfloor y \rfloor$ is the largest integer less or equal to y and $\lceil y \rceil$ is the smallest integer greater or equal to y.

It can be shown that there is no duality gap for the ring network considered here when N is even, by verifying that routing r^* in (5.28), rates $x_i(r^*)$ in (5.27), and prices $p_1(r^*), p_{N+1}(r^*)$ in (5.24)–(5.25) are indeed primal-dual optimal.[4] When N is odd, there is generally a duality gap due to integral constraint on r; see Appendix 5.8.1 for a proof. This duality gap disappears in the convexified problem when routing is allowed to take real value in $[0, N]$, a model we consider in the next subsection. This suggests that TCP together with minimum-cost routing based on prices can potentially maximize utility for

[4] This also follows from Theorem 5.1 and the fact that $r = N/2$ is by symmetry the equilibrium routing when N is even.

this ring network. We next show, however, that minimum-cost routing based only on prices is unstable.

Given routing r, we can combine (5.24)–(5.25) and (5.27) to obtain the prices $p_1(r)$ and $p_{N+1}(r)$ on links 1 and $N+1$

$$p_1(r) \;=\; U'\left(\frac{1}{r}\right) \text{ and } p_{N+1}(r) \;=\; U'\left(\frac{1}{N-r}\right). \tag{5.30}$$

The path cost for node i in the counterclockwise direction is

$$D^-(i;r) = \sum_{j=1}^{i} b\tau_j + ap_1(r) = b\sum_{j=1}^{i}\tau_j + aU'\left(\frac{1}{r}\right), \tag{5.31}$$

and the path cost in the clockwise direction is

$$D^+(i;r) \;=\; \sum_{j=i+1}^{N+1} b\tau_j + ap_{N+1}(r) = b\sum_{j=i+1}^{N+1}\tau_j + aU'\left(\frac{1}{N-r}\right). \tag{5.32}$$

In the next period, each node i will choose the counterclockwise or clockwise direction accordingly as $D^-(i;r)$ or $D^+(i;r)$ is smaller. Define $f(r)$ as

$$f(r) \;:=\; \max\,\{i \mid D^-(i;r) \le D^+(i;r)\}. \tag{5.33}$$

Then the resulting routing satisfies the recursive relation

$$r(t+1) = \begin{cases} 0 & \text{if } D^-(1;r(t)) > D^+(1;r(t)) \\ N & \text{if } D^-(N;r(t)) < D^+(N;r(t)) \\ f(r(t)) & \text{otherwise.} \end{cases}$$

Theorem 5.4. *If $b = 0$ and $a > 0$, then, starting from any routing $r(0)$ except the equilibrium $N/2$ when N is even, the subsequent routing oscillates between 0 and N.*

Proof. For any $r(0) \in \{0, \ldots, N\}$, we have

$$\begin{aligned} D^-(1; r(0)) - D^+(1; r(0)) &= D^-(N; r(0)) - D^+(N; r(0)), \\ &= a\left(U'\left(\frac{1}{r(0)}\right) - U'\left(\frac{1}{N - r(0)}\right)\right). \end{aligned}$$

If N is even, then $N/2$ is the unique equilibrium routing that solves $D^-(i; N/2) = D^+(i; N/2)$. Suppose $r(0) \neq N/2$. If $r(0) > N/2$, then $1/r(0) < 2/N < 1/(N - r(0))$. Since U' is strictly decreasing, $U'(1/r(0)) > U'(1/(N - r(0))$ and hence $D^-(1; r(0)) > D^+(1; r(0))$ and $r(1) = 0$. Similarly, if $r(0) < N/2$, then $D^-(N; r(0)) < D^+(N; r(0))$ and $r(1) = N$. Hence r oscillates between 0 and N henceforth. □

Even though purely dynamic routing based on prices $(b = 0)$ maximizes utility, Theorem 5.4 says that it is unstable. We will henceforth, without loss of generality, set $b = 1$ and consider the effect of a on utility maximization and stability.

5.5.3 Maximum utility of minimum-cost routing

As mentioned above, the duality gap for the ring network we consider is of a trivial kind that disappears when integer constraint on routing is relaxed. For the rest of this section, we consider a continuous model where every point on the ring is a source. A point on the ring is labelled by $s \in [0, 1]$, and the common destination is the point 0 (or equivalently 1). The utility maximization problem becomes

$$\max_{r \in [0,1]} \max_{x(\cdot)} \quad \int_0^1 U(x(u))du \tag{5.34}$$

$$\text{subject to} \quad \int_0^r x(u)du \leq 1, \text{ and } \int_r^1 x(u)du \leq 1. \tag{5.35}$$

As in the discrete case, both constraints are active at optimality, and hence the problem reduces to

$$\max_{r \in (0,1)} \quad rU\left(\frac{1}{r}\right) + (1 - r)U\left(\frac{1}{1 - r}\right),$$

which, by concavity, yields the optimal routing r^* and maximum utility V^*

$$r^* \;=\; \frac{1}{2}, \quad \text{and} \quad V^* \;=\; U(2). \tag{5.36}$$

To see that there is no duality gap, note that the problem (5.34)–(5.35) is equivalent to

$$\begin{aligned} \max_{r\in[0,1]} \; \max_{x^-,x^+\geq 0} \quad & rU(x^-) + (1-r)U(x^+), \\ \text{subject to} \quad & rx^- \leq 1, \quad (1-r)x^+ \leq 1. \end{aligned}$$

Define the Lagrangian as

$$L(r, x^-, x^+, p^-, p^+) = rU(x^-) + (1-r)U(x^+) + p^-(1 - rx^-) + p^+(1-(1-r)x^+).$$

It is easy to verify that

$$r^* = \frac{1}{2}, \;\; x^{-*} = x^{+*} = 2, \;\; p^{-*} = p^{+*} = U'(2) \tag{5.37}$$

are primal-dual optimal and there is no duality gap; see Appendix 5.8.2.

We now look at the maximum utility achievable by the equilibrium of minimum-cost routing. Let the delay from s to the destination in the counterclockwise direction be

$$T(s) \;\;:=\;\; \int_0^s \tau(u)du,$$

and the delay in the clockwise direction be

$$T(1) - T(s) \;\;=\;\; \int_s^1 \tau(u)du,$$

where $\tau(u)$, $u \in [0,1]$, is given. Here, $\tau(u)$ corresponds to link cost in the discrete model. Given routing $r \in [0,1]$, the price in the counterclockwise direction is $U'(1/r)$, and the price in the clockwise direction is $U'(1/(1-r))$. Then the cost of source s in the counterclockwise direction is

$$D^-(s;r) \quad = \quad T(s) + aU'\left(\frac{1}{r}\right), \tag{5.38}$$

and the cost in the clockwise direction is

$$D^+(s;r) \quad = \quad T(1) - T(s) + aU'\left(\frac{1}{1-r}\right). \tag{5.39}$$

A routing r is in equilibrium if the costs of source r in both directions are the same.

Definition 5.1. *A routing r is called an* equilibrium routing *if $D^-(r;r) = D^+(r;r)$. It is denoted by r_a or $r(a)$.*

By definition, r_a is the solution of

$$g(r) \quad := \quad 2T(r) - T(1) + a\left(U'\left(\frac{1}{r}\right) - U'\left(\frac{1}{1-r}\right)\right) = 0. \tag{5.40}$$

Since $g(0) < 0$, $g(1) > 0$ and $g'(r) > 0$, the equilibrium r_a is in $(0,1)$ and is unique. Given a routing r, its utility is

$$V(r) \quad := \quad rU\left(\frac{1}{r}\right) + (1-r)U\left(\frac{1}{1-r}\right).$$

The maximum utility achieved by minimum-cost routing, with parameter a, is then $V(r_a) \leq V(r^*) = V^*$.

The next result implies that r_a varies between r_0 and r^* and converges monotonically to r^* as $a \to \infty$. As a result, the loss $V^* - V(r_a) \geq 0$ in utility also approaches 0 as $a \to \infty$. Denote the interval in which $1/r_a$ and $1/(1-r_a)$ vary as $I := [2, 1/\min\{r_0, 1-r_0\}]$.

Theorem 5.5. *Suppose U'' exists and is bounded on I. For all $a \geq 0$, $|r_a - r^*|$ is a strictly decreasing function of a. Moreover, as $a \to \infty$, $|r_a - r^*|$ and $V^* - V(r_a)$ approach 0.*

Proof. The equation (5.40) defines the equilibrium routing $r(a) := r_a$ as an implicit function of a. By the implicit function theorem, $r'(a)$ satisfies

$$\frac{1}{r'(a)}\left[U'\left(\frac{1}{1-r_a}\right)-U'\left(\frac{1}{r_a}\right)\right]=2\tau(r_a)-\frac{a}{r_a^2}U''\left(\frac{1}{r_a}\right)-\frac{a}{(1-r_a)^2}U''\left(\frac{1}{1-r_a}\right).$$

The right-hand side is positive since U is strictly concave. Hence $r'(a)$ has the same sign as the term in the square bracket, i.e., since U' is decreasing,

$$r'(a) \;=\; \begin{cases} > \;\; 0 & \text{if } r_a < r^* \\ < \;\; 0 & \text{if } r_a > r^* \\ = \;\; 0 & \text{if } r_a = r^* \end{cases}. \tag{5.41}$$

This implies that $|r_a - r^*|$ is a strictly decreasing function of a; see Figure 5.3.

Hence $|r_a - r^*|$ converges to a limit as $a \to \infty$. Since U'' is bounded on the closed interval I, so is U'. Hence, from (5.40), we must have

$$U'(1/r_a)-U'(1/(1-r_a))\to 0, \quad \text{or} \quad U'(1/\lim_{a\to\infty} r_a)=U'(1/(1-\lim_{a\to\infty} r_a)).$$

Since U' is strictly decreasing, this implies that $\lim_{a\to\infty} r_a = 1-\lim_{a\to\infty} r_a = r^*$.

To show that $V^* - V(r_a) \geq 0$ also converges to 0, note that $V'(r^*) = 0$ and hence we have, by Taylor expansion,

$$V(r_a)-V^* \;=\; \frac{1}{2}V''(u)(r_a-r^*)^2$$

for some u between r_a and r^*. Here

$$V''(u) \;=\; \frac{1}{u^3}U''\left(\frac{1}{u}\right)+\frac{1}{(1-u)^3}U''\left(\frac{1}{1-u}\right)\geq -\frac{2\mu}{(\min\{r_0,1-r_0\})^3}$$

where μ is the upper bound of U'' on I. Hence

$$0 \;\leq\; V^*-V(r_a) \;\leq\; \frac{\mu(r_a-r^*)^2}{(\min\{r_0,1-r_0\})^3}.$$

Since $|r_a - r^*| \to 0$, the proof is complete. □

The shape of $r'(a)$ in (5.41) implies that, if $r(0) > r^*$ then $r(a) \geq r^*$ for all a but $r(a)$ decreases to r^* as $a \to \infty$, and if $r(0) < r^*$ then $r(a) \leq r^*$ for all a but $r(a)$ increases to r^* monotonically, as illustrated in Figure 5.3. This is a consequence of the continuity of $r(a)$.

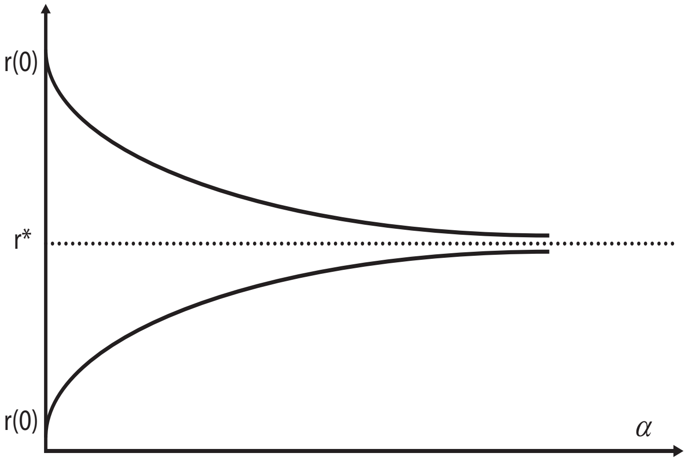


Figure 5.3: The routing $r(a)$.

5.5.4 Stability of minimum-cost routing

We now turn to the stability of r_a. For simplicity, we will take $U(x) = \log x$, the utility function of TCP Vegas [101] and FAST [69]. With this logarithm utility, $V'(r_a) = \log(1 - r)/r$ and hence Theorem 5.5 can be strengthened to show that $V^* - V(r_a)$ is a strictly decreasing function of a, and hence converges monotonically to 0 as $a \to \infty$.

Given r, let $f(r)$ denote the solution of

$$D^-(s; r) \;\; = \;\; D^+(s; r).$$

It is in the range $[0, 1]$ if and only if $0 \leq T(s) \leq T(1)$, or if and only if

$$r^* - \frac{T(1)}{2a} \;\; \leq \; r \; \leq \;\; r^* + \frac{T(1)}{2a}.$$

We will assume that $\min_{u\in[0,1]} \tau(u) > 0$. Then T^{-1} exists and

$$f(r) \;=\; T^{-1}\left(\frac{1}{2}(T(1)+a) - ar\right). \tag{5.42}$$

The routing iteration is

$$r(t+1) \;=\; [f(r(t))]_0^1\,, \tag{5.43}$$

where $[r]_0^1 = \max\{0, \min\{1, r\}\}$.

Definition 5.2. *The equilibrium routing r_a is* (globally) stable*, if starting from any routing $r(0)$, $r(t)$ defined by (5.42)–(5.43) converges to r_a as $t \to \infty$.*

Example 2: Uniform τ

Suppose delay is uniform on the ring, $\tau(u) = \tau$ for all $u \in [0,1]$, so that $T(r) = r\tau$. From (5.40), the equilibrium routing is

$$r_a \;=\; \frac{1}{2} \;=\; r^*, \quad \forall a \geq 0$$

coinciding with the utility-maximizing routing r^*.

Suppose $a < \tau$. Then the routing iteration becomes

$$r(t+1) \;=\; \frac{1}{2\tau}(\tau + a) - \frac{a}{\tau}r(t) \;=\; f(r(t)).$$

Since $|f(s) - f(r)| = (a/\tau)|s-r| < |s-r|$, $f(r)$ is a contraction mapping and hence r_a is globally stable for all $0 \leq a < \tau$.

Hence for the uniform delay case, adding a static component to link cost stabilizes routing provided the weight on prices is smaller than link delay. Moreover, the static component does not lead to any loss in utility ($r_a = r^*$). The stability condition generalizes to the general delay case. The following theorem says that if a is smaller than the minimum “link delay,” then r_a is globally stable; if a is bigger than the maximum “link delay,” then it is globally unstable (diverge from any initial routing except r_a); otherwise, it may converge or diverge depending on initial routing.

Theorem 5.6. *The stability is affected by parameter a such that*

1. *If $a < \min_{u\in[0,1]} \tau(u)$ then r_a is globally stable.*
2. *Suppose $a \geq T(1)$. Then there exists $\underline{r} < r_a < \overline{r}$ such that*
   (a) *If $r(0) = \underline{r}$ or $r(0) = \overline{r}$ then subsequent routings oscillate between $\overline{r}$ and $\underline{r}$.*
   (b) *If $r(0) < \underline{r}$ or $r(0) > \overline{r}$ then subsequent routings after a finite number of iterations oscillate between 0 and 1.*
   (c) *If $\underline{r} < r(0) < \overline{r}$ then $r(t)$ converges to r_a provided $a < \min_{u\in(\underline{r},\overline{r})} \tau(u)$.*
3. *If $a > \max_{u\in[0,1]} \tau(u)$ then starting from any initial routing $r(0) \neq r_a$, subsequent routings after a finite number of iterations oscillate between 0 and 1.*

Proof. 1. We show that the routing iteration (5.43) is a contraction mapping if $a < \min_{u\in[0,1]} \tau(u)$. Now

$$\begin{aligned} \left|[f(s)]_0^1 - [f(r)]_0^1\right| &\leq |f(s) - f(r)|, \\ &= \left|T^{-1}\left(\frac{T(1)+a-2as}{2}\right) - T^{-1}\left(\frac{T(1)+a-2ar}{2}\right)\right|, \\ &= \left|\frac{1}{T'(u)}(as-ar)\right|, \\ &\leq \frac{a}{\min_{u\in[0,1]}\tau(u)}|s-r|, \end{aligned}$$

for some u between r and s, by the mean value theorem. Hence $h(r)$ is a contraction mapping and starting from any $r(0) \in [0,1]$, $r(t)$ converges exponentially to r_a.

2. Define $h(r) = (T(1)+a)/2 - ar$. The routing iteration can be written as

$$T(r(t+1)) \quad = \quad [h(r(t))]_0^1. \tag{5.44}$$

Define the following sequences

$$\begin{aligned} a_0 &= 0, \qquad & b_0 = T(0), \\ a_{n+1} &= h^{-1}(b_n), \qquad & b_{n+1} = T(a_{n+1}). \end{aligned}$$

Note that $(a_n, n \geq 0)$ is a routing sequence going backward in time. The following lemma is proved in the appendix, following [103].

Lemma 5.1. *Let $T_a = T(r_a) = h(r_a)$. Then*

$$\begin{aligned} 0 = a_0 \ < \ a_2 \ < \cdots < \ \ r_a \ \ < \cdots < \ a_3 \ < \ a_1 < 1, \\ T(0) = b_0 \ < \ b_2 \ < \cdots < \ \ T_a \ \ < \cdots < \ b_3 \ < \ b_1 < T(1). \end{aligned}$$

Since the sequences are monotone, the lemma implies that there are $\underline{r}$ and $\overline{r}$ with $0 < \underline{r} < r_a < \overline{r} < 1$ such that

$$\lim_{n\to\infty} a_{2n} = \underline{r}, \quad \text{and} \quad \lim_{n\to\infty} a_{2n+1} = \overline{r}.$$

By continuity of T and h, we have

$$T(\underline{r}) = h(\overline{r}), \quad \text{and} \quad T(\overline{r}) = h(\underline{r}).$$

This implies that starting from $r(0) = \underline{r}$ or $r(0) = \overline{r}$, the subsequent routings oscillate between $\underline{r}$ and $\overline{r}$.

To show the second claim, suppose $r(0) < \underline{r}$. Specifically, suppose $a_{2n-2} < r(0) < a_{2n}$ for some n. If $h(r(0)) > T(1)$ (possible since $a \geq T(1)$), then $r(1) = 1$ and subsequent routings oscillate between 0 and 1. Otherwise, from (5.44), $r(0) = h^{-1}(T(r(1)))$, and hence $a_{2n-2} < h^{-1}(T(r(1))) < a_{2n}$. Since h is strictly decreasing, we have $b_{2n-1} < T(r(1)) < b_{2n-3}$ by definition of b_n. Hence, since T is strictly increasing, $a_{2n-1} < r(1) < a_{2n-3}$. The same argument then shows that $a_{2n-4} < r(2) < a_{2n-2}$. Hence we have shown that $r(0) < a_{2n}$ implies $r(2) < a_{2n-2}$. This proves the second claim. The proof of the third claim follows the same argument of part 1.

3. By the mean value theorem, we have, for all α, α' in $[0,1]$,

$$|h^{-1}(T(\alpha)) - h^{-1}(T(\alpha'))| \quad = \quad \frac{T'(u)}{a}|\alpha - \alpha'|,$$

for some u between α and α'. Hence the iteration map

$$a_{n+1} \quad = \quad h^{-1}(T(a_n))$$

is a contraction provided $a > \max_{u\in[0,1]} \tau(u)$. This implies that the sequence $(a_n, n \geq 0)$ converges and, since r_a is the unique fixed point of $h^{-1}(T(\cdot))$, $\underline{r} = \overline{r} = r_a$. The assertion then follows from part 2(b). □

5.5.5 General network

It is difficult to derive an analytical bound on a to guarantee routing stability or to compute optimal routing for general networks. In this section, we present numerical results to illustrate that the intuition from the simple ring network analyzed in the previous subsections extends to general topology.

We generate a random network based on Waxman's algorithm [159]. The nodes are uniformly distributed in a two-dimensional plane. The probability that a pair of nodes u, v are connected is given by

$$\text{Prob}(u,v) \quad = \quad \alpha \ \exp\left(\frac{d(u,v)}{\beta L}\right),$$

where the maximum probability $\alpha > 0$ controls connectivity, $\beta \leq 1$ controls the length of the edges with a larger β favoring longer edges, $d(u,v)$ is the Euclidean distance between nodes u and v, and L is the maximum distance between any two nodes. In our example, we set the number of nodes $N = 30$, $\alpha = 0.8$, $\beta = 0.3$, which generated 90 bidirectional links; see Figure 5.4. The fixed delay τ_l of each link l is randomly chosen according to a uniform distribution over [100, 400]ms. The link capacities are randomly chosen from the interval

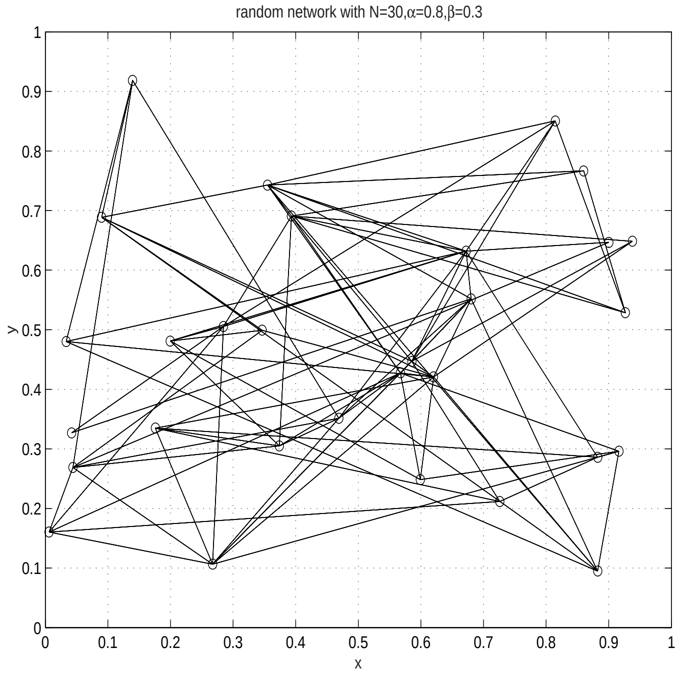


Figure 5.4: A random network.

[1000, 4000] packets/sec, also with uniform distribution. There are 60 flows on the network with randomly chosen source and destination nodes. Routing on this network is computed using the Bellman-Ford minimum-cost algorithm, with link cost $d_l(t) = \tau_l + ap_l(t)$ in each update period t, on a slower timescale than congestion control. In each routing period t, we first solve the link prices based on the current routing, using the gradient projection algorithm of [97]. We iterate the source algorithm to update rates and the link algorithm to update prices, until they converge. The link prices are then used to compute the minimum-cost for the next period.

We measure the performance of the scheme at different a by the sum of all of the source's utilities. If the routing is stable (at small a), the aggregate utility is computed using the equilibrium routing. Otherwise, the routing oscillates and the time-averaged aggregate utility is used. The result is shown in Figure 5.5.

As expected, when a is small, routing is stable and the aggregate utility increases with a, as in the ring network analyzed in Section 5.5.3 (Theorem 5.5). When $a < 4$, the static delay τ_l dominates the link cost, and the routes computed with $d_l(t)$ remain the same as with static routing ($a = 0$), and hence the aggregate utility is independent of a. Routing

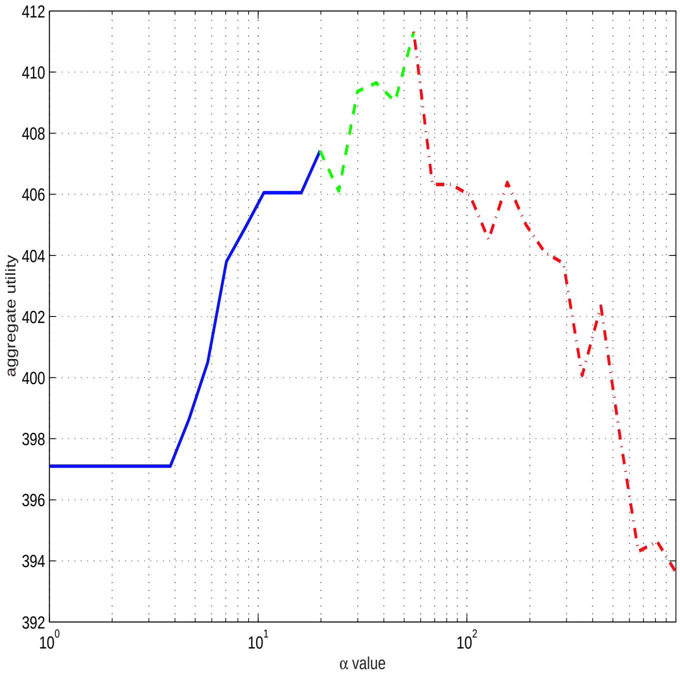


Figure 5.5: Aggregate utility as a function of a for random network

becomes unstable at around $a = 10$. Even though the time-averaged utility continues to rise after routing instability sets in, eventually it peaks and drops to a level less than the utility achievable by the necessarily stable static routing.

5.6 Resource provisioning

Results in the previous sections show that even though an equilibrium of TCP/IP, when it exists, maximizes utility under pure dynamic routing, it can be unstable and hence not attainable by the TCP/IP system. In this section, we show that if the link capacities are optimally provisioned, however, pure *static* routing is enough to maximize utility. Moreover, it is optimal even within the class of multi-path routing: again, there is no penalty in not splitting traffic across multiple paths.

Suppose it costs $\alpha_l > 0$ amount to provision a unit of capacity at link l, and let $\alpha = (\alpha_l,$ for all $l)$ be the vector of unit costs. For instance, a longer link may have a larger α_l. The total capacity cost over the entire network is $\alpha^T c$. Suppose the budget for provisioning the capacity is B. Consider the problem of optimally selecting capacities, routing, and source rates to maximize utility

$$\max_{c\geq 0}\max_{R\in\mathcal{R}_m}\max_{x\geq 0} \quad \sum_i U_i(x_i), \tag{5.45}$$

$$\text{subject to} \quad Rx \leq c, \tag{5.46}$$

$$\alpha^T c \leq B. \tag{5.47}$$

where U_i are concave increasing utility functions. Note that R ranges in $\mathcal{R}_m$ and hence multi-path routing is allowed. This problem may arise when optical lightpaths can be dynamically reconfigured at a connection timescale.

Theorem 5.7. *Suppose $U_i'(x_i) > 0$ for all i and $x_i \geq 0$. At optimality,*

1. *There is an optimal solution (c^*, R^*, x^*) where $R^* \in \mathcal{R}_s$ is a single-path routing.*
2. *Moreover, R^* is pure static routing using α_l as link costs.*
3. *$R^*x^* = c^*$, i.e., there is no slack capacity.*
4. *$\alpha^T c = B$, i.e., there is no slack in budget.*
5. *Link prices generated by TCP–AQM are proportional to the provisioning costs, $p^* = \gamma^*\alpha$ for some $\gamma^* > 0$.*

Proof. It is easy to show the existence of an equilibrium. Define the Lagrangian of (5.45)-(5.47) as

$$L(c, R, x, p, \gamma) = \sum_i U_i(x_i) - p^T(Rx - c) - \gamma(\alpha^T c - B).$$

At optimality, the KKT condition holds: there exist $p^* \geq 0$ and $\gamma^* \geq 0$ such that

$$U^{'}(x_i^*) = (R^*)^T p^*, \tag{5.48}$$

$$p^* = \gamma^* \alpha, \tag{5.49}$$

$$(p^*)^T(R^*x^* - c^*) = 0, \tag{5.50}$$

$$\gamma^*(\alpha^T c^* - B) = 0. \tag{5.51}$$

From (5.49), we obtain the last claim in the theorem. Moreover, (5.49) and $U_i'(x_i^*) > 0$ imply that $\gamma^* > 0$ and $p_l^* > 0$ for *all* l, since $\alpha > 0$. Hence, from (5.50) and (5.51), equality holds in (5.46) and (5.47), proving the third and fourth claims.

To prove the first two claims, express the routing matrix R as $R = HW$ where $W \in \mathcal{W}_m$. Using the equalities in (5.46) and (5.47) to eliminate c, we can transform the utility maximization problem (5.45)–(5.47) into:

$$\begin{aligned} \max_{W \in \mathcal{W}_m} \max_{x \geq 0} \quad & \sum_i U_i(x_i), \\ \text{subject to} \quad & \sum_i \left(\alpha^T H^i w^i \right) x_i \; = \; B, \end{aligned}$$

where W =diag(w^i). Since U_i is nondecreasing and both the objective and the constraints above are separable in i, in order to maximize utility, w^i should be chosen to be a solution of

$$\begin{aligned} \min_{w^i} \quad & \alpha^T H^i w^i, \\ \text{subject to} \quad & \mathbf{1}^T w^i = 1, \quad 0 \leq w_j^i \leq 1. \end{aligned}$$

Since this is a linear program, there exists an optimal point on the boundary. Hence there is an optimal $W^* \in \mathcal{W}_s$, i.e., minimum-cost single-path routing using α_l as link costs is optimal. □

5.7 Conclusion

Given a routing, TCP-AQM can be interpreted as a distributed primal-dual algorithm over the Internet to maximize aggregate utility over source rates. In this chapter, we study whether TCP-AQM together with IP (modelled by minimum-cost routing) can maximize utility over both source rates and routing, on a slower time scale. We show that we can indeed interpret TCP/IP as *attempting* to maximize utility in the sense that its equilibrium, if it exists, solves the utility maximization problem and its dual, provided congestion prices generated by TCP-AQM are used as link costs. TCP/IP equilibrium exists if and only if there is no penalty in not splitting traffic across multiple paths. Even if equilibrium exists, however, TCP/IP with pure dynamic routing can be unstable. Specializing to a special ring network, we show that routing is indeed unstable when link costs are congestion prices. It can be stabilized by adding a static component to the definition of link cost, but the static component reduces the achievable utility. There thus seems to be an inevitable tradeoff between routing stability and utility maximization, for a given set of link capacities. We show, however, that if link capacities are optimally provisioned, then pure static (and hence stable) routing is sufficient to maximize utility even for general networks, and link costs are proportional to the provisioning costs. Moreover single-path routing can achieve the same utility as multi-path routing. Hence, one can regard the layering of TCP and IP as a decomposition of the utility maximization problem over source rates and routes into a distributed and decentralized algorithm, carried out on different time scales, at least when network capacities are well provisioned.

5.8 Appendix

5.8.1 Proof of duality gap

We prove that there is generally a duality gap between the primal problem (5.22)–(5.23) and its dual when N is odd.

It is easy to see that the primal optimal routing is

$$r^* \;=\; \frac{N-1}{2}, \quad \text{or} \quad \frac{N+1}{2}.$$

Suppose without loss of generality that $r^* = (N-1)/2$ (the other case is similar). Then, the source rates are

$$x_1 = \cdots = x_{r^*} = \frac{2}{N-1}, \qquad \text{and} \qquad x_{r^*+1} = \cdots = x_N = \frac{2}{N+1}$$

yielding a primal objective value of

$$\begin{aligned} & \frac{N-1}{2}U\left(\frac{2}{N-1}\right)+\frac{N+1}{2}U\left(\frac{2}{N+1}\right) \\ = \;& N\left\{\left(\frac{1}{2}-\frac{1}{N}\right)U\left(\frac{2}{N-1}\right)+\left(\frac{1}{2}+\frac{1}{N}\right)U\left(\frac{2}{N+1}\right)\right\} \\ < \;& NU\left(\frac{2}{N}\right), \end{aligned}$$

where the last inequality follows from the strict concavity of U. We now show that the right-hand side is the optimal dual objective value, and hence there is a duality gap.

The dual problem of (5.22)–(5.23) is (e.g., [97])

$$\min_{p_1,p_{N+1}\geq 0}\left(\sum_{i=1}^{N}\max_{x_i}\ \phi(x_i,p_1,p_{N+1})+(p_1+p_{N+1})\right),$$

where $\phi(x_i,p_1,p_{N+1})=U(x_i)-x_i\ \min\{p_1,p_{N+1}\}$. First, note that the minimizing (p_1,p_{N+1}) must satisfy $p_1=p_{N+1}$, for otherwise, if (say) $p_1<p_{N+1}$, then the dual objective value is

$$\sum_{i=1}^{N}\max_{x_i}\ (U(x_i)-x_ip_1)+(p_1+p_{N+1})$$

and can be reduced by decreasing p_{N+1} to p_1. Hence the dual problem is equivalent to

$$\min_{p\geq 0}\qquad \sum_{i=1}^{N}\max_{x_i}\ (U(x_i)-x_i\,p)+2p. \tag{5.52}$$

Let p^* denote the minimizer and $x_i^*=x_i(p^*)=x(p^*)=:x^*$ denote the corresponding maximizers (they are equal for all i by symmetry). Then we have

$$U'(x^*)\;\;=\;\;p^*. \tag{5.53}$$

Differentiating the objective function in (5.52) with respect to p and setting it to zero, we have

$$0\;\;=\;\;N(U'(x^*)x'(p^*)-p^*x'(p^*)-x^*)+2. \tag{5.54}$$

Using (5.53), we have $x^* = 2/N,$ and hence the minimum dual objective value is

$$N(\max_{x^*} U(x^*) - x^* p^*) + 2p^* \quad = \quad NU\left(\frac{2}{N}\right)$$

as desired. □

5.8.2 Proof of primal-dual optimality

We prove that the solution given by (5.37) is primal-dual optimal using the saddle-point theorem (e.g., [14, pp. 427]). Clearly, (r^*, x^{-*}, x^{+*}) is primal feasible and (p^{-*}, p^{+*}) is dual feasible. We now show that $(r^*, x^{-*}, x^{+*}, p^{-*}, p^{+*})$ is a saddle point, i.e., for all (r, x^-, x^+, p^-, p^+). Now

$$L(r, x^-, x^+, p^{-*}, p^{+*}) \quad \leq \quad L(r^*, x^{-*}, x^{+*}, p^{-*}, p^{+*}) \leq L(r^*, x^{-*}, x^{+*}, p^-, p^+).$$

For the right inequality, substituting (r^*, x^{-*}, x^{+*}) from (5.37) into $L(r^*, x^{-*}, x^{+*}, p^-, p^+)$ to get, for all (p^-, p^+),

$$L(r^*, x^{-*}, x^{+*}, p^-, p^+) \quad = \quad U(2).$$

But $U(2) = L(r^*, x^{-*}, x^{+*}, p^{-*}, p^{+*})$, establishing the right inequality. For the left inequality, denoting $p^* := p^{-*} = p^{+*}$, from (5.37) we have

$$\begin{aligned} L(r, x^-, x^+, p^{-*}, p^{+*}) \quad &= \quad rU(x^-) + (1-r)U(x^+) - (rx^- + (1-r)x^+)p^* + 2p^* \\ &\leq \quad U(y) - yp^* + 2p^* \quad \text{(concavity of } U\text{)}, \end{aligned} \tag{5.55}$$

with $y := rx^- + (1-r)x^+$, where equality holds if and only if $x^- = x^+$ since U is *strictly* concave. Notice that the right-hand side is maximized over y if and only if y satisfies $U'(y) = p^*$. This implies that $y = x^{-*} = x^{+*} = 2$ since U' is strictly monotonic. Substitute $y = 2$ into (5.55) yields, for all (r, x^-, x^+),

$$L(r, x^-, x^+, p^{-*}, p^{+*}) \quad \leq \quad U(2)$$

as desired, since $U(2) = L(r^*, x^{-*}, x^{+*}, p^{-*}, p^{+*})$. □

5.8.3 Proof of Lemma 5.1

We will prove the lemma by induction. Note that $b_0 < T_a$ implies that $a_1 = h^{-1}(b_0) > h^{-1}(T_a) = r_a$. Since $a \geq T(1)$ and $h(1) < 0$, $a_1 = h^{-1}(b_0) < 1$ (see Figure 5.6). Hence

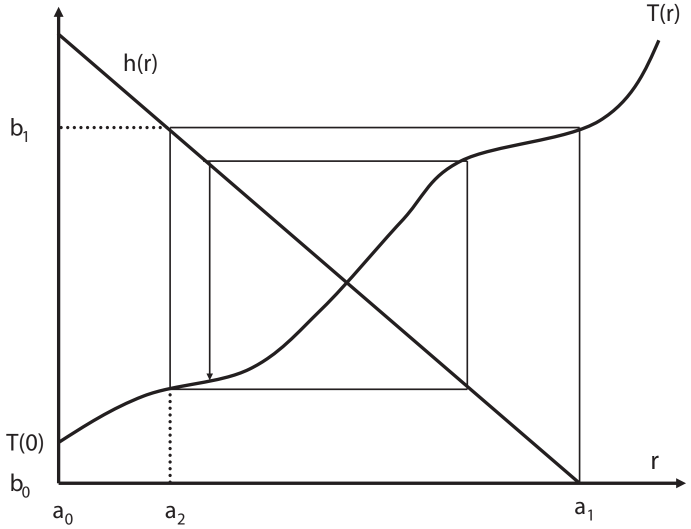


Figure 5.6: Proof of Lemma 5.1.

$$0 = a_0 < r_a < a_1 < 1.$$

This implies that $b_1 = T(a_1)$ satisfies

$$T(0) = b_0 < T_a < b_1 < T(1).$$

Since $b_1 < T(1) < h(0)$, $a_2 = h^{-1}(b_1) > h^{-1}(h(0)) = 0$, we have

$$0 = a_0 < a_2 < r_a < a_1 < 1.$$

Let the induction hypothesis be

$$\begin{array}{c} a_0 < \ldots < a_{2n} < r_a < a_{2n-1} < \ldots < a_1 \\ b_0 < \ldots < b_{2n-2} < T_a < b_{2n-1} < \ldots < b_1. \end{array}$$

Then $b_{2n} = T(a_{2n}) > T(a_{2n-2}) = b_{2n-2}$ and $b_{2n} = T(a_{2n}) < T(r_a) = T_a$. Hence,

$$b_{2n-2} < b_{2n} < T_a.$$

This implies that $r_a < a_{2n+1} < a_{2n-1}$, which in turn implies that $T_a < b_{2n+1} < b_{2n-1}$. This completes the induction. □

Chapter 6

Throughput, Fairness, and Capacity

6.1 Introduction

Recent studies, e.g. [80, 97, 116, 164, 101, 88, 96], have shown that a bandwidth allocation policy can be formulated as a utility maximization problem where the bandwidth allocation x^* (source rates) solves [80]

$$\max_x \sum_i U_i(x_i) \quad \text{subject to } Rx \leq c. \tag{6.1}$$

It is remarkable that as long as traffic sources adapt their rates to the aggregate (sum of) congestion in their paths, they are implicitly maximizing some utility. The optimization problem (6.1) is a convenient characterization of the equilibrium properties of various TCP/AQM systems. We can derive the underlying utility functions of various TCP algorithms and use them to study the relations among network throughput, fairness, and capacity. Our work reveals some counter-intuitive behaviors, which will be briefly presented in this chapter. See [144, 146] for more detailed results and proofs.

We refer to network throughput as the total traffic through the network, which measures the efficiency of the bandwidth allocation policy under which the network operates. There are many examples in the literature that point to an inevitable tradeoff between fairness and aggregate throughput (efficiency), yet there is no general theorem clarifying this folklore. How do we balance fairness and efficiency in designing bandwidth allocation policies? Will adding additional link capacities necessarily result in higher aggregate throughput?

In this chapter, we rigorously study these questions in general networks using an analytical model . Here are our main results.

Suppose that the bandwidth allocation policies, represented by utility functions, are parameterized by a common scalar $\alpha \geq 0$. We derive explicit expressions for the changes in source rates and congestion prices when the parameter α or the capacities change for general utility functions.

We specialize to a particular class of utility functions [116] that characterize various TCP variants and include various fairness criterions as special cases. The parameter α in these utility functions can be interpreted as a quantitative measure of fairness [107, 16], and an allocation is *fair* if α is large. All examples in the literature indicate that a fair allocation is necessarily inefficient. We quantitatively formulate the relations between fairness and efficiency in general networks. This characterization allows us both to produce the first counter-example (Theorem 6.3) and trivially explain all the previous supporting examples (Corollary 6.2). Surprisingly, the class of networks in our counter-example indicates that a fairer allocation could be *always* more efficient. In particular it implies that max-min fairness can achieve a higher aggregate throughput than proportional fairness.

Intuitively, we might expect that the aggregate throughput will always rise when some links increase their capacities. This turns out to be wrong, and we characterize exactly the condition under which this is true (Theorem 6.4). Not only can the aggregate throughput be reduced when some link increases its capacity, more strikingly, it can also be reduced even when *all* links increase their capacities by the same amount (Theorem 6.5). Moreover, this holds under all bandwidth allocation policies . This paradoxical result seems less surprising in retrospect: raising link capacities always increases the aggregate utility, but mathematically there is no a priori reason that it should also increase the aggregate throughput. If all links increase their capacities proportionally, however, the aggregate throughput will indeed increase, under the class of utility functions proposed in [116] (Theorem 6.6).

It is well known that counter-intuitive behavior can arise in a distributed system where agents optimize their own objectives, e.g., the Braess paradox in transportation networks. It was discovered theoretically in 1968 [18, 120, 44] and verified in real world years later [37]. It shows that adding a new road to a transportation network may cause *longer* travel time for *every* car. Subsequent paradoxes have been discovered in mechanical and electrical networks [27], in queueing networks [28, 136, 83, 11, 84], and in computer systems [72, 73]. Even though our results have the same flavor, they differ in important ways from the Braess paradox.

First, in the Braess paradox, the performance degradation is due to misalignment of individual and social optimalities. In our case, it is due to misalignment of two social objectives (utility maximization versus throughput maximization). Second, in the Braess paradox, the addition of new road leads to degraded performance for *all* flows, and hence the new equilibrium point is not Pareto optimal. In our case, all equilibrium points are Pareto optimal, and hence some flows are worse off and some better off in the new equilibrium point. Finally, examples of the Braess paradox always involve the addition of new paths and flows that re-route to maximize their own objectives. In our case, only link capacities are changed, while network topology and routing are fixed.

6.2 Model

A network consists of L links with finite capacity c_l. It is shared by N sources indexed by i. R is the routing matrix where $R_{li} = 1$ if source i uses link l and 0 otherwise. Let x_i be the transmission rate of source i, and $U_i(x_i;\alpha)$ be its utility. All the utility functions $U_i(x_i;\alpha)$ are parameterized by a scalar $\alpha \geq 0$. Suppose $U_i(x_i;\alpha)$ are concave in x_i for $\alpha \geq 0$ and strictly concave when $\alpha > 0$. When α and c are clear from the context, we may use $U_i(x_i)$ in place of $U_i(x_i;\alpha)$.

Consider the utility maximization problem

$$\max_{x \geq 0} \sum_i U_i(x_i;\alpha) \quad \text{subject to} \quad Rx \leq c, \tag{6.2}$$

and its Lagrangian dual

$$\min_{p \geq 0} \sum_i \max_{x_i \geq 0} \left(U_i(x_i;\alpha) - x_i \sum_l R_{li} p_l \right) + \sum_l c_l p_l. \tag{6.3}$$

A maximizer $x = x(\alpha, c)$ for (6.2) and a minimizer $p = p(\alpha, c)$ for (6.3) exist for $\alpha \geq 0, c > 0$, since the utility functions $U_i(x_i)$ are concave.

Unless otherwise specified, we will assume that $\alpha > 0$, and R is full row rank, so that the solutions $x = x(\alpha, c)$ and $p = p(\alpha, c)$ are unique. The aggregate throughput $T = T(\alpha, c)$ is defined in terms of the unique solution,

$$T(\alpha, c) \ := \ \sum_i x_i(\alpha, c). \tag{6.4}$$

From Lemma 6.1 below, $x(\alpha, c)$ and $p(\alpha, c)$ are continuous functions of α and c. Moreover, they are differentiable except at a finite number of points when the active constraint set at optimal changes as α or c is perturbed. We can study $\partial T / \partial \alpha$ and $\partial T / \partial c$ in between these points. Hence, all our statement should be interpreted piecewise in between non-differentiable point. For the rest of the paper, we will thus focus on the utility maximization with equality constraints that represent only those constraints that are active at optimality

$$\max_{x_i \geq 0} \ \sum_i U_i(x_i; \alpha) \ \text{ s.t. } \ Rx = c. \tag{6.5}$$

In this case the dual problem (6.3) should be interpreted as the Lagrangian dual of (6.2) with a possibly reduced R, as opposed to the dual of (6.5).

Suppose R has full row rank. Suppose $N \geq L$ and let $M = N - L$ be the difference between the number of sources and the number of links. Then M is the dimension of the null space of R. Let $(z_m, m = 1, \ldots, M)$, $z_m \in \Re^N$ be any basis of the null space of R, and let $Z = [z_1 \ z_2 \ \ldots \ z_M]$ be the matrix with z_m as its columns. Let $V = V(x; \alpha) := \sum_i U_i(x_i; \alpha)$ be the aggregate utility function. Let $D = D(\alpha, c)$ denote the curvature of the aggregate utility function

$$D \ := \ -\frac{\partial^2 V}{\partial x^2}, \tag{6.6}$$

and $b = b(\alpha, c)$ be

$$b \;:=\; \frac{\partial^2 V}{\partial x \partial \alpha} \tag{6.7}$$

at the optimal allocation $x = x(\alpha, c)$.

6.3 Basic results

In the rest of this chapter, we assume that the active constraint set is unchanged when α or c is perturbed locally (i.e., we consider problem (6.5) instead of problem (6.2)). When R is full row rank, the following lemma cited from [150] guarantees the differentiability of $x(\alpha, c)$ and $p(\alpha, , c)$.

Lemma 6.1. *For any $\alpha > 0$, $c > 0$, the unique solution $x(\alpha, c)$ and $p(\alpha, c)$ of (6.5) is continuous and differentiable at (α, c).*

The basic results on how throughput and prices vary as the utility parameter α and capacity c change are given in the next theorem. In the next two sections, we will specialize to a particular class of utility functions to study the throughput-fairness tradeoff and whether increasing capacity always raises throughput.

Theorem 6.1. *The optimal rates $x = x(\alpha, c)$ of (6.5) and optimal prices $p = p(\alpha, c)$ of (6.3) satisfy the following equations*

$$\begin{aligned}
\frac{\partial x}{\partial \alpha} &= (D^{-1} - D^{-1}R^T(RD^{-1}R^T)^{-1}RD^{-1})b, \\
\frac{\partial x}{\partial c} &= D^{-1}R^T(RD^{-1}R^T)^{-1}, \\
\frac{\partial p}{\partial \alpha} &= (RD^{-1}R^T)^{-1}RD^{-1}b, \\
\frac{\partial p}{\partial c} &= -(RD^{-1}R^T)^{-1},
\end{aligned}$$

where matrix D and vector b are defined in (6.6) and (6.7), respectively.

Proof: At the optimal point, the Karush-Kuhn-Tucker condition holds. We have

$$Rx = c \quad \text{and} \quad R^T p - \frac{\partial V}{\partial x} = 0. \tag{6.8}$$

Define

$$y := \begin{pmatrix} x \\ p \end{pmatrix}, \quad w := \begin{pmatrix} c \\ \alpha \end{pmatrix}, \quad \text{and} \quad G(w, y) = \begin{pmatrix} Rx - c \\ R^T p - \frac{\partial V}{\partial x} \end{pmatrix}.$$

Then (6.8) can be rewritten as $G(w, y) = 0$. The derivatives of function G are

$$\begin{aligned} \frac{\partial G}{\partial y} &= \begin{pmatrix} R & 0 \\ -\frac{\partial^2 V}{\partial x^2} & R^T \end{pmatrix} = \begin{pmatrix} R & 0 \\ D & R^T \end{pmatrix}, \\ \frac{\partial G}{\partial w} &= \begin{pmatrix} -I & 0 \\ 0 & -\frac{\partial^2 V}{\partial x \partial \alpha} \end{pmatrix} = -\begin{pmatrix} I & 0 \\ 0 & b \end{pmatrix}. \end{aligned}$$

Since R is full row rank and D is positive definite, $RD^{-1}R^T$ is positive definite. Then $\partial G/\partial y$ is always invertible, and it can be checked that

$$\begin{pmatrix} R & 0 \\ D & R^T \end{pmatrix}^{-1} = \begin{pmatrix} D^{-1}R^T(RD^{-1}R^T)^{-1} & D^{-1} - D^{-1}R^T(RD^{-1}R^T)^{-1}RD^{-1} \\ -(RD^{-1}R^T)^{-1} & (RD^{-1}R^T)^{-1}RD^{-1} \end{pmatrix}.$$

All the above matrices are well defined because $RD^{-1}R^T$ is invertible. From the implicit function theorem, the vector y can be uniquely solved in terms of w locally. Moreover

$$\frac{dy}{dw} = -\left(\frac{\partial G}{\partial y}\right)^{-1} \frac{\partial G}{\partial w} = \begin{pmatrix} R & 0 \\ D & R^T \end{pmatrix}^{-1} \begin{pmatrix} I & 0 \\ 0 & b \end{pmatrix}.$$

From the definitions of y and w, we have

$$\begin{aligned} \frac{\partial x}{\partial \alpha} &= (D^{-1} - D^{-1}R^T(RD^{-1}R^T)^{-1}RD^{-1})b, & \frac{\partial x}{\partial c} &= D^{-1}R^T(RD^{-1}R^T)^{-1}, \\ \frac{\partial p}{\partial \alpha} &= (RD^{-1}R^T)^{-1}RD^{-1}b, & \frac{\partial p}{\partial c} &= -(RD^{-1}R^T)^{-1}. \end{aligned}$$

□

Since the optimal x always satisfies the constraints $Rx = c$, for a fixed c, the change in x should be in the null space of R as α varies. This is captured by the following corollary.

Corollary 6.1. *The derivative $\partial x/\partial\alpha$ can also be expressed as*

$$\frac{\partial x}{\partial \alpha} \;=\; Z(Z^T DZ)^{-1}Z^T b,$$

where the columns of matrix Z form a basis of the null space of R.

Proof: Denote

$$\Delta := D^{-1} - D^{-1}R^T(RD^{-1}R^T)^{-1}RD^{-1} - Z(Z^T DZ)^{-1}Z^T.$$

From Theorem 6.1 and the definition of Δ, we only need to show $\Delta = 0$. By the definition of matrix Z we have

$$RZ = 0, \quad \text{and} \quad Z^T R^T = 0.$$

It is clear that

$$\begin{bmatrix} R \\ Z^T D \end{bmatrix} \Delta = \begin{bmatrix} RD^{-1} - RD^{-1} - 0 \\ Z^T - 0 - Z^T \end{bmatrix} = 0.$$

The next step is to show that the matrix $\begin{bmatrix} R \\ Z^T D \end{bmatrix}$ is full rank so that Δ must be the zero matrix. Suppose it is not, then there exists a nonzero vector v such that

$$\begin{bmatrix} R \\ Z^T D \end{bmatrix} v \;=\; 0. \tag{6.9}$$

Hence, $Rv = 0$, i.e., v is in the null space of R. Since the columns of Z form a basis of the null space of R, there exists w such that $v = Zw$. Substituting into (6.9), we have

$$Z^T D v = Z^T D Z w = 0.$$

Since $Z^T DZ$ is positive definite and invertible, we must have $w = 0$ and $v = Zw = 0$. This contradicts the assumption that $v \neq 0$. Therefore $\begin{bmatrix} R \\ Z^T D \end{bmatrix}$ is full rank and $\Delta = 0$. □

We will apply these results to a particular class of utility functions to study the effect of changes in fairness (α) and capacity (c) on throughput in general networks.

6.4 Is fair allocation always inefficient?

Now we apply the expression for $\partial x / \partial \alpha$ in Corollary 6.1 to study the effect of changes in fairness (α) on throughput $T(\alpha) = T(\alpha, c)$ for a fixed $c > 0$. It clarifies a folklore about the tradeoff between efficiency and fairness of a bandwidth allocation policy.

6.4.1 Conjecture

Recent studies show that bandwidth allocations can be formulated as a utility maximization problem [80, 97, 96], and allocation properties such as throughput and fairness can be studied by analyzing the underlying convex optimization problem.

Kelly et al. [80] introduce *proportional fairness*, characterized by utility function $U_i(x_i) = \log x_i$, which is achieved by TCP Vegas and FAST. Massoulie et al. [107] propose another allocation policy *minimum potential delay* with $U_i(x_i) = -1/x_i$, which has been shown to approximate the fairness of TCP Reno by Kunniyur and Srikant [88]. Mo and Walrand [116] present the following class of utility functions

$$U(x_i, \alpha) = \begin{cases} (1-\alpha)^{-1}\, x_i^{1-\alpha} & \text{if } \alpha \neq 1 \\ \log x_i & \text{if } \alpha = 1 \end{cases} . \tag{6.10}$$

It includes all the previously considered allocation policies as special cases–maximum throughput ($\alpha = 0$), proportional fairness ($\alpha = 1$), minimum potential delay ($\alpha = 2$), and max-min fairness ($\alpha = \infty$). It provides a convenient way to compare the fairness of different allocation policies . Moreover, it can also generate utility functions of major TCP congestion control algorithms, e.g., Reno ($\alpha = 2$), HSTCP [39] ($\alpha = 1.2$), and Vegas, FAST [69], Scalable TCP [81] ($\alpha = 1$).

We are not concerned with fairness *across different flows* under the same allocation policy represented by a given α value, as, e.g., Jain's fairness index is [67]. Rather, we want to compare fairness *across allocation policies*. While there are no generally accepted criteria to compare the fairness of allocation policies, many examples in the networking literature (e.g., [107, 116, 16, 105, 139]) informally compare specific allocation policies in terms of their α. For instance, $\alpha = 0$ maximizes throughput but can be extremely unfair. Proportional fairness ($\alpha = 1$) is considered fairer, and max-min fairness ($\alpha = \infty$) the fairest because it generalizes equal sharing at a single resource to a network of resources in a way that maintains Pareto optimality [45, 15]. Comparison of fairness of these polices [107] in a simple network shows that the minimum-potential-delay policy ($\alpha = 2$) "penalizes more (less) severely long routes than max-min (proportional) fairness." We extrapolate this intuition based on special cases to a continuum of allocation policies indexed by α, and interpret α as a quantitative measure of fairness.

Is a fairer policy (larger α) always less efficient (smaller aggregate throughput $T(\alpha)$)? This conjecture is prompted by the various examples in resource allocation in the literature of wired networks [107, 116, 16], wireless networks, [105, 139], economics, [22], etc. These examples seem to illustrate (quoted from [105])

> "*the fundamental conflict between achieving flow fairness and maximizing overall system throughput. ... The basic issue is thus the tradeoff between these two conflicting criteria.*"

This conjecture can be analytically expressed as

Conjecture 6.1. $T(\alpha)$ *is non-increasing*

$$\frac{\partial T}{\partial \alpha} \;\leq\; 0 \quad \textit{for } \alpha > 0.$$

6.4.2 Special cases

We review several examples in the literature that have motivated Conjecture 6.1. The conjecture is checked in special networks for max-min fairness, minimum potential delay, proportional fairness, and the maximum-throughput policy by analytically solving (6.2) or numerically computing $T(\alpha)$. However, these techniques are not applicable to general networks. As is shown in the next subsection, the underlying network topology in these examples possesses a special structure that leads to trivial sufficient conditions for the conjecture to be true.

Example 1: Linear network with uniform capacity [107, 16]

Consider the classical linear network with L unitary capacity links and $N = L + 1$ competing sources, shown in Figure 6.1. The rates $x_i(\alpha)$ are computed by solving (6.2) [16] , which gives

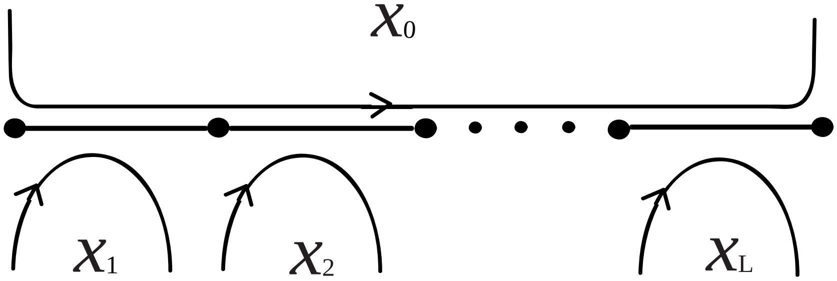


Figure 6.1: Linear network.

$$x_0(\alpha) = \frac{1}{L^{1/\alpha}+1}, \quad \text{and} \quad x_i(\alpha) = \frac{L^{1/\alpha}}{L^{1/\alpha}+1} \text{ for } i \geq 1.$$

Using this, we can easily check that, for $\alpha > 0$,

$$\frac{\partial T}{\partial \alpha} \quad = \quad \frac{-L^{1/\alpha}(L-1)\log L}{\alpha^2\left(1+L^{1/\alpha}\right)^2} \begin{cases} = 0, & L = 1 \\ < 0, & L \geq 2 \end{cases} .$$

Hence, except for the single-link case, $T(\alpha)$ is strictly decreasing in α for the linear network with uniform link capacity. After examining this special case with several special α values, Massoulie et al. [107] made a cautious comment: “It is not known whether the same ordering holds for arbitrary network topologies.”

Example 2: Linear network with nonuniform capacity [116]

The same network topology in Example 1 is considered in [116] with $L = 2$ with link capacities $c_1 < c_2$. The source rates under max-min fairness are

$$x_0(\infty) = x_1(\infty) = \frac{c_1}{2}, \qquad x_2(\infty) = c_2 - \frac{c_1}{2}, \quad T(\infty) = c_2 + \frac{c_1}{2}.$$

It is not hard to solve (6.2) to obtain the source rates and derive the aggregate throughput for proportional fairness

$$T(1) \;=\; \frac{2}{3}c_1 + \frac{1}{3}\sqrt{c_1^2 + c_2^2 - c_1 c_2} + \frac{2}{3}c_2 \geq c_2 + \frac{c_1}{2} = T(\infty),$$

which supports the conjecture for $\alpha = 1$ and $\alpha = \infty$.

Example 3: Linear network with two long flows

Consider a linear network with two long flows, shown in Figure 6.2. The link capacities are $c = (500, 400, 300, 200, 500)^T$, and the aggregate throughput $T(\alpha)$ can be numerically solved for any $\alpha \geq 0$. The result is shown in Figure 6.3. It suggests that the conjecture is true for all $\alpha > 0$ for this network. Corollary 6.2 below implies that, indeed, it is.

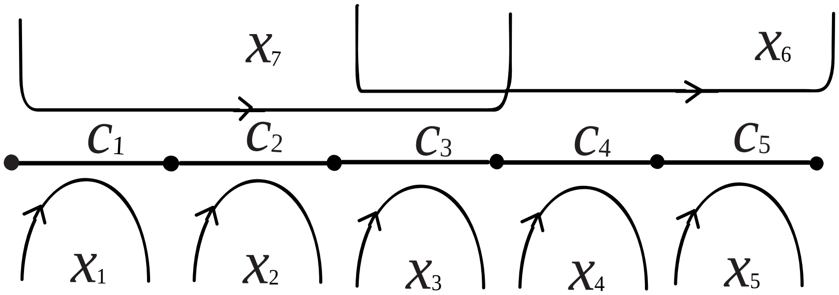


Figure 6.2: Linear network with two long flows.

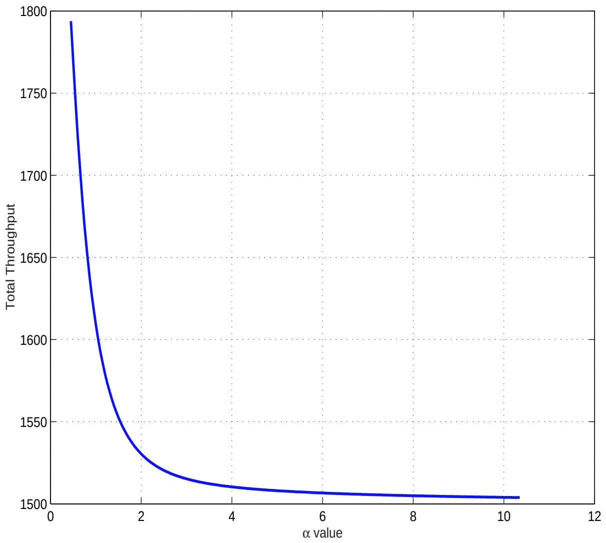


Figure 6.3: Fairness-efficiency tradeoff.

6.4.3 Necessary and sufficient conditions

We now investigate the conjecture in general networks. The aggregate throughput T is a function of source rate $x(\alpha)$

$$T(x(\alpha)) \;=\; \mathbf{1}^T x(\alpha), \tag{6.11}$$

where $\mathbf{1} = (1, \ldots, 1)^T$. From Corollary 6.1, we have

$$\frac{\partial T}{\partial \alpha} \;=\; \mathbf{1}^T Z(Z^T DZ)^{-1} Z^T b. \tag{6.12}$$

When the utility function $U(x, \alpha)$ is defined as in (6.10), the matrix D and vector b defined in (6.6) and (6.7) take the forms

$$D = \alpha \text{ diag}(x_1^{-\alpha-1}, \ldots, x_N^{-\alpha-1}), \qquad b = (x_1^{-\alpha}\log x_1, \ldots, x_N^{-\alpha}\log x_N)^T,$$

where $x = x(\alpha) = x(\alpha, c)$ are the optimal rates. Let $\mu = \mu(\alpha, c)$, $\beta = \beta(\alpha, c)$ and $A = A(\alpha, c)$ be defined by

$$\mu_i \;:=\; z_i^T b, \quad \beta_i \;:=\; -\mathbf{1}^T z_i, \quad \text{and} \;\; A \;:=\; Z^T DZ, \tag{6.13}$$

where z_i are the ith columns of Z. Note that A is positive definite and hence invertible. Let $\bar{A}_i(\alpha, c)$ denote the matrix obtained from replacing the ith row of A with row vector $\beta^T = (\beta_1, \beta_2 \ldots \beta_M)$. From the above definitions and (6.12) we have

$$\frac{\partial T}{\partial \alpha} \quad = \quad -\beta^T A^{-1} \mu. \tag{6.14}$$

Our first main result is a necessary and sufficient condition for the conjecture to hold. Note that the condition is a function of α even though this is not explicit in the notation.

Theorem 6.2. *For any* $\alpha > 0$

$$\frac{\partial T}{\partial \alpha} \leq 0 \quad \textit{if and only if} \quad \sum_{i=1}^{M} \mu_i \, \det \bar{A}_i \geq 0.$$

Proof: The key observation is the following expression for the row vector

$$\beta^T A^{-1} = \frac{1}{\det A} \left(\det \bar{A}_1, \det \bar{A}_2, \ldots, \det \bar{A}_n \right), \tag{6.15}$$

which follows from the following formula for matrix inverse [62]

$$A^{-1} = \frac{1}{\det A} A^*,$$

where A^* is the adjoint matrix of A. Combining (6.14) and (6.15), we have

$$\frac{\partial T}{\partial \alpha} \quad = \quad -\frac{1}{\det A} \sum_{i=1}^{M} \mu_i \det \bar{A}_i.$$

□

Theorem 6.2 characterizes exactly the set of networks (R, c) in which Conjecture 6.1 is true. Though difficult to understand intuitively, this characterization leads directly to two sufficient conditions that explain all the examples in Section 6.4.2. The first condition implies that the conjecture is true when every link has a single-link flow and there is only one long flow. This condition is satisfied by Examples 1 and 2. The second condition implies that the conjecture is true when there are two long flows but both pass through the same number of links. This condition is satisfied by Example 3. The corollary implies that, while the diversity of capacities c_l in Examples 2 and 3 makes the optimization problem (6.2) hard to solve and the previous analysis methods complicated, they are not relevant at all to the truth of the conjecture for these examples.

Corollary 6.2. *Suppose every link has a single-link flow.*

1. *If $dim(Z) = 1$, then $\partial T/\partial\alpha \le 0$ for all $\alpha > 0$.*
2. *If $dim(Z) = 2$ and the only two long flows pass through the same number of links, then $\partial T/\partial\alpha \le 0$ for all $\alpha > 0$.*

To prove the corollary, we now specialize to a particular basis Z of the null space of the routing matrix R, making use of the fact that every link has a single-link flow. Rearrange the column of routing matrix R to express R as

$$R = \begin{bmatrix} I_L & R_1 \end{bmatrix},$$

where I_L is the $L \times L$ identity matrix and R_1 is a $L \times M$ matrix, $N = L + M$. We choose a set of basis for the null space of R such that matrix Z can be expressed as

$$Z = \begin{bmatrix} -R_1 \\ I_M \end{bmatrix}.$$

Clearly rank(Z) = dim$(Z) = M$.

Lemma 6.2. *Suppose every link has a single-link flow. For Z in the form of (6.16), we have*

1. *$\mu_m \ge 0$ for $m = 1, \ldots, M$.*
2. *$a_{mm} \ge a_{mn}$ for all $m, n = 1, \ldots, M$.*

Proof: The proof is a series calculations based on the Karush-Kuhn-Tucker conditions. See [144, 146] for the detailed proof. □

We are now ready to prove Corollary 6.2 with the above lemma.

Proof of Corollary 6.2: 1) In this case, $M = 1$ and $Z \in \Re^{N\times 1}$ is a column vector. There are L single-link flows, one at each of the L links, and exactly one other flow that can traverse one or more links. This means $\sum_{j=1}^{L} -z_{1j} \geq 1$ since the long flow at least transverses one link. Hence

$$\beta_1 \;\; = \;\; -\mathbf{1}^T z_1 = \sum_{j=1}^{L} -z_{1j} - 1 \; \geq \; 0.$$

From Lemma 6.2, we know that $\mu_1 > 0$. From Theorem 6.2 we have

$$\frac{\partial T}{\partial \alpha} \;\; = \;\; -\frac{\mu_1 \beta_1}{\det A} \; \leq \; 0$$

since matrix A is positive definite.

2) In addition to the L single-link flows, there are two flows that traverse one or more links. Since they traverse the same number of links, we have

$$\beta_1 = \beta_2 = -\mathbf{1}^T z_1 \;\; \geq \;\; 0, \tag{6.16}$$

as in the first assertion. We also have

$$\mu_1 \det \bar{A}_1 + \mu_2 \det \bar{A}_2 = \beta_1 \left[\mu_1 (a_{22} - a_{21}) + \mu_2 (a_{11} - a_{12}) \right].$$

Lemma 6.2 and (6.16) then imply that the above quantity is nonnegative. Hence,

$$\frac{\partial T}{\partial \alpha} \;\; = \;\; -\frac{\mu_1 \det \bar{A}_1 + \mu_2 \det \bar{A}_2}{\det A} \; \leq \; 0.$$

□

6.4.4 Counter-example

The condition in the second part of Corollary 6.2 that both long flows pass through the same number of links is important. When it fails, there are networks where the *opposite* of the conjecture is true!

Theorem 6.3. *When $dim(Z) \geq 2$, for any $\alpha_0 > 0$, there exists a network such that*

$$\frac{\partial T}{\partial \alpha} > 0 \qquad \textit{for all } \alpha > \alpha_0$$

Proof: See the detailed proof in [144, 146] . □

Since in reality there are many more flows than bottleneck links and therefore dim(Z) is typically large, it is conceivable that Conjecture 6.1 is wrong more often than right in practice.

Example 4: Counter-example

Consider the linear network with $L = 5$ links and $N = 7$ sources, shown in Figure 6.4. The

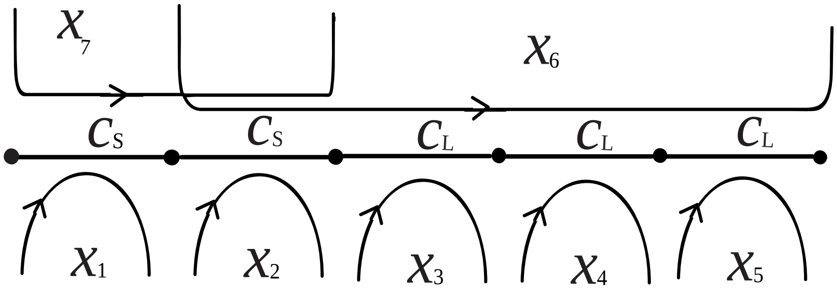


Figure 6.4: Network for counter-example in Theorem 6.3.

null space of R has a dimension dim$(Z) = N - L = 2$. There are five one-link flows with rates $x_1, \ldots, x_5$ and two long flows with rates x_6, x_7. Links 1 and 2 have a small capacity c_S, and links 3, 4, and 5 have a large capacity c_L. We solve the utility maximization (6.5) numerically to compute $T(\alpha)$ for $\alpha \in [0.5, 10]$.

The aggregate throughput $T(\alpha)$ is plotted in Figure 6.5 as a function of α, for $c_S = 10$ and $c_L = 1,000$. The minimal throughput is achieved around $\alpha_0 = 0.95$ and will be achieved around $\alpha_0 = 0.75$ if we change c_L to be $5,000$. $T(\alpha)$ is strictly increasing beyond α_0. In particular,

$$T(\infty) \ > \ T(2) \ > \ T(1).$$

The example is surprising at first because the conventional wisdom in networking is that increasing α favors long flows that take up more resources, leading to a drop in aggregate throughput. This is not exactly right. Recall that the price p_l at a link is a precise measure of congestion at that link. A more precise intuition is that increasing α favors "expensive" flows, flows that have the largest sum of link prices in their paths. In Example 4, the link capacity c_S is small and c_L is large, so that prices are high at links 1 and 2, and low at links 3, 4, and 5. Even though x_6 traverses more links, it has a lower aggregate price over its path than x_7. Hence, when α increases, x_7 increases, leading to a reduction in x_6 (because of sharing at link 2). This reduction allows increases in flows x_3, x_4,and x_5 , so that the net change in aggregate throughput $T(\alpha)$ is positive. Hence the counter-example relies on the design that the longest flow is not the most expensive.

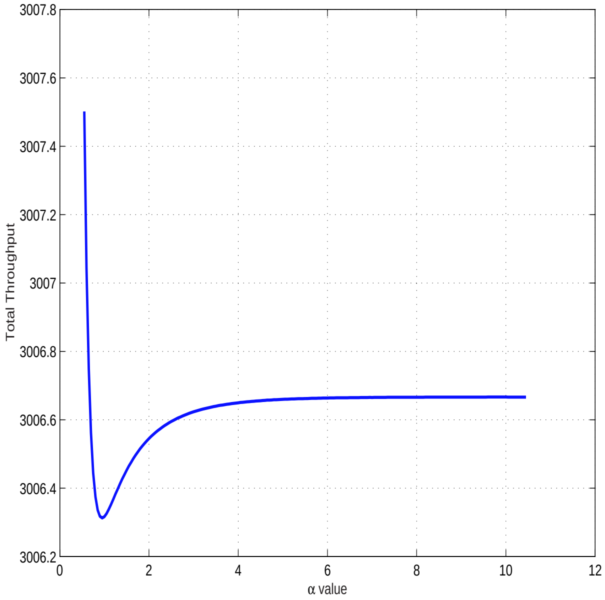


Figure 6.5: Throughput versus efficiency α in the counter-example.

Indeed, one can prove that for the network in Figure 6.4, $\partial x_7/\partial\alpha > 0$ and $\partial x_6/\partial\alpha < 0$

for all $\alpha > \alpha_0$, as illustrated in Figure 6.6. In this example, the decrease in x_6 allows increases in just three one-link flows, and yet this is enough to produce a net increase in the aggregate throughput. Our example actually is compact in that our proof shows that x_6 has to pass through at least three links (link 3,4,5) to make $\partial T/\partial\alpha > 0$.

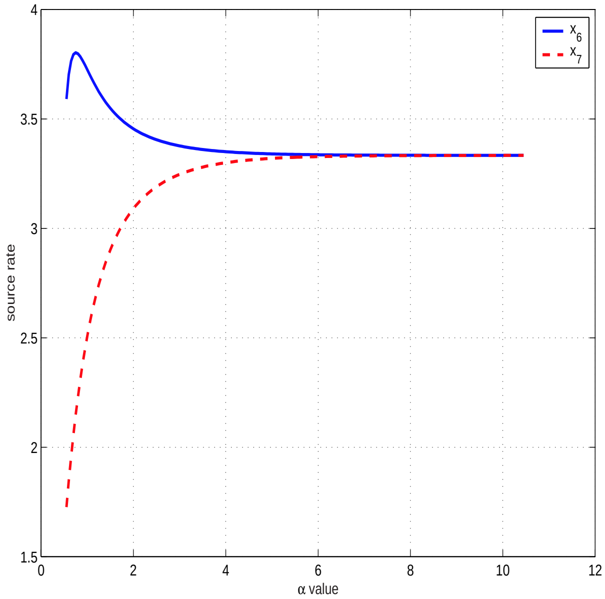


Figure 6.6: Source rates versus α in the counter-example.

One may notice that the amount of increment in Figure 6.5 is quite small. In fact, an easy and loose upper-bound for the increment of aggregate throughput is $c_S/2$. Currently, we don't know whether this small variation is true only for this example or for general networks (R, c).

6.5 Does increasing capacity always raise throughput?

We have seen how fairness, as measured by α, can affect efficiency, as measured by T, in unexpected ways due to interaction among sources in general networks. In this section, we study how increasing capacity c affects the aggregate throughput T. The results here can be useful in deciding in which links resources should be invested to maximize aggregate throughput.

Let $\delta \in R^L$ be the vector that represents the increases in link capacities in the entire network. For instance, when $\delta = \epsilon e_l$, where $\epsilon > 0$ is a small scalar, and e_l is a L-vector that has all its entries 0 except the l-th entry which is 1, then only link l increases its capacity from c_l to $c_l + \epsilon$. When $\delta = \epsilon\mathbf{1}$, then all links increase their capacities by ϵ unit. When $\delta = \epsilon c$, then all links increase their capacities by amounts proportional to their current capacities.

The change in aggregate throughput per unit of an infinitesimal change δ in capacities is measured by the directional derivative $\mathbf{D}T$ of T in direction δ, defined as

$$\mathbf{D}T(\alpha, \delta) = \mathbf{D}T(\alpha, \delta; c) \quad := \quad \lim_{\epsilon \longrightarrow 0} \frac{T(\alpha, c) - T(\alpha, c + \epsilon\delta)}{\epsilon}.$$

From (6.11), we have

$$\mathbf{D}T(\alpha, \delta) \quad = \quad \mathbf{1}^T \frac{\partial x}{\partial c} \delta,$$

where $\partial x / \partial c$ is evaluated at the optimal rate $x(\alpha, c)$. We will take δ to denote the *direction* of increase in capacity, with the understanding that $\epsilon\, \mathbf{D}T(\alpha, \delta)$ provides an estimate of change in aggregate throughput when c is changed to $c + \epsilon\delta$. Our results should be interpreted in the context of small perturbations that do not change the active constraint set in (6.2).

Define $B = RD^{-1}R^T$, $\eta = \mathbf{1}^T D^{-1} R^T$, and $\bar{B}_i$ is the matrix obtained by replacing ith row of B by η. A similar argument to the proof of Theorem 6.2 yields the following:

Theorem 6.4. *For any* $\delta, \alpha > 0$

$$\boldsymbol{D}T(\alpha, \delta) \geq 0 \quad \textit{if and only if} \quad \sum_{i=1}^{L} \delta_i \det \bar{B}_i \geq 0.$$

Theorem 6.4 characterizes exactly the set of all networks (R, c), and directions δ, in which aggregate throughput will increase, for *all* fairness $\alpha > 0$. An easy consequence is the following:

Corollary 6.3. *If* R *has only two rows, then* $\boldsymbol{D}T(\alpha, \delta) \geq 0$ *for any* $\alpha > 0$ *and any* $\delta \geq 0$.

Proof: Let B_{ij} denote the (i, j) element of B. A similar argument to the proof of Lemma

6.2 shows that

$$\eta_i = B_{ii}, \quad \text{and} \qquad B_{ii} \geq B_{ij} = B_{ji} \quad \text{for} \quad i, j = 1, 2.$$

Then

$$\det(\bar{B}_1) \;=\; \eta_1 B_{22} - \eta_2 B_{21} = b_{22}(B_{11} - B_{21}) \geq 0.$$

Similarly we have $\det(\bar{B}_2) \geq 0$. From Theorem 6.4, we have $\mathbf{D}T(\alpha, \delta) \geq 0$. □

Corollary 6.3 says that increasing link capacity always raises aggregate throughput, provided there are only two bottleneck links. Intuitively, one might expect this to hold more generally. This is however not the case. We provide three interesting examples, with different instantiations of direction δ, as an illustration.

The first result says that not only can the aggregate throughput be reduced when some link increases its capacity; paradoxically, it can also be reduced when *all* links increase their capacities by the same amount. This is true for almost all fairness α.

Theorem 6.5. *Given any $\alpha_0 > 0$,*

1. *there exists a network (R, c) such that for all $\alpha > \alpha_0$, $\boldsymbol{D}T(\alpha, e_l) < 0$ for some link l.*
2. *there exists a network (R, c) such that for all $\alpha > \alpha_0$, $\boldsymbol{D}T(\alpha, \boldsymbol{1}) < 0$.*

Proof: The proof is by construction. For the first claim, consider the network in Figure 6.7.

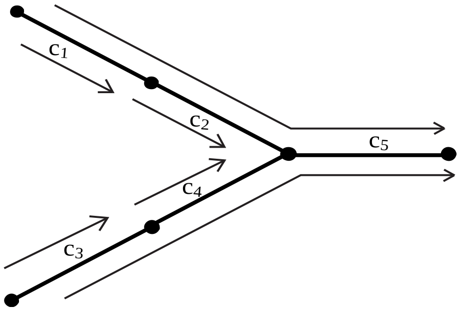


Figure 6.7: Counter-example for Theorem 6.5(1).

There is a single-link flow x_l at each link l, for $l = 1, \ldots 4$. The flow x_5 transverses links

$1, 2$, and 5, and flow x_6 transverses links $3, 4$, and 5 respectively. The capacities of the links are $c_1 = c_2 = c_3 = c_4 = c_L$ and $c_5 = c_S$. We increase only link 5's capacity by 1, which corresponds to $\delta = e_5$. For any fixed $\alpha_0 > 0$, we can choose c_L/c_S large enough, such that for any $\alpha > \alpha_0$ all links are fully utilized. Calculating the change in aggregate throughput using Mathematica gives

$$\mathbf{D}T(\alpha, e_5) \quad = \quad \mathbf{1}^T \frac{\partial x}{\partial c} e_5 = \mathbf{1}^T D^{-1} R^T \left(R D^{-1} R^T\right)^{-1} e_5 = -1.$$

For the second claim, consider the network shown in Figure 6.8. There is a single-link flow x_l at each link l, for $l = 1, ..., 10$. The link capacities are $c_l = c_S$ for $l = 1, 2, 3$ and $c_l = c_L$ for $l = 4, \ldots, 10$. We skip the detailed proof of $\mathbf{D}T(\alpha, \mathbf{1}) < 0$, which can be found in [144, 146]. □

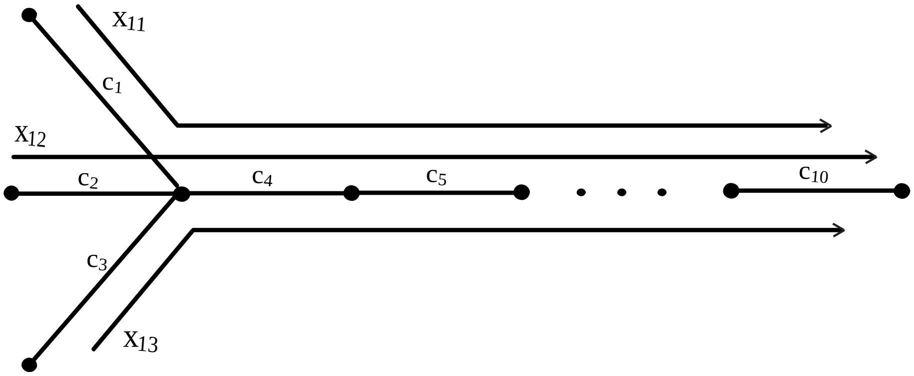


Figure 6.8: Counter-example for Theorem 6.5(2).

Example 5: $\mathbf{D}T(\infty, \mathbf{1}) < 0$ for some c in Figure 6.8

To illustrate, we calculate the change in aggregate throughput for the network in Figure 6.8 under max–min policy $\alpha = \infty$. Let the link capacities be $c_S = 2$ and $c_L = 10$. The source can be easily calculated under max-min fairness, and it is easy to check that the aggregate throughput is 55. When all capacities are increased by 2ϵ with $0 \leq \epsilon \leq 1$, we can check that the total throughput $T(\epsilon)$ changes into $55 - \epsilon$, i.e., $T(\epsilon)$ is a decreasing function of ϵ. Indeed $\mathbf{D}T(\infty, \mathbf{1}) = -1/2 < 0$ in this situation.

If link capacities are increased proportionally, i.e., if c is increased to $(1+\epsilon)c$, then aggregate throughput will always rise. Note that even though increasing all link capacities proportionally may be interpreted as changing the unit of capacity, it does *not* imply that general utility functions increase proportionally in optimal flow vectors. It does however imply this for the class of utility functions defined in (6.10).

Theorem 6.6. *For any network (R,c) and for all $\alpha > 0$, **D**$T(\alpha, c) > 0$.*

Proof: The necessary and sufficient condition for any $x \geq 0$ and $p \geq 0$ to be primal and dual optimal are

$$Rx = c, \quad \text{and} \quad R^T p = \frac{\partial V}{\partial x} = (x_1^{-\alpha}, \ldots, x_N^{-\alpha})^T.$$

Suppose x and p are optimal with link capacities c. When c is increased to $(1+\epsilon)c$ for $\epsilon > 0$, we claim that $x(1+\epsilon)$ and $(1+\epsilon)^{-\alpha} p$ are the new optimal rate and price vectors, respectively. We can check that these vectors satisfy the optimality condition for capacity $(1+\epsilon)c$

$$\begin{aligned} Rx(1+\epsilon) &= c(1+\epsilon), \\ R^T p(1+\epsilon)^{-\alpha} &= (1+\epsilon)^{-\alpha}(x_1^{-\alpha}, \ldots, x_N^{-\alpha})^T = \frac{\partial V}{\partial x}. \end{aligned}$$

Therefore, the aggregate throughput is increased from the original value T to $(1+\epsilon)T$. Hence, we have **D**$T(\alpha, c) = T > 0$. □

6.6 Conclusion

A bandwidth allocation policy can be defined in terms of utility functions parameterized by some protocol parameter α. We have studied how throughputs and prices change as link capacities or α changes. We then focus on a specific class of utility functions where α can be interpreted as a quantitative measure of fairness. We say an allocation is fair if α is large and efficient if the aggregate throughput is large. We use this model to investigate whether a fairer allocation is always more inefficient and whether increasing link capacities always raises throughput. We characterize exactly the set of all networks (R, c) in which the answers are “yes.” Though these characterizations are difficult to understand intuitively, they have led to simple corollaries that explain all the examples we found in the literature and to the discovery of the first counter examples.

There are a number of ways this preliminary work can be extended. First, we have focused of how throughputs x change in response to changes in α and c, which is only half of Theorem 6.1. The application of the other half of Theorem 6.1 on how prices change has not been exploited. Second, the necessary and sufficient conditions for the conjectures are hard to understand intuitively and check for large networks. It is not clear whether this condition is likely to hold or fail in practice. It would be useful to derive equivalent characterizations that are more intuitive or more general corollaries than reported here. Finally, we have assumed that every source has the same utility function. It would be interesting to see how the fairness definition and tradeoff results should generalize when sources have the same class of utility functions but with different α_i parameters, or have different utility functions.

Chapter 7

Other Related Projects

7.1 Network equilibrium with heterogeneous protocols

7.1.1 Introduction

As we have seen in the previous chapters, congestion control protocols can be modelled as distributed algorithms to maximize the aggregate utility, e.g., [80, 97, 116, 164, 88, 96]. However, these studies assume that all sources are homogeneous, that is, even though they may control their rates using different algorithms, they all adapt to the same type of congestion signals, e.g., loss probabilities in TCP Reno and queueing delay in FAST TCP [69]. When sources with *heterogeneous* protocols that react to different congestion signals share the same network, the current duality framework is no longer applicable. With new congestion control algorithms proposed for large bandwidth-delay product networks and usage of congestion signals other than packet losses (including explicit feedbacks with ECN), we need a rigorous framework to understand the behavior of large-scale networks with heterogeneous protocols.

A congestion control protocol generally takes the form

$$\dot{p}_l = g_l\left(\sum_{j:l\in L(j)} x_j(t), p_l(t)\right), \qquad \dot{x}_j = f_j\left(x_j(t), \sum_{l\in L(j)} m_l^j(p_l(t))\right). \tag{7.1}$$

As we have shown in Chapter 2, here $g_l(\cdot)$ models a queue management algorithm, and $f_j(\cdot)$ models a TCP algorithm. The effective prices $m_l^j(p_l(t))$ are functions of the link prices $p_l(t)$, which in general can vary depending on the links and source.

When all algorithms use the same pricing signal (i.e., homogeneous protocols with $m_l^j = m_l$ for all j), the equilibrium of (7.1) turns out to be very simple. At the equilibrium, the source rates x_i solve a utility maximization problem, and the link congestion measure p_l serves as a Lagrange dual. When heterogeneous algorithms that use different pricing signals share the same network, i.e., m_l^j are different for different sources j, the situation is much more complicated. For instance, when TCP Reno and FAST TCP share the same network, neither loss probability nor queueing delay can serve as the Lagrange multiplier at the link, and (7.1) can no longer be interpreted as solving the standard utility maximization problem. Basic questions, such as the existence and uniqueness of equilibrium, and its local and global stability, need to be re-examined.

7.1.2 Model

A network consists of a set of L links with finite capacities c_l. There are J different protocols indexed by superscript j, and there are N^j sources using protocol j indexed by (j, i). The total number of sources is $N := \sum_j N^j$. The $L \times N^j$ routing matrix R^j for type j sources is defined by $R_{li}^j = 1$ if source (j, i) uses link l, and 0 otherwise. The overall routing matrix is denoted by $R = \left[\begin{array}{cccc} R^1 & R^2 & \cdots & R^J \end{array} \right]$.

Every link l has a price p_l. A type j source reacts to the "effective price" $m_l^j(p_l)$ in its path. By specifying function m_l^j, we can let the link feed back different congestion signals to sources using different protocols. The end-to-end prices for source (j, i) is $q_i^j = \sum_l R_{li}^j m_l^j(p_l)$. Let $q^j = (q_i^j, i = 1, \ldots, N^j)$, $q = (q^j, j = 1 \ldots, J)$, $m^j(p) = (m_l^j(p_l), l = 1, \ldots L)$ and $m(p) = (m^j(p_l), j = 1, \ldots J)$ be vector forms. Then $q^j = \left(R^j\right)^T m^j(p)$ and $q = R^T m(p)$.

Let x^j be a vector with the rate x_i^j of source (j, i) as its ith entry, and let x be the vector of x^j. We suppose that source (j, i) has a utility function $U_i^j(x_i^j)$ that is strictly concave and increasing.

A network is in equilibrium when each source (j, i) maximizes its net benefit and the demand for and supply of bandwidth at each bottleneck link are balanced. Formally, a network equilibrium is defined as follows.

Given prices p, the end-to-end price vector q is formulated as $q = R^T m(p)$. The source rate x_i^j is uniquely determined by q_i^j, and it uniquely solves $\max_{z \geq 0} \; U_i^j(z) - zq_i^j$. Therefore, the source rates vector x is a function of link prices p, denoted as $x(p)$. Denote $y(p)$ as the aggregate source rates at links, then $y(p) = Rx(p)$.

In equilibrium, the aggregate rate at each link is no more than the link capacity, and they are equal if the link price is strictly positive. Formally, we call p an *equilibrium* if it satisfies

$$P(y(p) - c) = 0, \;\; y(p) \leq c, \;\; p \geq 0 \tag{7.2}$$

where $P :=$ diag(p_l) is a diagonal matrix. We will study the existence and uniqueness properties of network equilibrium specified by the above equations.

7.1.3 Existence of equilibrium

We prove the existence of equilibrium under the following assumptions.

A1: Utility functions U_i^j are strictly concave, increasing, and twice differentiable. Price mapping functions m_l^j are differentiable and strictly increasing with $m_l^j(0) = 0$.

A2: For any $\epsilon > 0$, there exists a number $p_{\max}$ such that if $p_l > p_{\max}$ for link l, then $x_i^j(p) < \epsilon$ for all (j, i) with $R_{li}^j = 1$.

These assumptions are mild. Concavity and monotonicity of utility functions are often assumed in network pricing for elastic traffic. The assumption on m_l^j preserves the relative order of prices and maps zero price to zero effective price. Assumption A2 says that when p_l is high enough, then every source going through link l has a rate less than ϵ.

Theorem 7.1. *Suppose A1 and A2 hold. There exists an equilibrium price p^* for any network (c, m, R, U).*

The mathematical tool used to prove this theorem is the Nash theorem in game theory [121, 10], which is an application of Kakutani's generalized fixed point theorem. The detailed proof can be found in [147].

7.1.4 Examples of multiple equilibria

Theorem 7.1 guarantees the existence of network equilibrium; however the equilibrium may not be unique. We show two examples of multiple equilibria.

In a single-protocol network, if the routing matrix R has full row rank, then there is a unique active constraint set. In contrast, the active constraint set in a multi-protocol network can be non-unique even if R has full row rank as shown in Example 1. Clearly, the equilibrium prices associated with different active constraint sets are different.

Example 1: Multiple equilibria with different active constraint sets

Consider a symmetric network in Figure 7.1 with three flows. The link 1 and link 3 have identical parameters, and flows $(1, 1)$ and $(1, 2)$ have identical utility function. We show that [142], under certain conditions, the network has two equilibria with different congested links. We also carry out experiments with TCP Reno, which reacts to loss probability, and TCP Vegas/FAST, which reacts to delay, and we set the experiment parameters such that the conditions are satisfied. The prices (queues) at link 1 and link 2 are shown in Figure 7.2. This result unambiguously exhibits that there are two equilibria with different active constraint sets. The queue flip is produced when the network operates at different equilibria.

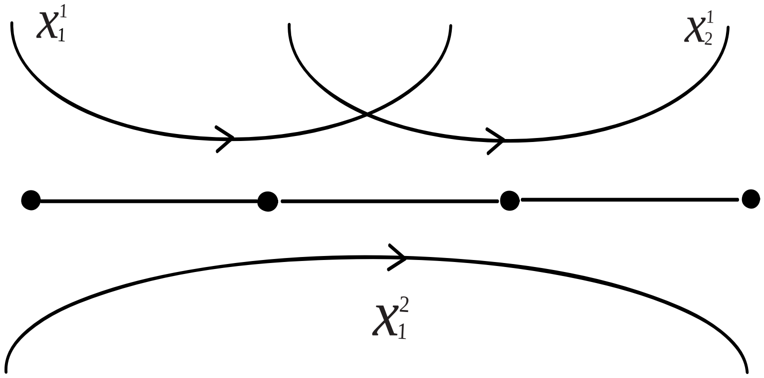


Figure 7.1: Example 2: two active constraint sets.

Example 2: Multiple equilibria with a unique active constraint set

When the active constraint set is unique, it is still possible to have multiple equilibria, and even uncountable many of them. We show that such an example with $J = 3$. The

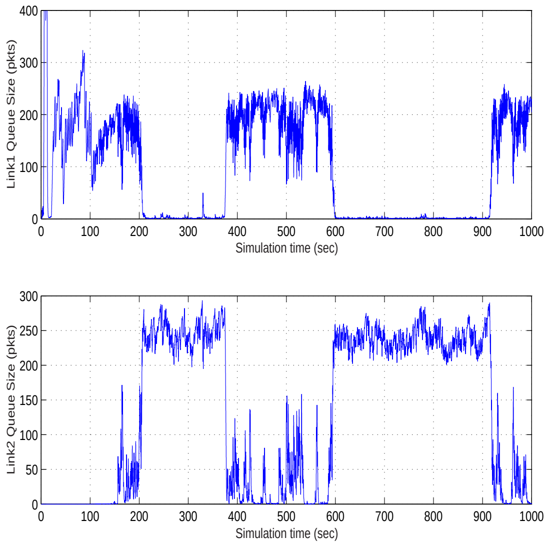


Figure 7.2: Shifts between the two equilibria with different active constraint sets.

network is shown in Figure 7.3 with three unit-capacity links, $c_l = 1$. All the sources use

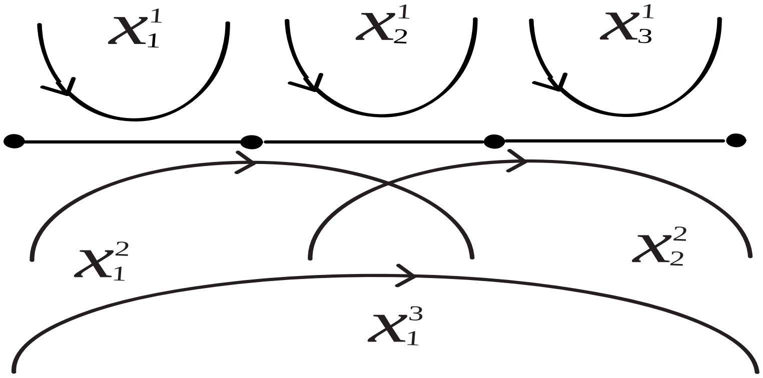


Figure 7.3: Example 2: uncountably many equilibria.

the same utility function $U_i^j(x_i^j) = -\left(1 - x_i^j\right)^2/2$, and the price mapping function is linear $m^j(p) = K^j p$, where K^j are $L \times L$ diagonal matrices with $K^1 = I$, $K^2 = \text{diag}(5, 1, 5)$, and $K^3 = \text{diag}(1, 3, 1)$.

It can be shown that the equilibrium price p satisfies

$$\sum_j R^j (R^j)^T K^j p \quad = \quad \sum_j R^j \mathbf{1} - c$$

which is a linear equation in p. It has a unique solution if the determinant $\det\left(\sum_j R^j (R^j)^T K^j\right)$

is nonzero, but has no or multiple solutions otherwise. By choosing appropriate parameters (as shown above), we can make this determinant zero. It is easy to check that all of the following are equilibrium prices for this network

$$p_1^1 = p_3^1 = 1/8 + \epsilon, \quad \text{and} \quad p_2^1 = 1/4 - 2\epsilon \quad \text{where} \quad \epsilon \in [0, 1/24].$$

The corresponding source rates can also be derived, and all capacity constraints are tight with these rates, yet there are uncountably many equilibria.

Examples 1 and 2 show that global uniqueness is generally not guaranteed in a multi-protocol network. We will show, however, that local uniqueness is basically a generic property of the equilibrium set.

7.1.5 Local uniqueness of equilibrium

We denote E as the equilibrium set where p is in E if and only if it satisfies (7.2). Fix an equilibrium price $p^* \in E$. Let the *active constraint set* $\hat{L} = \hat{L}(p^*) \subseteq L$ be the set of links at which $p_l^* > 0$. Consider the reduced system that consists only of links in $\hat{L}$, and denote all variables in the reduced system by $\hat{c}$, $\hat{p}$, $\hat{y}$, etc. Then, since $y_l(p) = c_l$ for every $l \in \hat{L}$, we have $\hat{y}(\hat{p}) = \hat{c}$. Let the Jacobian for the reduced system be $\hat{\mathbf{J}}(\hat{p}) = \partial \hat{y}(\hat{p})/\partial \hat{p}$, and

$$\hat{\mathbf{J}}(\hat{p}) \;\; = \;\; \sum_j \hat{R}^j \frac{\partial x^j(\hat{p})}{\partial \hat{q}^j} \left(\hat{R}^j\right)^T \frac{\partial \hat{m}^j(\hat{p})}{\partial \hat{p}}. \tag{7.3}$$

Since the equilibrium price $\hat{p}^*$ for the links in $\hat{L}$ is a solution of $\hat{y}(\hat{p}) = \hat{c}$, by the inverse function theorem, the equilibrium price $\hat{p}^*$, is *locally unique* if the Jacobian matrix $\hat{\mathbf{J}}(\hat{p}^*) = \partial \hat{y}/\partial \hat{p}$ is nonsingular at $\hat{p}^*$. We call a network *regular* if all its equilibrium prices are locally unique. The following theorem shows that almost all networks are regular and that regular networks have finitely many equilibrium prices. This implies that the uncountablly many equilibria shown in Example2 almost never happens in real networks.

Theorem 7.2. *Suppose assumptions A1 and A2 hold. Given any price mapping functions* m*, any routing matrix* R *and utility functions* U*,*

1. *the set of link capacities c for which not all equilibrium prices are locally unique has Lebesgue measure zero in $\Re^L_+$.*

2. *the number of equilibria for a regular network (c, m, R, U) is finite.*

We now narrow our attention to networks that satisfy an additional assumption:

A3: Every link l has a single-link flow. (j, i) with $\left(U_i^j\right)'(c_l) > 0$.

Assumption A3 says that when the price of link l is small enough, the aggregate rate through it will exceed its capacity. It implies that the active constraint set is unique and contains every link.

Since all the equilibria of a regular network have nonsingular Jacobian matrices, we can define the *index* $I(p)$ of $p \in E$ as

$$I(p) = \begin{cases} 1 & \text{if } \det(\mathbf{J}(p)) > 0 \\ -1 & \text{if } \det(\mathbf{J}(p)) < 0 \end{cases} .$$

Theorem 7.3. *Suppose assumptions A1–A3 hold. Given any regular network, we have*

$$\sum_{p \in E} I(p) = (-1)^L$$

where L is the number of links.

The proof of this theorem is based on the Poincare-Hopf index Theorem [149, 113]. First, we construct a vector field formed by a continuous-time gradient project algorithm [97] with multiple protocols. Clearly, p^* is an equilibrium point of this vector field if and only if it is a network equilibrium. Under the assumption A3, there will be a contraction region in this vector field, and all the equilibria are in this region. The Jacobian matrix of the vector field equilibrium point is the same as 7.3 if uniform stepsize is used. Since the network is regular, every equilibrium has an index. Using the Poincare-Hopf index theorem gives us the result in Theorem 7.3. An obvious consequence of this theorem is:

Corollary 7.1. *Suppose assumptions A1–A3 hold. A regular network has an odd number of equilibria.*

Corollary 7.1 also implies the existence of an equilibrium, although we show this in a more general setting in Section 7.1.3. Next we will present another example to illustrate Theorem 7.3 and Corollary 7.1.

Example 3: Illustration of Theorem 7.3 and Corollary 7.1

Recall that in Example 1, there are uncountably many equilibria. The components x_1^1 and q_1^1 of these equilibrium points are shown by the (red) solid line in Figure 7.4. We can change the utility function of every source into the following form

$$U_i^j(x_i^j, \alpha_i^j) = \begin{cases} \beta_i^j (x_i^j)^{1-\alpha_i^j}/(1-\alpha_i^j) & \text{if } \alpha_i^j \neq 1 \\ \beta_i^j \log x_i^j & \text{if } \alpha_i^j = 1 \end{cases},$$

where α_i^j and β_i^j are parameters. We pick two points (the two black dots), and choose appropriate parameters α_i^j and β_i^j for every source, such that these two points will be isolated equilibria.

After this perturbation, we can check whether the two designed equilibria are locally unique, and the network is regular. Corollary 7.1 predicts an odd number of equilibria. We indeed can find another equilibrium, and three of them in total 7.4.

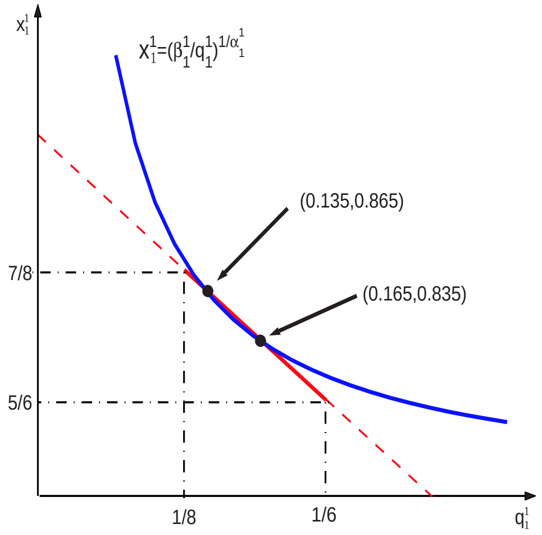


Figure 7.4: Example 3: construction of multiple isolated equilibria.

We further check the local stability of these three equilibria under the gradient algorithm. It turns out that one of them is not stable and has index 1, while the other two are stable with index -1. The dynamics of this network under the gradient algorithm can be illustrated by a vector field. We draw the vector field restricted on the plane $p_1 = p_3$, and the phase portrait is shown in Figure 7.5. The (red) dots represent the three equilibria. Note that the equilibrium in the middle is a saddle point, and it is therefore unstable. The (red) arrows give the direction of this vector field. Individual trajectories are plotted with slim (blue) lines.

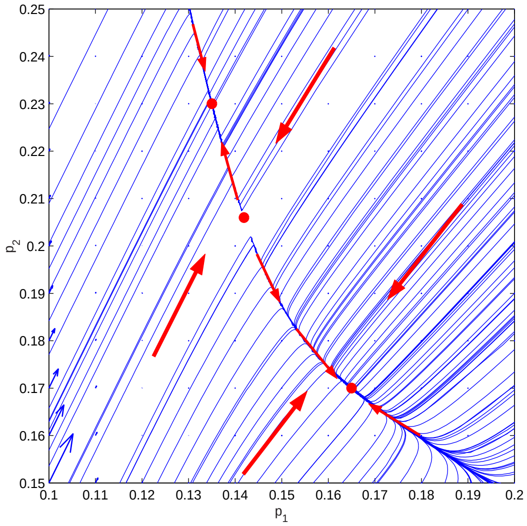


Figure 7.5: Example 3: vector field of (p_1, p_2).

7.1.6 Global uniqueness of equilibrium

The exact condition under which network equilibrium is globally unique is generally hard to prove. We provide several special cases for global uniqueness.

Theorem 7.4. *Suppose assumptions A1–A3 hold. If all equilibria have index* $(-1)^L$*, then* E *contains exactly one point. In particular, if all equilibria are locally stable, then* E *contains exactly one point.*

The first claim of the theorem directly follows from Theorem 7.3. It can be checked that a local stable equilibrium also has an index of $(-1)^L$, and the second claim also holds.

This result relates the local stability of an algorithm to the uniqueness property of a network. Local stability can be checked in several ways, and it can be used to prove global uniqueness. We will concentrate on several special cases in the rest of this section.

Theorem 7.5. *Suppose assumptions A1–A3 hold and R has full row rank. If for all j and l, $m_l^j(p_l) = k^j p_l$ for some scalar $k^j > 0$, then there is a unique network equilibrium .*

Under the assumption of Theorem 7.5, it is easy to show that we have an unusual situation in the theory of heterogeneous protocols where the equilibrium rate vector x solves the following concave maximization problem

$$\max_x \sum_{i,j} k^j U_i^j(x_i^j) \qquad \text{s. t. } Rx \leq \ c.$$

Therefore, such a network always has a globally unique equilibrium when U_i^j are strictly concave. It can also be proved using Theorem 7.4 by showing that every equilibrium is locally stable under the gradient projection algorithm.

Theorem 7.6. *Suppose assumptions A1–A2 hold. The linear network in Figure 7.6 has a unique equilibrium.*

7.6. We can show that every equilibrium in this network is locally stable and that even every source uses different protocols. By Theorem 7.4, the equilibrium is globally unique.

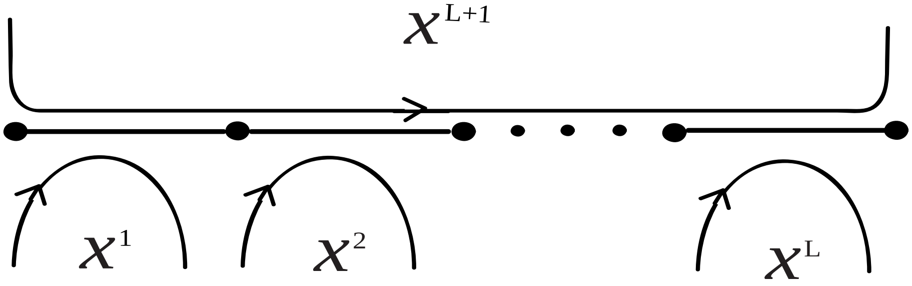


Figure 7.6: Corollary 7.6: linear network.

Theorem 7.4 also implies the global uniqueness of equilibrium for any network in which no flow passes through more than 2 links in the active constraint set, when A1–A3 hold. In this case, the Jacobian matrix is strictly diagonally dominant with negative diagonal entries, and hence its determinant is $(-1)^L$.

Theorem 7.7. *Suppose assumptions A1–A3 hold. A network where all flows using at most two links has a unique equilibrium.*

7.1.7 Conclusion

When sources sharing the same network react to different pricing signals, the current duality model no longer explains the equilibrium of bandwidth allocation. We have introduced a mathematical formulation of network equilibrium for multi-protocol networks and studied several fundamental properties, such as existence, local uniqueness, number of equilibria, and global uniqueness. We prove that equilibria exist and are almost always locally unique. The number of equilibria is almost always finite and must be odd when they are associated with the same active constraint set. We provide four sufficient conditions for global uniqueness.

The utility maximization problem that underlies a single-protocol network implies that the equilibrium source rates exist and are always unique [96]. In the heterogeneous protocol case, we prove that equilibrium still exists, under mild conditions, despite the lack of an underlying concave optimization problem. There can be uncountably many equilibria, and the bottleneck links set can be also be non-unique. However, we prove that almost all networks have finitely many equilibria and that they are necessarily locally unique. Non-uniqueness can arise in two ways. First, the equilibria associated with different sets of bottleneck links are always distinct. Second, the number of equilibria associated with each set of bottleneck links can be more than one, though always odd. Moreover, these equilibria cannot all be locally stable unless the equilibrium is globally unique. We also provide several special cases for global uniqueness of network equilibrium. We also provide numerical examples to illustrate the theorem and equilibrium properties.

7.2 Control unresponsive flow–CHOKe

7.2.1 Introduction

TCP is believed to be largely responsible for preventing congestion collapse while the Internet has undergone dramatic growth in the last decade. Indeed, numerous measurements have consistently shown that more than 90% of traffic on the current Internet still consists of TCP packets, which, fortunately, are congestion controlled. Without a proper incentive structure, however, this state of affairs is fragile and can be disrupted by the growing number of non-rate-adaptive (e.g., UDP-based) applications that can monopolize network bandwidth to the detriment of rate-adaptive applications. This has motivated several active queue management schemes, e.g., [111, 41, 94, 141, 123, 126, 35], that aim at penalizing aggressive flows and ensuring fairness. The scheme, CHOKe, of [126] is particularly interesting in that it does not require any state information and yet can provide a minimum throughput to TCP flows.

The basic idea of CHOKe is explained in the following quotation from [126]:

> "When a packet arrives at a congested router, CHOKe draws a packet at random from the FIFO (first-in-first-out) buffer and compares it with the arriving packet. If they both belong to the same flow, then they are both dropped; else the randomly chosen packet is left intact and the arriving packet is admitted into the buffer with a probability that depends on the level of congestion (this probability is computed exactly as in RED). "

The surprising feature of this simple scheme is that it can bound the bandwidth share of UDP flows regardless of their arrival rate. Extensive simulation results in [126] show that as the arrival rate of UDP packets increases without bound, their bandwidth share peaks and then drops to zero! It seems intriguing that a flow that maintains a much larger number of packets in the queue does not receive a larger share of bandwidth, as in the case of a regular FIFO buffer. We provide an analytical model of CHOKe that explains both this throughput behavior and the spatial characteristics of its leaky buffer. In this section, we will present the model, analysis, and simulations of CHOKe very briefly. See [155, 143, 145] for details.

7.2.2 Model

We focus on a network with a single bottleneck link with capacity c pkts/sec, which is shared by by N identical TCP sources and a UDP flow with a constant sending rate. We study the network's equilibrium behavior.

Equilibrium quantities (rate, dropping probability, etc.) associated with the UDP flow are indexed by 0. Since the TCP sources are identical, we will use index 1 for all TCP flows. The definition of all variables and some of their obvious properties are collected below:

d: common round-trip propagation delay for TCP sources.

τ: common queueing delay, and round-trip delay is $d + \tau$.

b_i: packet backlog from flow i, $i = 0, 1$.

b: total backlog; $b = b_0 + b_1 N$.

r: congestion based dropping probability. In general, $r = g(b, \tau)$ where g is a function of aggregate backlog b and queueing delay τ.

x_i: source rate of flow i. In general, $x_1 = f(p_1, \tau)$ where f is a function of overall loss probability p_1 and queueing delay τ.

h_i: the probability that an incoming packet of flow i is dropped by CHOKe $h_i = b_i/b$.

p_i: overall probability that a packet of flow i is dropped before it gets through.

A packet may be dropped, either on arrival due to CHOKe or congestion (e.g., according to RED) or after it has been admitted into the queue when a future arrival from the same flow triggers a comparison. Every arrival packet from flow i can trigger either 0 packet loss from the buffer, 1 packet loss due to RED, or 2 packet losses due to CHOKe. These events happen with respective probabilities of $(1 - h_i)(1 - r)$, $(1 - h_i)r$, and h_i. Hence, each arrival to the buffer is accompanied by an average packet loss of

$$p_i = 2h_i + (1 - h_i)r + 0 \cdot (1 - h_i)(1 - r) = 2h_i + r - rh_i. \tag{7.4}$$

Consider a packet of flow i that eventually goes through the queue without being dropped. The probability that it is not dropped on arrival is $(1 - r)(1 - h_i)$. Once it enters the queue, it takes τ time to go through it. In this time period, there are on average τx_i packets from flow i that arrive at the queue. The probability that this packet is not chosen for comparison is $(1-1/b)^{\tau x_i}$. Hence, the overall probability that a packet of flow i survives the queue is

$$1 - p_i = (1-r)(1-h_i)\left(1-\frac{1}{b}\right)^{\tau x_i}. \tag{7.5}$$

The rate of the flow i's packets getting through the buffer is $x_i(1-p_i)$. Since the link is fully utilized, the flow throughputs sum to link capacity:

$$x_0(1-p_0) + Nx_1(1-p_1) = c.$$

The model is derived by putting together all the above equations. The only independent variable is UDP rate x_0, and there are ten dependent variables. In summary, the model is described by the following ten equations:

$$p_i = 2h_i + r - rh_i, \quad i = 0, 1 \tag{7.6}$$

$$p_i = 1 - (1-r)(1-h_i)\left(1-\frac{1}{b}\right)^{\tau x_i}, \quad i = 0, 1 \tag{7.7}$$

$$h_i = \frac{b_i}{b}, \quad i = 0, 1 \tag{7.8}$$

$$b = b_0 + Nb_1 \tag{7.9}$$

$$c = x_0(1-p_0) + Nx_1(1-p_1) \tag{7.10}$$

$$x_1 = f(p_1, \tau) \quad \text{(TCP)} \tag{7.11}$$

$$r = g(b, \tau) \quad \text{(e.g. RED)} \tag{7.12}$$

Substituting (f, g) with the analytical model of Reno/RED, this set of nonlinear equations (7.6)–(7.12) can be solved numerically using Matlab. The solution is accurately validated with ns-2 simulations shown in Section 7.2.5. This solution can then be used in the differential equation model described later to solve for spatial properties of the leaky buffer.

7.2.3 Throughput analysis

By making three approximations, we can derive the maximum achievable UDP throughput, and prove that UDP throughput approaches zero when it sends infinitely fast.

First, we approximate the system by one in which the order of congestion based dropping and CHOKe is reversed. Second, we assume that N is large such that a comparison triggered by a *TCP* arriving packet never yields a match. The last assumption is that we can approximate $(1-1/b)^b \simeq e^{-1}$. Under these assumptions, we can eliminate τ in the model (7.6)–(7.12) and get our key equation

$$\frac{1-h_0}{1-2h_0} \quad = \quad \exp\left(\frac{x_0(1-r)(1-h_0)}{c-x_0(1-r)(1-2h_0)}\right). \tag{7.13}$$

Let $\mu_0 = \mu_0(x_0)$ denote the UDP throughput share, $\mu_0 = x_0(1-p_0)/c$, and let $\mu_0^* = \max_{x_0 \geq 0} \mu_0(x_0)$ denote the maximum achievable UDP share. The UDP throughput behavior can be totally captured using equation (7.13), which is independent of TCP and AQM algorithms. We show the bandwidth properties in Theorem 7.8 and visualize it in Figure 7.7 which shows UDP throughput versus sending rates.

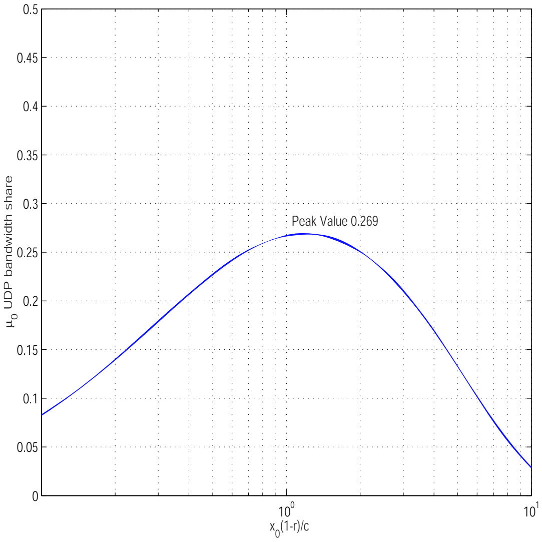


Figure 7.7: Bandwidth share μ_0 v.s. Sending rate $x_0(1-r)/c$.

Theorem 7.8. *The UDP throughput has the following properties:*

1. *The maximum UDP bandwidth share is* $\mu_0^* = (e+1)^{-1} = 0.269$. *It is attained when the UDP input rate after congestion based dropping is* $x_0^*(1-r^*) = c(2e-1)/(e+1) = 1.193c$. *In this case, the CHOKe dropping rate for UDP is* $h_0^* = (e-1)/(2e-1) = 0.387$.

2. *As the UDP sending rate grows without bound, even though UDP packets occupy up to half of the queue, its throughput drops to zero. That is, as $x_0 \to \infty$, $b_0 \to b/2$ but $\mu_0 \to 0$.*

The second result of the theorem can also be proved without using the three approximations. See proofs and more details in [155, 143, 145].

7.2.4 Spatial characteristics

We now derive the spatial characteristics of the leaky buffer under CHOKe that give rise to the macroscopic properties of maximum and asymptotic throughput proved in the previous subsection.

Let $y \in [0, b]$ denote a position in the queue, with $y = 0$ being the tail. Define $v(y)$ as the velocity at which the packet at position y moves toward the head of the queue $v(y) = dy/dt$. For instance, the velocity at the head of the queue equals the link capacity, $v(b) = c$. Let $\rho_i(y)$ be the probability that the packet at position y belongs to flow i, $i = 0, 1$. The bandwidth share μ_i is the probability that the head of the queue is occupied by a packet from flow i, $\mu_i = \rho_i(b)$. We can derive an ordinary differential equation (ODE) model of these two quantities

$$\begin{aligned} v'(y) &= \beta\left(\rho_0(y)x_0 + (1-\rho_0(y))x_1\right), && (7.14)\\ \rho_0'(y) &= \beta(x_0 - x_1)\,\rho_0(y)(1-\rho_0(y))\,\frac{1}{v(y)}, && (7.15)\end{aligned}$$

where $\beta = \log(1 - 1/b)$, and the boundary conditions are

$$v(b) = c, \quad \text{and} \quad \rho_i(0)v(0) = x_i(1-r)(1-h_i), \quad i = 0, 1.$$

The spatial characteristics of the leaky buffer under CHOKe are totally captured by these differential equations. Now we present some structural properties of the velocity $v(y)$ and spatial distribution $\rho_0(y)$, which are shown in Theorems 7.9, 7.10, and illustrated in Figure 7.8.

Theorem 7.9. *1. For all $x_0 \geq 0$, packet velocity $v(y)$ is a convex and strictly decreasing function. It is linear if and only if $x_0 = x_1$.*

2. Suppose $x_0 > x_1$. Then $\rho_0(y)$ is a strictly decreasing function. Moreover, if $\rho_0(0) \leq \rho^*$, then $\rho_0(y)$ is convex. If $\rho_0(b) \geq \rho^*$, then $\rho_0(y)$ is concave. If $\rho_0(b) < \rho^* < \rho_0(0)$, then $\rho_0(y)$ is first concave and then convex (as it is shown in Figure 7.8(b)) , where $\rho^* = (x_0 - 2x_1)/(3(x_0 - x_1))$.

Now we study the asymptotic properties of $v(y)$ and $\rho_0(y)$ as x_0 goes to infinity. We assume that the pointwise limits of $v(y)$ and $\rho_0(y)$ exis and denote this by $v^\infty(y)$ and $\rho_0^\infty(y)$. We describe the asymptotic properties in the following theorem.

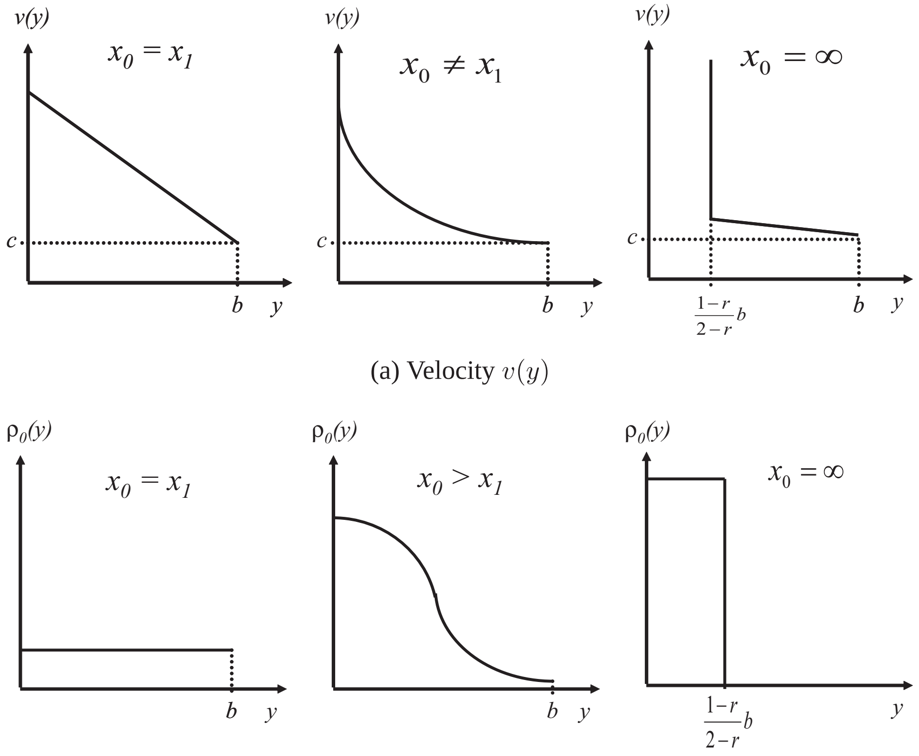
Theorem 7.10. *For any x_0, every flow, including UDP flow, occupies less than half of the queue. When $x_0 \rightarrow \infty$, we have*

1. *the buffer size b^∞ is finite. The UDP's share of this buffer is $h_0^\infty = (1-r^\infty)/(2-r^\infty)$.*

2. *the throughput of UDP source goes to zero, i.e., $x_0(1 - p_0) = \rho_0(b)c \rightarrow 0$.*

3. *let $y^* = b^\infty(1-r^\infty)/(2-r^\infty)$. When $0 \leq y \leq y^*$, we have $\rho_0^\infty = 1$ and $v^\infty(y) = \infty$. When $y^* < y \leq b^\infty$, we have $\rho_0^\infty = 0$ and $v^\infty(y) = c - \beta^\infty x_1^\infty\ (b^\infty - y)$*

When the UDP input rate increases, even though the total number of UDP packets in the queue increases, their spatial distribution becomes more and more concentrated near the tail of the queue and drops rapidly to zero toward the head of the queue. This means that most of the UDP packets are dropped before they reach the head. It is therefore possible to simultaneously maintain a large number of packets (concentrated near the tail) and receive a small bandwidth share, in stark contrast to the behavior of a non-leaky FIFO buffer. Indeed, as x_0 grows without bound, UDP share drops to 0. This also confirms the approximate throughput analysis of Theorem 7.8. Second, the packet velocity is infinite before the position y^* because UDP packets are being dropped at an infinite rate until y^*.



(a) Velocity $v(y)$

(b) Spatial distribution $\rho_0(y)$

Figure 7.8: Illustration of Theorems 7.9 and 7.10.

7.2.5 Simulations

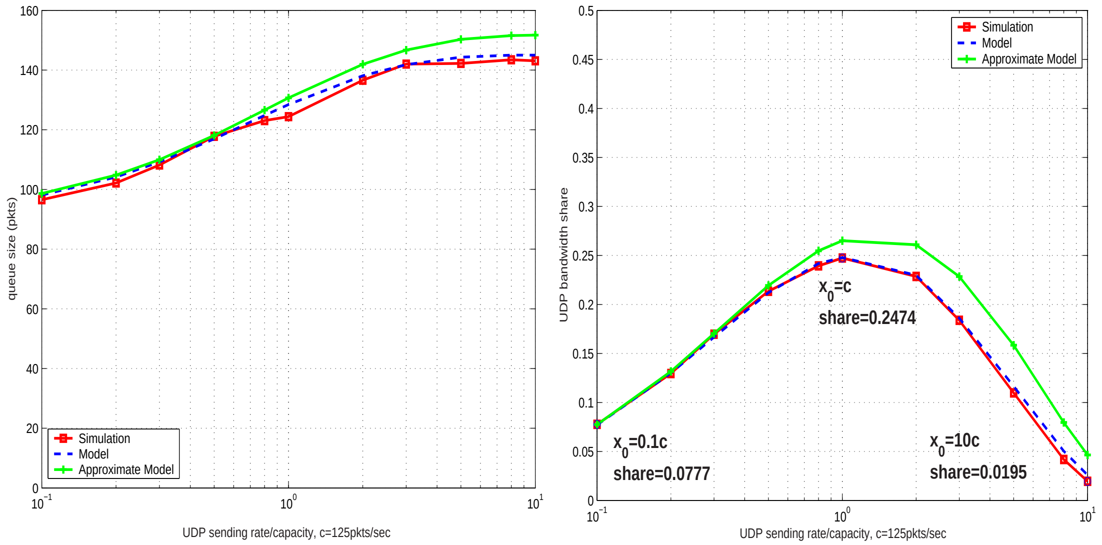
We implement a CHOKe module in ns-2 version 2.1b9. We have conducted extensive simulations with a single bottleneck link network. This network is shared by N newReno TCP sources and one UDP source which sends data at a constant rate. We present three sets of simulation results. The first set illustrates the accuracy of our TCP/CHOKe model (7.6)–(7.12) and its macroscopic properties. The second set illustrates the spatial properties proved in Theorem 7.10. The third example uses TCP Vegas and illustrates that these properties are insensitive to the specific TCP algorithms.

For the newReno simulations in the first two sets the link capacity is fixed at 125 pkts/sec, and round-trip propagation delay is 100ms. We use RED+CHOKe as the queue management with RED parameters minth $\underline{b} = 20$ packets, maxth $\overline{b} = 520$ packets, p_{max} = 0.5. The corresponding analytical model for Reno (function f) and RED (function g) can be found in Chapter 3.

In our simulation, we vary the UDP sending rate x_0 from $0.1c$ to $10c$ and measure the aggregate queue size b, UDP bandwidth share $\mu_0 = \rho_0(b)$, and TCP throughput μ_1. We also solve for these quantities using the analytical model (7.6)–(7.12) and the approximate model described in Section 7.2.3. The results, shown in Figures 7.9, illustrate both the macroscopic behavior of TCP/CHOKe and the accuracy of our analytical models.

As can be seen from Figure 7.9, the aggregate queue length b steadily increases as the UDP rate x_0 rises. UDP bandwidth share $\mu_0 = \rho_0(b)$ rises, peaks, and then drops to less than 5% as x_0 increases from $0.1c$ to $10c$, while the total TCP throughput follows an opposite trend, eventually exceeding 95% of the capacity (not shown). These results match closely those obtained in [126], for both the analytical model and the approximate mode. Figure 7.9(b) also displays the UDP bandwidth share measured from the simulations for the cases $x_0 = 0.1c, c, 10c$. It verifies Theorems 7.8 that predicts that the UDP bandwidth share peaks at around 0.269 and tends to zero as x_0 increases.

The next set of results measures the spatial distributions $\rho_0(y)$ of UDP packets in the above simulations shown in Figure 7.9. The simulation results, and analytical solutions, are both shown in Figure 7.10. They match well Theorem 7.10 and agree with Figure 7.8(b)



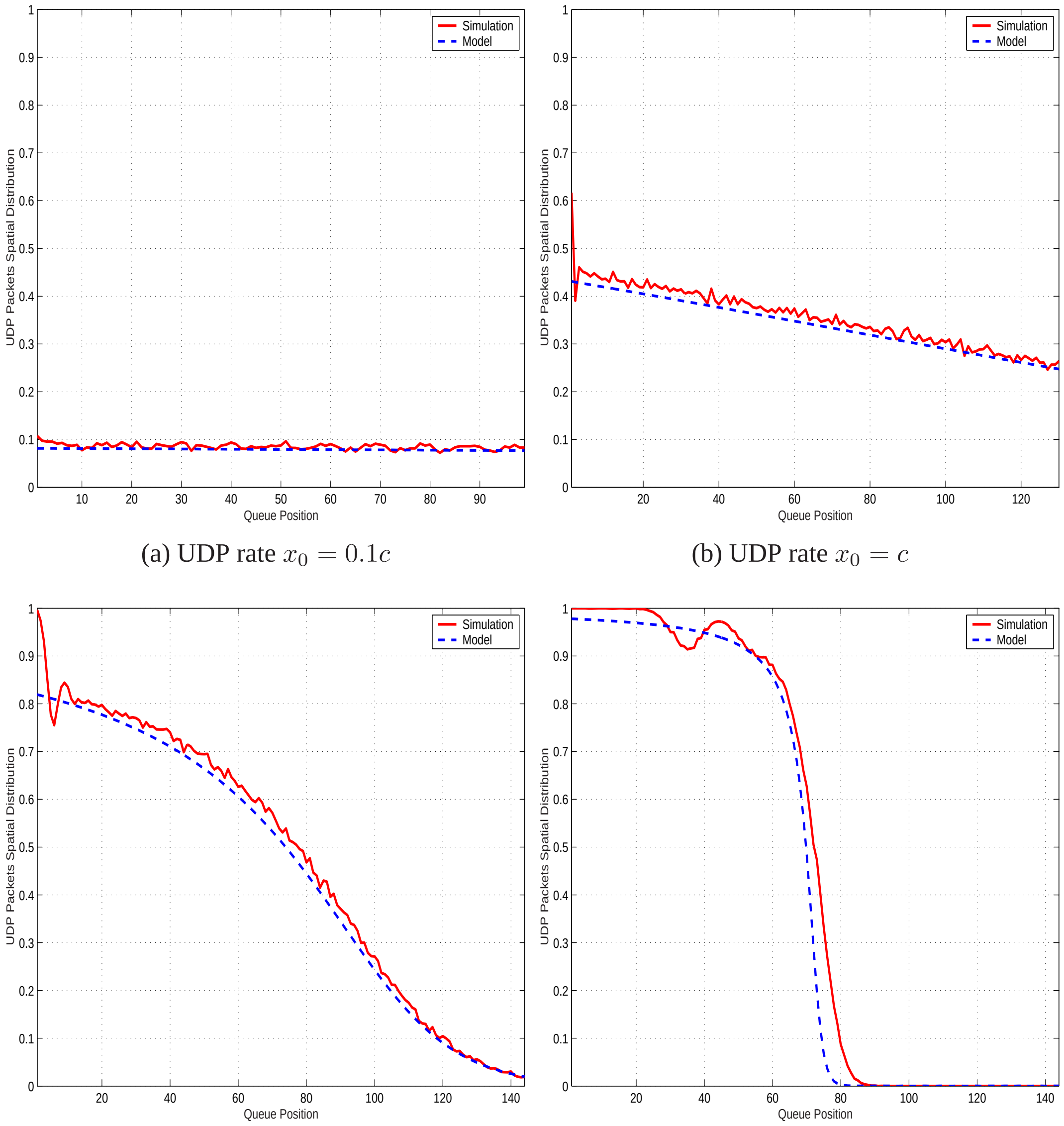
(a) Queue size b

(b) UDP bandwidth share μ_0

Figure 7.9: Experiment 1: effect of UDP rate x_0 on queue size and UDP share

in Section 7.2.4. When $x_0 = 0.1c$ (Figure 7.10(a)), UDP packets are distributed roughly uniformly in the queue, with probability close to 0.08 at each position. As a result, its bandwidth share is roughly 10%. As x_0 increase, $\rho_0(y)$ concentrates more and more near the tail of the queue and drops rapidly toward the head, as predicted by Theorem 7.10. Also marked in Figure 7.9(b) are the UDP bandwidth shares corresponding to UDP rates in Figure 7.10. As expected the UDP bandwidth shares in 7.9(b) are equal to $\rho_0(b)$ in Figure 7.10. When $x_0 > 10c$, even though roughly half of the queue is occupied by UDP packets, almost all of them are dropped before they reach the head of the queue!

In the last set of simulations, we use TCP Vegas [20] instead of newReno. In these simulations, the link capacity is fixed at $c = 1875$ pkts/sec., the round-trip propagation delay is $d = 100$ms, and the number of TCP sources is $N = 100$. We set Vegas parameter $\alpha d = 20$ packet, and use RED parameters $(20, 1020, 0.1)$. The UDP sending rate varies from $0.1c$ to $10c$. We measure the UDP bandwidth share μ_0 and queue length b, and compare them with the numerical solutions of the full model and those of the approximate model described. The results are shown in Figure 7.11. Comparison of this with Figure 7.9 for NewReno simulations confirms that the qualitative behavior of TCP/CHOKe is insensitive to TCP algorithms.



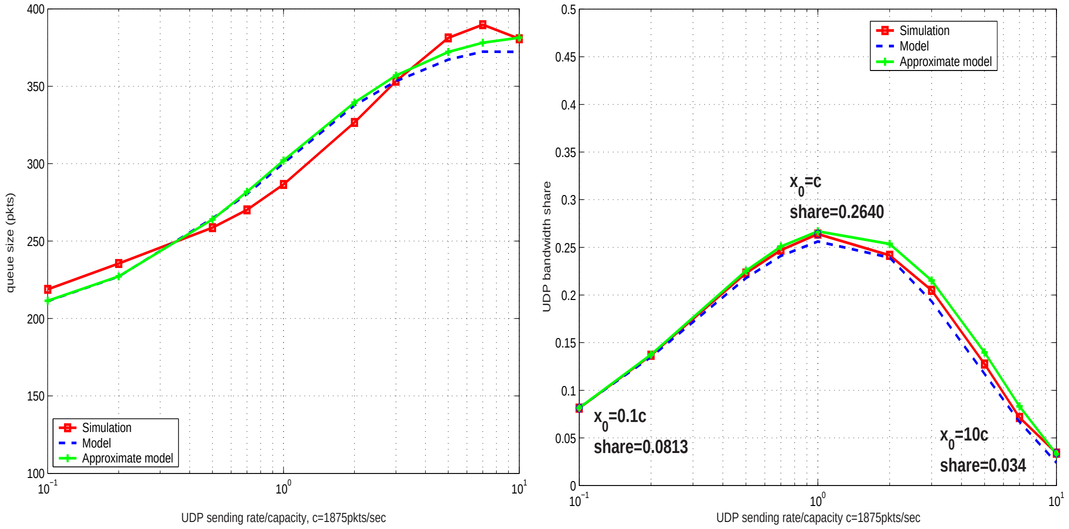
(a) UDP rate $x_0 = 0.1c$

(b) UDP rate $x_0 = c$

(c) UDP rate $x_0 = 10c$

(d) UDP rate $x_0 = 100c$

Figure 7.10: Experiment 2: spatial distribution $\rho(y)$.



(a) Queue size b

(b) UDP bandwidth share μ_0

Figure 7.11: Experiment 3: effect of UDP rate x_0 on queue size and UDP share with TCP Vegas.

7.2.6 Conclusions

We have developed a model of CHOKe. Its key features are the incorporation of the feedback equilibrium of TCP and a detailed modelling of the queue dynamics. We prove that as the UDP input rate increases, its bandwidth peaks at $(e+1)^{-1} = 0.269$ when the UDP input rate is slightly larger than link capacity, and drops to zero as the UDP input rate tends to infinity. To explain this phenomenon, we have introduced the concepts of spatial distribution and velocity of packets in the queue. We prove that structural and asymptotic properties of these quantities make it possible for UDP to simultaneously maintain a large number of packets in the queue and receive a vanishingly small bandwidth share, the mechanism through which CHOKe protects TCP flows.

Chapter 8

Summary and Future Directions

We have studied the equilibrium and dynamics of the Internet congestion control using tools recently developed from feedback-control theory and optimization theory. We summarize our work and list several future research directions in this chapter.

The models and dynamics of TCP Reno and FAST TCP have been studied in chapters 3 and 4. We point out several future directions in the studies of TCP dynamics.

1. TCP dynamics can be studied in several different settings, e.g., single link vs. general network, homogeneous sources vs. heterogenous sources, local stability vs. global stability, without feedback delay vs. with feedback delay. In general, the latter settings are more difficult to deal with. Our studies only cover part of them and should be extended to global stability with feedback delays in general networks.

2. As we mentioned in Section 7.1, during the incremental deployment of congestion control schemes, there is an inevitable phase of heterogenous protocols running on the same network. While the equilibrium properties of heterogenous protocols have been studied in [146], the dynamics of such systems are still open and is one of our future directions.

3. The fluid model of TCP has been widely used to study TCP dynamics; however this model can not capture the self-clocking feature in the packet level. We have shown in Chapter 4 that this model may give wrong predictions about stability. A discrete-time model is introduced to capture this *self-clocking* effect. However, we also found several scenarios where its predictions also disagree with the experiments. It seems

that both models are inaccurate. We need to clarify the discrepancies in these models and hopefully derive a better one.

Recent studies have shown that TCP/AQM algorithms can be interpreted as carrying out a distributed primal-dual algorithm over the Internet to maximize aggregate utility. The equilibrium properties (i.e., fairness, throughput, capacity, and routing) of TCP/AQM systems are studied using the utility maximization framework in Chapter 5, and 6. These studies can be also extended in several ways.

1. We have focused on how network throughput changes in response to changes in fairness and capacity in Chapter 6. However, how link prices and network revenue changes has not been investigated.

2. We have identified the existence of a non-trial duality gap, in the joint utility maximization problem in Chapter 5, and we need to derive a bound for this gap which is the penalty for not splitting the traffic.

3. Even though numerical examples suggest that the tradeoff between routing stability and utility maximization exists in a more general network, we have not been able to find an analytical proof.

4. When a static component is included in link cost, it is not known if TCP/IP has an equilibrium, whether the equilibrium jointly solves a certain optimization problem, and under what condition it is stable.

Bibliography

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

[2] M. Allman. A web server's view of the transport layer. *ACM Computer Communication Review*, 30(5), October 2000.

[3] M. Allman, V. Paxson, and W. Stevens. TCP congestion control. RFC 2581, Internet Engineering Task Force, April 1999. `http://www.ietf.org/rfc/rfc2581.txt`.

[4] T. Alpcan and T. Basar. A utility-based congestion control scheme for internet-style networks with delay. In *Proceedings of IEEE Infocom*, 2003.

[5] S. Athuraliya, V. H. Li, S. H. Low, and Q. Yin. REM: Active queue management. *IEEE Network*, 15(3):48–53, May/June 2001.

[6] S. Athuraliya and S. Low. Optimization flow control – ii: Implementation, 2000.

[7] B. Awerbuch, Y. Azar, and S. Plotkin. Throughput competitive online routing. In *Proceedings of 34th IEEE symposium on Foundations of Computer Science*, pages 32–40, 1993.

[8] J. Aweya, M. Ouellette, and D. Y. Montuno. A control theoretic approach to active queue management. *Computer Networks*, 36:203–235, 2001.

[9] F. Baccelli and D. Hong. Interaction of TCP Flows as Billiards. In *Proceedings of IEEE Infocom*, 2003.

[10] T. Basar and G. J. Olsder. *Dynamic Noncooperative Game Theory*. Academic Press, London and San Diego, second edition, 1995.

[11] N. Bean, F. Kelly, and P. Taylor. Braess's paradox in loss networks. *Journal of Applied Probability*, pages 155–159, 1997.

[12] D. P. Bertsekas. Dynamic behavior of shortest path routing algorithms for communication networks. *IEEE Transactions on Automatic Control*, pages 60–74, February 1982.

[13] D. P. Bertsekas. *Linear network optimization: algorithms and codes*. MIT Press, 1991.

[14] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1995.

[15] D. P. Bertsekas and R. Gallager. *Data Networks*. Prentice-Hall Inc., second edition, 1992.

[16] T. Bonald and L. Massoulie. Impact of fairness on Internet performance. In *Proceedings of ACM Sigmetrics*, pages 82–91, June 2001.

[17] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang. Recommendations on queue management and congestion avoidance in the Internet. RFC 2309, Internet Engineering Task Force, April 1998. `http://www.ietf.org/rfc/rfc2309.txt`.

[18] D. Braess. Über ein paradoxen aus der verkehrsplanung. *Unternehmensforschung*, 12:258–268, 1968.

[19] L. Brakmo, S. O'Malley, and L. Peterson. TCP Vegas: New techniques for congestion detection and avoidance. In *Proceedings of ACM Sigcomm*, pages 24–35, Aug. 1994.

[20] L. S. Brakmo and L. L. Peterson. TCP Vegas: End to end congestion avoidance on a global Internet. *IEEE Journal on Selected Areas in Communications*, 13(8):1465–1480, 1995.

[21] H. Bullot, R. L. Cottrell, and R. Hughes-Jones. Evaluation of advanced TCP stacks on fast long-distance production networks. In *Proceedings of Second International Workshop on Protocols for Fast Long-Distance Networks*, 2004.

[22] M. Butler and H. Williams. The allocation of shared fixed costs, 2002. `http://www.lse.ac.uk/collections/operationalResearch/pdf/lseor02-52.pdf`.

[23] F. M. Callier and C. A. Desoer. *Linear System Theory*, pages 368–374. Springer-Verlag, New York, 1991.

[24] H. Choe and S. H. Low. Stabilized Vegas. In *Proceedings of IEEE Infocom*, April 2003. `http://netlab.caltech.edu`.

[25] M. Christiansen, K. Jaffay, D. Ott, and F. D. Smith. Tuning RED for web traffic. In *Proceedings of ACM Sigcomm*, pages 139–150, 2000.

[26] M. Christiansen, K. Jeffay, D. Ott, and F. D. Smith. Tuning red for web traffic. *IEEE/ACM Trans. Netw.*, 9(3):249–264, 2001.

[27] J. Cohen and P. Horowitz. Paradoxical behaviour of mechanical and electrical networks. *Nature*, 352:699–701, August 1991.

[28] J. Cohen and F. Kelly. A paradox of congestion in a queuing network. *J. Appl. Prob.*, 27:730–734, 1990.

[29] S. Cosares and I. Saniee. An optimization problem related to balancing loads on SONET rings. *Telecommunications Systems*, 3:165–181, 1994.

[30] S. Deb and R. Srikant. Congestion control for fair resource allocation in networks with multicast flows. *IEEE/ACM Transactions on Networking*, 12(2):274–285, 2004.

[31] C. A. Desoer and Y. T. Yang. On the generalized nyquist stability criterion. *IEEE Transactions on Automatic Control*, 25:187–196, 1980.

[32] D. Dutta, A. Goel, and J. Heidemann. Oblivious aqm and nash equilibria. In *Proceedigns of IEEE Infocom*, 2003.

[33] FAST-Team. FAST implementation code. `http://netlab.caltech.edu/FAST/`.

[34] W. Feng, D. D. Kandlur, D. Saha, and K. G. Shin. A self-configuring RED gateway. In *Proceedings of IEEE Infocom*, volume 3, pages 1320–1328, March 1999.

[35] W. Feng, K. G. Shin, D. Kandlur, and D. Saha. Stochastic Fair Blue: A queue management algorithm for enforcing fairness. In *Proceedings of IEEE Infocom*, 2001.

[36] V. Firoiu and M. Borden. A study of active queue management for congestion control. In *Proceedings of IEEE Infocom*, pages 1435–1444, March 2000.

[37] C. Fisk and S. Pallottino. Empirical evidence for equilibrium paradoxes with implications for optimal planning strategies. *Transportation Research*, 15A:245–248, 1981.

[38] S. Floyd. Connections with multiple congested gateways in packet-switched networks part 1: One-way traffic. *Computer Communications Review*, 21(5):30–47, Ocotober 1991.

[39] S. Floyd. Highspeed TCP for large congestion windows. RFC 3649, IETF Experimental, December 2003. `http://www.faqs.org/rfcs/rfc3649.html`.

[40] S. Floyd and T. Henderson. The NewReno modification to TCP's fast recovery algorithm. RFC 2582, Internet Engineering Task Force, April 1999. `http://www.ietf.org/rfc/rfc2582.txt`.

[41] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, 1993.

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

[43] C. Fraleigh, S. Moon, B. Lyles, M. Khan, D. Moll, R. Rockell, T. Seely, and C. Diot. Packet level traffic measurements from the sprint IP backbone. *IEEE Network*, 17(6):6–16, Nov.-Dec. 2003.

[44] M. Frank. The braess paradox. *Mathematical Programming*, 20:283–302, 1981.

[45] E. M. Gafni and D. P. Bertsekas. Dynamic control of session input rates in communication networks. *IEEE Transactions on Automatic Control*, 29(11):1009–1016, January 1984.

[46] R. G. Gallager and S. J. Golestani. Flow control and routing algorithms for data networks. In *Proceedings of the 5th International Conf. Comp. Comm.*, pages 779–784, 1980.

[47] M. Garey and D. Johnson. *Computers and intractability: a guide to the theory of NP-completeness*. W. H. Freeman and Co., 1979.

[48] N. Garg and J. Konemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In *Proceedings of IEEE Symposium on Foundations of Computer Science*, pages 300–309, 1998.

[49] R. Garg, A. Kamra, and V. Khurana. A game-theoretic approach towards congestion control in communication networks. *SIGCOMM Comput. Commun. Rev.*, 32(3):47–61, 2002.

[50] R. J. Gibbens and F. P. Kelly. Resource pricing and the evolution of congestion control. *Automatica*, 35:1969–1985, 1999.

[51] T. G. Griffin, F. B. Shepherd, and G. Wilfong. The stable paths problem and interdomain routing. *IEEE/ACM Trans. Netw.*, 10(2):232–243, 2002.

[52] K. Gummadi, R. Dunn, S. Saroiu, S. Gribble, H. Levy, and J. Zahorjan. Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. In *Proceedings of the 19th ACM SOSP*, Bolton Landing, NY, October 2003.

[53] L. Guo and I. Matta. The war between mice and elephants. In *Proceedings of IEEE ICNP*, November 2001.

[54] M. Handley, S. Floyd, J. Padhye, and J. Widmer. TCP Friendly Rate Control (TFRC): Protocol specification. RFC 3168, Internet Engineering Task Force, January 2003. `http://www.ietf.org/rfc/rfc3384.txt`.

[55] E. Hashem. Analysis of random drop for gateway congestion control. Technical Report LCS TR-465, Laboratory for Computer Science, MIT, Cambridge MA, 1989.

[56] C. Hedrick. Routing information protocol. RFC 1058, Internet Engineering Task Force, June 1988. `http://www.ietf.org/rfc/rfc1058.txt`.

[57] J. C. Hoe. Improving the start-up behavior of a congestion control sheme for TCP. In *Proceedings of ACM Sigcomm*, pages 270–280, New York, 1996.

[58] C. Hollot, V. Misra, and W. Gong. On designing improved controllers for AQM routers supporting TCP flows. In *Proceedings of IEEE Infocom*, 2001.

[59] C. Hollot, V. Misra, and W. Gong. Analysis and design of controllers for AQM routers supporting TCP flows. *IEEE Transactions on Automatic Control*, 47(6), 2002.

[60] C. Hollot, V. Misra, D. Towsley, and W. Gong. A control theorietic analysis of RED. In *Proceedings of IEEE Infocom*, 2001.

[61] C. Hollot, V. Misra, D. Towsley, and W. Gong. On designing improved controller for AQM routers supporting TCP flows. In *Proceedings of IEEE Infocom*, 2001.

[62] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

[63] V. Jacobson. Congestion avoidance and control. In *Proceedings of ACM Sigcomm*, pages 314–329, Stanford, CA, Aug. 1988.

[64] V. Jacobson, R. Braden, and D. Borman. TCP extensions for high performance. RFC 1323, Internet Engineering Task Force, May 1992. `http://www.ietf.org/rfc/rfc1323.txt`.

[65] R. Jain. A delay-based approach for congestion avoidance in interconnected heterogeneous computer networks. *Comput. Commun. Rev.*, 19(5):56C–71, October 1989.

[66] R. Jain. Congestion control in computer networks: Issues and trends. *IEEE Network*, pages 24–30, May 1990.

[67] R. Jain, W. Hawe, and D. Chiu. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. Technical Report DEC TR-301, Digital Equipment Corporation, September 1984.

[68] C. Jin, D. X. Wei, and S. H. Low. FAST TCP: motivation, architecture, algorithms, performance. Technical Report CaltechCSTR:2003:010, Caltech, December 2003.

[69] C. Jin, D. X. Wei, and S. H. Low. FAST TCP: motivation, architecture, algorithms, performance. In *Proceedings of IEEE Infocom*, March 2004. `http://netlab.caltech.edu`.

[70] C. Jin, X. Wei, S. H. Low, G. Buhrmaster, J. Bunn, D. H. Choe, R. L. A. Cottrell, J. C. Doyle, W. Feng, O. Martin, H. Newman, F. Paganini, S. Ravot, and S. Singh. FAST TCP: From Theory to Experiments. *IEEE Network*, 19(1):4–11, January/February 2005.

[71] R. Johari and D. K. Tan. End-to-end congestion control for the Internet: delays and stability. *IEEE/ACM Transactions on Networking*, 9(6):818–832, 2001.

[72] H. Kameda, E. Altman, T. Kozawa, and Y. Hosokawa. Braess-like paradoxes in distributed computer systems. *IEEE Transactions on Automatic Control*, 45(9):1687–1690, 2000.

[73] H. Kameda and O. Pourtallier. Paradoxes in distributed decisions on optimal load balancing for networks of homogeneous computers. *Journal of the ACM*, 49:407–433, 2002.

[74] K. Kar, S. Sarkar, and L. Tassiulas. Optimization based rate control for multipath sessions. In *Proceedings of 7th International Teletraffic Congress (ITC)*, December 2001.

[75] K. Kar, S. Sarkar, and L. Tassiulas. Optimization based rate control for multirate multicast sessions. In *Proceedings of IEEE Infocom*, pages 123–132, April 2001.

[76] D. Katabi, M. Handley, and C. Rohrs. Congestion control for high bandwidth-delay product networks. In *Proceedings of ACM Sigcomm*, 2002.

[77] F. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8:33–37, 1997.

[78] F. Kelly and T. Voice. Stability of end-to-end algorithms for joint routing and rate control. *Computer Communication Review*, 35(2):5–12, 2005.

[79] F. P. Kelly. Fairness and stability of end-to-end congestion control. *European Journal of Control*, 9:159–176, 2003.

[80] F. P. Kelly, A. Maulloo, and D. Tan. Rate control for communication networks: Shadow prices, proportional fairness and stability. *Journal of Operations Research Society*, 49(3):237–252, March 1998.

[81] T. Kelly. Scalable TCP: improving performance in highspeed wide area networks. *ACM SIGCOMM Computer Communication Review.*, 33(2):83–91, 2003.

[82] K. B. Kim and S. H. Low. Analysis and design of aqm for stabilizing tcp. Technical Report CaltechCSTR:2002:009, Caltech, March 2002.

[83] Y. Korilis, A. Lazar, and A. Orda. Capacity allocation under noncooperative routing. *IEEE Trans. on Automatic Control*, 42(3):309–325, 1997.

[84] Y. Korilis, A. Lazar, and A. Orda. Avoiding the Braess paradox in non-cooperative networks. *J. Appl. Prob.*, 36:211–222, 1999.

[85] S. Kunniyur and R. Srikant. End-to-end congestion control schemes: Utility functions, random losses and ECN marks. In *Proceedings of IEEE Infocom*, pages 1323–1332, March 2000.

[86] S. Kunniyur and R. Srikant. Analysis and design of an adaptive virtual queue (avq) algorithm for active queue management. In *Proceedings of ACM Sigcomm*, pages 123–134, New York, NY, USA, 2001. ACM Press.

[87] S. Kunniyur and R. Srikant. A time–scale decomposition approach to adaptive ECN marking. In *Proceedings of IEEE Infocom*, pages 1330–1339, April 2001. `http://comm.csl.uiuc.edu:80/~srikant/pub.html`.

[88] S. Kunniyur and R. Srikant. End-to-end congestion control: utility functions, random losses and ECN marks. *IEEE/ACM Transactions on Networking*, 11(5):689 – 702, October 2003.

[89] R. J. La and V. Anantharam. Charge-sensitive TCP and rate control in the internet. In *INFOCOM (3)*, pages 1166–1175, 2000.

[90] T. V. Lakshman and U. Madhow. The performance of TCP/IP for networks with high bandwidth-delay products and random loss. *IFIP Transactions C-26, High Performance Networking*, pages 135–150, 1994.

[91] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Trans. Netw.*, 2(1):1–15, 1994.

[92] L. Li, D. Alderson, W. Willinger, and J. Doyle. A first-principles approach to understanding the Internet's router-level topology. In *Proceedings of ACM Sigcomm*, pages 3–14, 2004.

[93] L. Li, D. Alderson, W. Willinger, and J. Doyle. Towards a theory of scale free graphs, definition, properties and implications. *Internet Mathematics*, 2005. To appear.

[94] D. Lin and R. Morris. Dynamics of random early detection. In *Proceedings of the ACM Sigcomm*, pages 127–137, 1997.

[95] X. Lin and N. B. Shroff. Utility maximization for communication networks with multi-path routing. *Submitted to IEEE Transactions on Automatic Control*, 2004. http://min.ecn.purdue.edu/~linx/.

[96] S. H. Low. A duality model of TCP and queue management algorithms. *IEEE/ACM Transactions on Networking*, 11(4):525–536, August 2003. http://netlab.caltech.edu.

[97] S. H. Low and D. E. Lapsley. Optimization flow control I: basic algorithm and convergence. *IEEE/ACM Transactions on Networking*, 7(6):861–874, 1999.

[98] S. H. Low, F. Paganini, and J. C. Doyle. Internet congestion control. *IEEE Control Systems Magazine*, 22(1):28–43, Feb. 2002.

[99] S. H. Low, F. Paganini, J. Wang, S. Adlakha, and J. C. Doyle. Dynamics of TCP/RED and a scalable control. In *Proceedings of IEEE INFOCOM*, June 2002. http://netlab.caltech.edu.

[100] S. H. Low, F. Paganini, J. Wang, and J. C. Doyle. Linear stability of TCP/RED and a scalable control. *Computer Networks Journal*, 43(5):633–647, 2003. http://netlab.caltech.edu.

[101] S. H. Low, L. Peterson, and L. Wang. Understanding Vegas: a duality model. *Journal of ACM*, 49(2):207–235, March 2002. http://netlab.caltech.edu.

[102] S. H. Low and R. Srikant. A mathematical framework for designing a low-loss, low-delay internet. *Networks and Spatial Economics*, 4(1), 2004.

[103] S. H. Low and P. Varaiya. Dynamic behavior of a class of adaptive routing protocols (IGRP). In *Proceedings of IEEE Infocom*, pages 610–616, March 1993.

[104] D. G. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley Publishing Company, second edition, 1984.

[105] H. Luo, S. Lu, V. Bharghavan, J. Cheng, and G. Zhong. A packet scheduling approach to qos support in multihop wireless networks. *Mob. Netw. Appl.*, 9(3):193–206, 2004.

[106] L. Massoulie. Stability of distributed congestion control with heterogenous feedback delays. *IEEE Transactions on Automatic Control*, 47:895–902, 2002.

[107] L. Massoulie and J. Roberts. Bandwidth sharing: objectives and algorithms. *IEEE/ACM Transactions on Networking*, 10(3):320–328, June 2002.

[108] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow. TCP selective acknowledgment options. RFC 2018, Internet Engineering Task Force, October 1996. `http://www.ietf.org/rfc/rfc2018.txt`.

[109] M. Mathis, J. Semke, J. Mahdavi, and T. Ott. The macroscopic behavior of the TCP congestion avoidance algorithm. *Computer Communication Review*, 27(1), July 1997.

[110] M. May, T. Bonald, and J.-C. Bolot. Analytic evaluation of RED performance. In *Proceedings of IEEE Infocom*, 2000.

[111] P. McKenny. Stochastic fairness queueing. In *Proceedings of IEEE Infocom*, pages 733–740, 1990.

[112] M. Mehyar, D. Spanos, and S. H. Low. Optimization flow control with estimation error. In *Proceedings of IEEE Infocom*, March 2004.

[113] J. Milnor. *Topology from the Differentiable Viewpoint*. The University Press of Virginia, Charlottesville, 1972.

[114] V. Misra, W. Gong, and D. Towsley. Stochastic differential equation modeling and analysis of tcp-windowsize behavior, 1999.

[115] V. Misra, W. Gong, and D. Towsley. Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED. In *Proceedings of ACM Sigcomm*, 2000.

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

[117] R. T. Morris. *Scalable TCP Congestion Control*. PhD thesis, Harvard University, 1999.

[118] G. Motwani and K. Gopinath. Evaluation of advanced TCP stacks in the iSCSI environment using simulation model. In *Proceedings. 22nd IEEE / 13th NASA Goddard Conference on Mass Storage Systems and Technologies*, pages 210–217, 2005.

[119] J. Moy. OSPF version 2. RFC 2328, Internet Engineering Task Force, April 1998. `http://www.ietf.org/rfc/rfc2328.txt`.

[120] J. Murchland. Braess's paradox of traffic flow. *Transpn. Res.*, 4:391–394, 1970.

[121] M. Osborne and A. Rubinstein. *A Course in Game Theory*. The MIT Press, 1994.

[122] T. Ott, J. Kemperman, and M. Mathis. The stationary behavior of ideal TCP congestion avoidance. ftp://ftp.bellcore.com/pub/tjo/TCPwindow.ps.

[123] T. J. Ott, T. V. Lakshman, and L. Wong. SRED: Stabilized RED. In *Proceedings of IEEE Infocom*, 1999. `ftp://ftp.bellcore.com/pub/tjo/SRED.ps`.

[124] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP Reno performance: A simple model and its empirical validation. *IEEE/ACM Transactions on Networking*, 8(2):133–145, April 2000.

[125] F. Paganini, Z. Wang, J. C. Doyle, and S. H. Low. Congestion control for high performance, stability, and fairness in general networks. *IEEE/ACM Transactions on Networking*, 13(1):43–56, 2005.

[126] R. Pan, B. Prabhakar, and K. Psounis. CHOKe: A stateless AQM scheme for approximating fair bandwidth allocation. In *Proceedings of IEEE Infocom*, March 2000.

[127] A. Papachristodoulou. Global stability analysis of a TCP/AQM protocol for arbitrary networks with delay. In *Proceedings of IEEE Confernece on Decision and Control*, December 2004.

[128] A. Papachristodoulou, L. Li, and J. C. Doyle. Methodological frameworks for large-scale network analysis and design. *Computer Communication Review*, 34(3), 2004.

[129] L. L. Peterson and B. S. Davie. *Computer Networks: A Systems Approach*, chapter 5.2, pages 383–389. Morgan Kaufmann Publishers, Inc., second edition, 2000.

[130] S. Plotkin, D. Shmoys, and E. Tardos. Fast approximation algorithms for fractional packing and covering problems. *Math of Oper. Research*, pages 257–301, 1995.

[131] K. Ramakrishnan, S. Floyd, and D. Black. The addition of explicit congestion notification (ECN) to IP. RFC 3168, Internet Engineering Task Force, September 2001. `http://www.ietf.org/rfc/rfc3168.txt`.

[132] L. Rizzo. IP dummynet. `http://http://info.iet.unipi.it/~luigi/ip_dummynet/`.

[133] S. Ryu, C. Rump, and C. Qiao. Advances in active queue management(AQM) based TCP congestion control. *Telecommunication System*, 25(3):317–351, 2004.

[134] S. Shakkottai, R. Srikant, and S. P. Meyn. Bounds on the throughput of congestion controllers in the presence of feedback delay. *IEEE/ACM Trans. Netw.*, 11(6):972–981, 2003.

[135] J. L. Sobrinho. Network routing with path vector protocols: theory and applications. In *Proceedings of ACM Sigcomm*, pages 49–60, New York, NY, USA, 2003. ACM Press.

[136] B. C. W. Solomon and I. Ziedins. Braess's paradox in a queuing network with state depending routing. *Journal of Applied Probability*, 34:134–154, 1997.

[137] Sprint ATL. IP monitoring project, `http://ipmon.sprint.com/packstat/packetoverview.php`.

[138] R. Srikant. *The Mathematics of Internet Congestion Control*. Birkhauser, 2004.

[139] R. Srinivasan and A. Somani. On achieving fairness and efficiency in high-speed shared medium access. *IEEE/ACM Transactions on Networking*, 11(1), February 2003.

[140] W. Stevens. TCP slow start, congestion avoidance, fast retransmit, and fast recovery. RFC 2001, Internet Engineering Task Force, January 1997. `http://www.ietf.org/rfc/rfc2001.txt`.

[141] I. Stoica, S. Shenker, and H. Zhang. Core-stateless fair queueing: Achieving approximately fair bandwidth allocations in high speed networks. In *Proceedings of ACM Sigcomm*, pages 118–130, 1998.

[142] A. Tang, J. Wang, S. Hedge, and S. H. Low. Equilibrium and fairness of networks shared by TCP Reno and Vegas/FAST. *To appear in Telecommunication Systems*, 2005.

[143] A. Tang, J. Wang, and S. H. Low. Understanding CHOKe. In *Proceedings of IEEE Infocom*, April 2003. `http://netlab.caltech.edu`.

[144] A. Tang, J. Wang, and S. H. Low. Is fair allocation always inefficient. In *Proceedings of IEEE Infocom*, 2004.

[145] A. Tang, J. Wang, and S. H. Low. Understanding choke: Throughput and spatial characteristics. *IEEE/ACM Trans. Netw.*, 12(4):694–707, 2004.

[146] A. Tang, J. Wang, and S. H. Low. Counter-intuitive throughput behaviors in networks under end-to-end control. *To appear in IEEE/ACM Transactions on Networking*, June 2006.

[147] A. Tang, J. Wang, S. H. Low, and M. Chiang. Network equilibrium of heterogeneous congestion control protocols. In *Proceedings of IEEE Infocom*, Miami, FL, March 2005.

[148] P. Tinnakornsrisuphap and A. M. Makowski. Limit behavior of ECN/RED gateways under a large number of TCP flows. In *Proceedings of IEEE Infocom*, pages 873–883, April 2003.

[149] H. Varian. A third remark on the number of equilibria of an economy. *Econometrica*, 43:985–986, 1975.

[150] H. R. Varian. *Microeconomic analysis*. W. W. Norton & Company, third edition, 1992.

[151] G. Vinnicombe. On the stability of end-to-end congestion control for the Internet. CUED/F-INFENG/TR.398, December 2000. `http://www-control.eng.cam.ac.uk/gv/internet/`.

[152] G. Vinnicombe. On the stability of networks operating TCP-like protocols. In *Proceedings of IFAC*, 2002. `http://netlab.caltech.edu/pub/papers/gv_ifac.pdf`.

[153] J. Wang, L. Li, S. H. Low, and J. C. Doyle. Can TCP and shortest-path routing maximize utility? In *Proceedings of IEEE Infocom*, April 2003. `http://netlab.caltech.edu`.

[154] J. Wang, L. Li, S. H. Low, and J. C. Doyle. Cross-layer optimization in TCP/IP networks. *IEEE/ACM Transactions on Networking*, 2005. `http://netlab.caltech.edu`.

[155] J. Wang, A. Tang, and S. H. Low. Maximum and asymptotic UDP throughput under CHOKe. In *Proceedings of ACM Sigmetrics*, pages 82–90, 2003.

[156] J. Wang, A. Tang, and S. H. Low. Local stability of FAST TCP. In *Proceedings of IEEE Conference on Decision and Control*, December 2004.

[157] J. Wang, D. X. Wei, and S. H. Low. Modeling and stability of FAST TCP. In *Proceedings of IEEE Infocom*, Miami, FL, March 2005.

[158] Z. Wang and F. Paganini. Global stability with time delay in network congestion control. In *Proceedings of IEEE CDC*, December 2002.

[159] B. M. Waxman. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 6(9):1617–1622, 1988.

[160] D. X. Wei. Congestion control algorithms for high speed long distance tcp. Master's Thesis, Caltech, 2004 `http://www.cs.caltech.edu/~weixl/research/msthesis.ps`.

[161] J. T. Wen and M. Arak. A unifying passivity framework for network flow control. In *Proceedings of IEEE Infocom*, 2003.

[162] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson. Self-similarity through high-variability: statistical analysis of ethernet lan traffic at the source level. *IEEE/ACM Trans. Netw.*, 5(1):71–86, 1997.

[163] L. Xu, K. Harfoush, and I. Rhee. Binary increase congestion control for fast long-distance networks. In *Proceedings of IEEE Infocom*, March 2004.

[164] H. Yaiche, R. R. Mazumdar, and C. Rosenberg. A game theoretic framework for bandwidth allocation and pricing in broadband networks. *IEEE/ACM Transactions on Networking*, 8(5):667–678, 2000.

[165] R. H. Zakon. Hobbes' Internet timeline, v8.0, 2005, link: http://www.zakon.org/robert/internet/timeline/.

[166] H. Zhang, L. Baohong, and W. Dou. Design of a robust active queue management algorithm based on feedback compensation. In *Proceedings of ACM Sigcomm*, pages 277–285, 2003.

[167] X. Zhu, J. Yu, and J. C. Doyle. Heavy tails, generalized coding and optimal web layout. In *Proceedings of IEEE Infocom*, 2001.