New Bandwidth Allocation Methods to Provide Quality-of-Experience Fairness for Video Streaming Services

by

Mahdi Hemmati

Thesis submitted
in partial fulfillment of the requirements
for the Ph.D. degree in
Electrical and Computer Engineering

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

Video streaming over the best-effort networks is a challenging problem due to the time-varying and uncertain characteristics of the links. When multiple video streams are present in a network, they share and compete for the common bandwidth. In such a setting, a bandwidth allocation algorithm is required to distribute the available resources among the streams in a *fair* and *efficient* way. Specifically, it is desired to establish fairness across end-users' Quality of Experience (QoE).

In this research, we propose three novel methods to provide QoE-fair network bandwidth allocation among multiple video streaming sessions. First, we formulate the problem of bandwidth allocation for video flows in the context of Network Utility Maximization (NUM) framework, using sigmoidal utility functions, rather than conventional but unrealistic concave functions. An approximation algorithm for Sigmoidal Programming (SP) is utilized to solve the resulting nonconvex optimization problem, called NUM-SP. Simulation results indicate improvements of at least 60% in average utility/QoE and 45% in fairness, while using slightly less network resources, compared to two representative methods.

Subsequently, we take a collaborative decision-theoretic approach to the problem of rate adaptation among multiple video streaming sessions, and design a multi-objective foresighted optimization model for network resource allocation. A *social welfare* function is constructed to capture both fairness and efficiency objectives at the same time. Then, assuming a common altruistic goal for all network users, we use multi-agent decision processes to find the optimal policies for all players.

We propose a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) model for the conventional IP networks and a Multi-agent Markov Decision Process (MMDP) model for the SDN-enabled wireless networks. By *planning* these cooperative decision process models, we find the optimal network bandwidth allocation that leads to social welfare maximization. Distributed multi-agent reinforcement *learning* algorithms are also designed and proposed as a low-complexity model-free solution to these optimization problems.

Simulations of the proposed methods show that the resulting optimal policies of the novel *Social Utility Maximization* (SUM) framework outperform existing approaches in terms of both efficiency and fairness. The Dec-POMDP model applied to a server-side rate adaptation results in 25% improvement in efficiency and 13% improvement in fairness, compared to one popular protocol of congestion control for multimedia streaming. Our performance evaluations also show that the MMDP model applied to a client-side rate adaptation like DASH improves efficiency, fairness, and social welfare by as much as 18%, 24%, and 25%, respectively compared to current state-of-the-art.

Acknowledgements

I would like to express my gratitude to my supervisor Prof. Shervin Shirmohammadi for his continuous support of my PhD research, for his trust, flexibility, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Jiangchuan Liu, Prof. Chris Joslin, Prof. Amiya Nayak, and Prof. Tet Yeap, for their insightful comments and constructive feedback.

Also many thanks to the professors who prepared this helpful thesis template for LaTeX: Colin Campbell, Ray White, Stephen M. Carr and Wail Gueaieb.

Additionally, I would like to thank the Canadian federal and provincial governments for funding my research over the past couple of years, through QEII-GSST and NSERC-PGS scholarships.

My sincere thanks also goes to Dr. Abdulsalam Yassine, for the stimulating and fruitful discussions we had over the course of my PhD research.

Last but not least, I am profoundly grateful to all my family members: my amazing wife Isar, my lovely children Kosar and Sadra, my dedicated parents, my siblings, and my in-laws for their strong and continuous support which propelled me to complete this work.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

Acronyms

AIMD	Additive Increase Multiplicative Decrease
DASH	Dynamic Adaptive Streaming over HTTP
DCCP	Datagram Congestion Control Protocol
Dec-POMDP	Decentralized Partially-Observable Markov Decision Process
HAS	HTTP Adaptive Streaming
IETF	Internet Engineering Task Force
MAS	Multi-Agent System
MDP	Markov Decision Process
MMDP	Multi-agent Markov Decision Process
NUM	Network Utility Maximization
NUM-SP	Network Utility Maximization using Sigmoidal Programing
PLR	Packet Loss Rate
POMDP	Partially-Observable Markov Decision Process
QoE	Quality of Experience
QoS	Quality of Service
RAP	Rate Adaptation Protocol
RTP	Real-time Transport Protocol
RTT	Round-Trip Time
SUM	Social Utility Maximization
TCP	Transmission Control Protocol
TFRC	TCP-Friendly Rate Control
UDP	User Datagram Protocol

Chapter 1

Introduction

Video traffic on the Internet has been growing at a rapid pace during recent years. Globally, IP video traffic will be 82% of all consumer Internet traffic by 2020, up from 70% in 2015, growing fourfold during this time period [1]. This percentage does not even include video exchanged through peer-to-peer file sharing. An increasing fraction of this video traffic comes from video streaming services, such as live streaming, video-on-demand and over-the-top (OTT) video services from the likes of Netflix, YouTube, and Amazon Video, the top three sources of video traffic in North America [2]. The growth in video traffic has been accompanied by a modern technology: *HTTP-based Adaptive Streaming* (HAS) [3] [4], which has received a great popularity in the market and a lot of research interest in academia [5], [6], [7], [8], [9] and industry [10]. In OTT services, whether using HAS or the legacy technologies, the video has to be streamed in a continuous manner and without interruption in playback or degradation in quality, as much as possible.

Due to the shared and best-effort nature of the Internet, it is difficult to stream these videos without fluctuations in quality, or interruptions in playback. Since there are multiple video streams competing for the shared and limited bandwidth, packet losses and bandwidth fluctuations will adversely affect the user's Quality of Experience (QoE). To overcome this, we need a *fair* and *efficient* method that allocates the optimum amount of bandwidth for each video stream such that congestion is avoided as much as possible while video quality is as stable as possible and balanced in a fair manner among all users. Many related work have attempted to design and implement such a method, but as we will discuss in next section, existing methods suffer from one or more of the following shortcomings: they are not aware of QoE, and simply optimize the available bandwidth without considering the user-perceived quality of the video; they do not consider the multi-agent nature of the problem resulting from many interacting players and their inherent conflicts and cooperations; they do not consider the longer-term expected quality caused by the dynamic characteristics of the network and how any change in one user's allocated rate would affect all other users, in turn affecting the first user in subsequent cycles; or they do not address the fairness of the QoE, and at best address the fair distribution of bandwidth.

1.1 Motivation

To offer an acceptable QoE for consumers, video playback should be as smooth as possible, without any interruption or degradation in quality. But doing so requires high bandwidth and low packet loss, which imposes new challenges to the existing best-effort Internet. These challenges are exacerbated by the fact that there is not one but multiple video streams concurrently running on the network, potentially competing for the limited bandwidth. Hence, a *fair* and *efficient* video rate allocation model is required to 1) prevent congestion, and 2) provide a balanced video quality to all end users.

Efficiency is synonymous with *Pareto optimality*: a resource allocation is efficient or Pareto optimal if it is impossible to make one user better off without making some other users worse off. Efficiency is necessary, but not sufficient, for social optimality. Together with fairness, they define a socially optimal resource allocation.

Traditionally, a distributed solution for video rate adaptation is needed for congestion avoidance and bandwidth sharing among multiple video streams. By leveraging some aspects of the novel paradigm of Software-Defined Networking (SDN), one can also think of a centralized authority for network resource allocation. The design and implementation of such a solution, whether distributed and end-node-driven, or centralized and network-assisted, is a challenging problem in today's video delivery industry. Most conventional congestion control and rate adaptation methods (e.g. [11], [12], [13], [14], and [15]) are basically flow-based and have four major shortcomings when it comes to multimedia streaming:

1. **Not aware of media quality**
   Most transport layer congestion control approaches are application-agnostic [16]. They only attempt to avoid network congestion by adjusting the sending rates, without considering the impact on the application's performance, i.e. quality of media transported in the case of multimedia streaming. Maximizing the bandwidth utilization does not necessarily result in an optimum video quality. Although end-user's *Quality of Experience* (QoE) is the final goal of any network service, the bulk of existing literature on network optimization does not take QoE into consideration, and instead tries to maximize network utilization or throughput. QoE, which is generally defined as "a measure of the overall acceptability of an application or service,

as perceived subjectively by the end-user", is the ultimate measure of utility to be maximized in any multimedia application. In order to incorporate QoE into resource allocation process, some sort of mapping of specific network performance metrics, also known as Quality of Service (QoS) metrics to QoE is required. While there is an obvious relationship between packet loss, delay and jitter and QoE, no clear mapping has been easily drawn and employed due to the complexity of the media compression and delivery of the services.

2. **Not capturing the multi-agent nature of the problem**
Users of any shared resource like network bandwidth are generally facing a strategic situation, which involves more than one utility-maximizing decision-maker interacting with each other and competing for obtaining larger piece of the cake. Although multimedia streaming is obviously a multi-agent problem with many interacting players, there are very few works in the literature that are established upon models of Multi-Agent Systems (MAS), studying the conflicts and cooperations among different players. Most existing approaches use a stochastic passive model to represent network behavior as an aggregation of all other active decision makers, neglecting the essence of interaction between different users of the network.

3. **Myopic adaptation only based on instantaneous rates**
Due to dynamic characteristics of the network, any changes in current sending rate of a user would affect the state of the network and all other users, which could in turn affect the utility of the original user in subsequent times. Therefore, it is important to consider not only the instantaneous multimedia quality, but also how the immediate adaptation impacts the long-run expected quality in future.

4. **Not explicitly addressing fairness**
Although the concept of TCP-friendliness has been well developed during past decade to avoid congestion collapse of the network in presence of (mainly non-TCP) multimedia traffic, most existing congestion control solutions for video streaming (even the so-called TCP-friendly methods) fail to provide a fair allocation of network bandwidth among competing users. Even if fairness is taken into account, it is about fair distribution of throughput, while a quality-based fairness is more desirable.

In summary, a quality-driven bandwidth allocation mechanism with an explicit decision-theoretic model of competition and/or cooperation among different users for network resources, and an optimization framework to maximize a foresighted expected utility of users while maintaining some notions of fairness among users are still missing for multimedia streaming applications.

1.2 Approach

In this research, we propose a number of optimization formulation for the problem of network bandwidth allocation for video streaming applications. First, in the context of *Network Utility Maximization* (NUM), we discard the conventional but unrealistic assumption of concavity for utility functions. This results in a much more realistic optimization problem, which is not easy to solve though. An approximation algorithm for *Sigmoidal Programming* (SP) is utilized to solve the resulting nonconvex optimization problem, called NUM-SP.

We then take a *collaborative decision-theoretic approach* to the problem of rate adaptation among multiple video streaming sessions, and design a multi-objective foresighted optimization model for network resource allocation. Optimal sequential decision making under uncertainty have been extensively studied in artificial intelligence [17] [18] and stochastic control [19] literature. The basic theoretical foundations of this area are the concept of *state* and the *Markov property* –postulating that the future states of the stochastic process depend only on the present state, not on the past history of events. *Markov Decision Process* (MDP) [19] models decision problems under uncertainty when the full state information is available. In many real world problems this is not the case and only incomplete state information might be observable. *Partially Observable Markov Decision Process* (POMDP) [20] provides a powerful modeling framework for such problems.

Our adaptive video streaming problem in the case of conventional IP networks, such as the Internet, is a decision making problem with Markov property and partially observable information about the network state. However, since there are several active decision-makers interacting with the network, a decentralized or multi-agent modeling tool would be required. In this research, we attempt to address all of the shortcomings mentioned in Section 1.1 and target a quality-driven fairness-aware end-to-end congestion control and bandwidth sharing mechanism. We propose a decision-theoretic model, called *Decentralized Partially Observable Markov Decision Process* (Dec-POMDP), to formulate the interaction of multiple concurrent video streaming sessions over the Internet. Aiming at maximizing the perceived quality of end-users while maintaining fairness in network bandwidth allocation, we employ a QoE model and introduce a *social welfare* function by combining the main objectives of *efficiency* and *fairness*. The solution of the proposed multi-agent decision process provides an optimal policy for all network users to adapt their streaming rates in the best interests of the entire network, leading to an optimum fair distribution of QoE among users. We evaluate the performance of this rate adaptation scheme through simulations, showing its advantages over the TFRC.

We further look at the same problem in an SDN-enabled wireless network, where all mobile users of the network would be able to observe the network state and its congestion level. This removes the restricting assumption of partial-observability and allows us to employ fully-observable multi-agent decision process models. We use *Mutli-agent Markov Decision Process* (MMDP) framework, to formulate this dynamic interaction between network users. Due to lower computational complexity of MMDP in contrast to Dec-POMDP, a centralized planning solution becomes feasible in this case. We compare the performance of the proposed resource optimizer to two other representative existing rate adaptation methods, PANDA and FESTIVE.

1.3 Contributions

The scientific contributions of this thesis could be summarized as:

- Shifting the video rate adaptation paradigm from the classical well-established rate-distortion optimization to subjective QoE maximization. Our framework can model different types of multimedia users with different utility functions, in the form of predictive models of their subjective QoE, instead of objective measures of video quality.

- Proposing a stochastic foresighted optimization framework for bandwidth sharing and congestion avoidance as opposed to existing deterministic reactive rate control schemes and myopic optimization methods.

- Introducing a novel concept of *Social Utility Maximization* (SUM), as opposed to the well-known *Network Utility Maximization* (NUM), by explicitly incorporating fairness into the objective function of the rate allocation optimization problem. We design a social welfare function by integrating efficiency and fairness into a single performance index.

- Proposing two multi-agent decision process models for collaborative bandwidth allocation among concurrent video streaming users:
  1) Dec-POMDP model for conventional IP networks, leading to a distributed *learning* solution, and
  2) MMDP model for SDN-enabled wireless networks, leading to a centralized *planning* solution.

- Experimental performance evaluation of the proposed methods, validating the improvements in terms of QoE-efficiency and QoE-fairness.

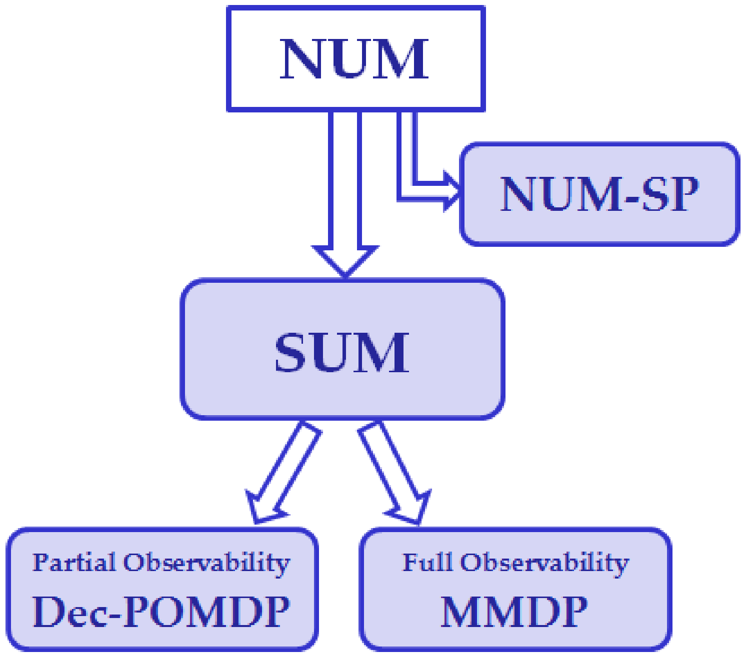

Figure 1.1: A flow diagram of the contributions of this thesis

As illustrated in Figure 1.1, both research directions originate from the well-known NUM framework; NUM-SP deals with the sigmoidal utility of multimedia traffic (Chapter 4) and SUM incorporates the fairness objective into the network resource optimization problem (Chapter 5). The proposed SUM framework is applied to two different network settings with partial and full observability of the network congestion level, resulting in development of Dec-POMDP (Chapter 6) and MMDP (Chapter 7) decision process models, respectively.

1.4 Publications

(J3) M. Hemmati, S. Shirmohammadi, "QoE-Fair Allocation of Network Bandwidth in Rate-Adaptive Video Streaming: A Multi-Agent Reinforcement Learning Approach," submitted to *ACM Transactions on Multimedia Computing, Communications and Applications.*

- The contents of this paper will appear in Chapter 6.

(J2) M. Hemmati, S. Shirmohammadi, A. Yassine, "A Multi-Agent Decision Process Approach to QoE-Fair Adaptive Video Streaming in SDN-Enabled Wireless Networks," accepted with minor revisions for publication in *IEEE Transactions on Multimedia.*

- The contents of this paper will appear in Chapter 7.

(J1) M. Hemmati, B. McCormick, S. Shirmohammadi, "QoE-Aware Bandwidth Allocation for Video Traffic Using Sigmoidal Programming," accepted for publication in *IEEE Multimedia*.

- The contents of this paper will appear in Chapter 4.

(C3) M. Hemmati, B. McCormick, S. Shirmohammadi, "Fair and Efficient Bandwidth Allocation for Video Flows Using Sigmoidal Programming," in Proc. *IEEE International Symposium on Multimedia (ISM)*, San Jose, USA, 2016.

(C2) M. Hemmati, A. Yassine, S. Shirmohammadi, "A Dec-POMDP Model for Congestion Avoidance and Fair Allocation of Network Bandwidth in Rate-Adaptive Video Streaming," in Proc. *IEEE Symposium on Computational Intelligence for Communication Systems and Networks (CI Comms)*, Cape Town, South Africa, 2015.

(C1) M. Hemmati, A. Yassine, S. Shirmohammadi, "An Online Learning Approach to QoE-Fair Distributed Rate Allocation in Multi-User Video Streaming," in Proc. *International Conference on Signal Processing and Communication Systems (ICSPCS)*, Gold Coast, Australia, 2014.

1.5 Organization of Thesis

In next chapter we review some theoretical background from areas related to the topic of this research. It includes congestion control for multimedia streaming, NUM framework, fairness in bandwidth allocation, quality of experience, and network-assisted rate adaptation. Then, in Chapter 3, we will present a survey of the state-of-the-art in bandwidth allocation for video streaming. The related works have been reviewed in three categories based on the approach taken by the authors: MDP-based, Fairness-aware, and Network-assisted.

Chapter 4 contains the first contribution of this thesis. In this chapter, we formulate the problem of bandwidth allocation for video flows in the context of NUM framework, using sigmoidal utility functions rather than conventional but unrealistic concave functions. An approximation algorithm for Sigmoidal Programming (SP) is utilized to solve the resulting nonconvex optimization problem, called NUM-SP.

Chapter 5, lays out the conceptual foundation of the other contributions of this thesis. We first present the core components of the modeling framework utilized for tackling the problem of collaborative bandwidth allocation for video streaming in order to achieve fairness in users' QoE. Then a *social welfare* function is developed to capture both fairness and efficiency objectives at the same time and the novel framework of Social Utility Maximization (SUM) is presented. Subsequently, a formal mathematical description of the proposed multi-agent modeling frameworks would be presented for two different cases of the rate-adaptive video streaming problem: conventional IP networks (Chapter 6) and SDN-enabled wireless networks (Chapter 7). A summary of obtained results and some directions for future research in this area appear in Chapter 8, which concludes this thesis.

Chapter 2

Background

In this chapter we will provide a review of theoretical backgrounds used in formulating the problem of fair bandwidth allocation in video streaming. We start with the general topic of congestion control for multimedia streaming and then describe the mathematical framework of Network Utility Maximization, which formulates the congestion control and rate allocation question into a distrusted optimization problem. Since the main theme of this thesis is QoE-fairness, we provide an overview of the concept of fairness in bandwidth sharing and different ways of modeling QoE. MDP-based optimization and decision making under uncertainty, as the cornerstones of the proposed methods in this thesis, are reviewed with an emphasis on two multi-agent decision processes, namely Dec-POMDP and MMDP. We conclude this chapter with a brief introduction of network-assisted rate adaption using SDN.

2.1 Congestion Control for Multimedia Streaming

The current Internet congestion control is performed by TCP [11], best suited for data traffic, which is insensitive to rate fluctuations. Congestion in the Internet is controlled by halving the sending rate each time a single packet loss is experienced. TCP assumes that packet losses happen only due to the congestion in the network. Congestion control actions are performed at the network end-hosts, with intermediate Internet routers only indicating the congestion by dropping or marking packets that arrive at a congested buffer. This mechanism has been very successful, producing a highly stable and scalable network.

However, the prevalence of multimedia applications during past decade has brought the necessity for a change in many of the Internet design concepts. Real-time multimedia traffic has stringent requirements on delay, jitter and packet loss, and guaranteed throughput with limited fluctuations. Multimedia applications can broadly be classified into three classes: on-demand streaming of stored content, live streaming and real-time interactive multimedia. Multimedia traffic in the current Internet can be transported over either TCP or UDP. UDP does not perform any congestion control. Multimedia traffic is sent over UDP at a constant rate equal to the drain rate at the receiver. This could result in stability and fairness issues when other traffic flows are competing for shared network resources.

TCP, on the other hand, does perform congestion control, and thus it creates large fluctuations in the fill rate in the receiver buffer. This is far from optimal for the multimedia traffic, since a typical video traffic flow is highly sensitive to sudden and large rate changes. Although TCP provides a reliable ordered delivery of the traffic, multimedia applications generally place an emphasis on timeliness over reliability. As such, there has been a broad consensus among researchers that TCP is not well-suited to multimedia traffic [21]. To this end, we have been witnessing a large number of proposals of congestion control mechanisms for multimedia traffic that tries to reconcile on one hand the requirements imposed by the multimedia applications in terms of jitter, latency and throughput and on the other hand the requirement from the Internet to ascertain that the traffic injected into the network competes fairly with other traffic handled by TCP congestion control. The solution which was advocated by the Internet Engineering Task Force (IETF) is that all applications which produce long lived flows should mimic the behavior of a TCP source. We say that such applications are "*TCP friendly*". In other words, all applications, except short transactions, should behave, from a traffic point of view, as a TCP source.

In this section, we provide a brief review of the existing literature on congestion control for multimedia applications. There are two main approaches into providing congestion control for multimedia streaming: *window-based* and *rate-based*. Window-based methods increase their sending rates as a result of the successful transmission of a window of packets, and decrease the sending rate upon the detection of a packet loss event. Rate-based congestion control attempts to smooth the transmission rate by estimating the available bandwidth in some other way. A large subset of rate-based methods is called *equation-based* rate control. In this scheme the sending rate is controlled by an equation which estimates the allowed sending rate based on feedback from the receiver. The trade-off usually is that rate-based mechanisms are less aggressive in increasing their sending rate, but in case of packet losses they lower their sending rate slower than window-based approaches. For that reason they are often called slowly-responsive algorithms.

2.1.1 Window-based Methods

TCP uses the Additive Increase Multiplicative Decrease (AIMD) algorithm to adjust its congestion window size. The size of the window is increased upon a successful transmission of a window of packets, and decreased upon a detection of a packet loss. AIMD rules could be described as follows, where w_t stands for the congestion window size at time t, whereas $\alpha > 0$ and $0 < \beta < 1$ are constants (in TCP Reno $\alpha = 1$ and $\beta = 0.5$):

$$w_{t+1} = \begin{cases} w_t + \alpha, & \text{if } success \\ w_t - \beta . w_t, & \text{if } fail. \end{cases} \tag{2.1}$$

Motivated in part by the needs of streaming audio and video applications, for which AIMD's drastic reduction in transmission rate upon each packet loss is problematic, a class of nonlinear congestion control algorithm called *binomial algorithms* were introduced in [22]. Binomial algorithms generalize the AIMD rules in the following simple manner:

$$w_{t+1} = \begin{cases} w_t + \alpha / w_t^k, & \text{if } success \\ w_t - \beta . w_t^l, & \text{if } fail, \end{cases} \tag{2.2}$$

using the addition of two algebraic terms with different exponents k and l.

These rules generalize the class of all linear control algorithms. For $k = 0, l = 1$, we get AIMD; for $k = -1, l = 1$, we get MIMD (Multiplicative Increase Multiplicative Decrease, used by slow start in TCP); for $k = -1, l = 0$, we get MIAD; and for $k = 0, l = 0$ we get AIAD, thereby covering the entire class of linear algorithms. Varying the values of k and l will result in the so-called binomial algorithms, which were shown to be TCP-friendly for suitable values of α and β if and only if $k + l = 1$ and $l \leq 1$ [23]. Figure 2.1 summarizes the qualitative features of binomial control algorithms in the (k, l) space, including the points where it corresponds to the four linear algorithms, the line segment where it is TCP-friendly, and the regions where it is more and less aggressive than TCP AIMD.

2.1.2 Rate-based Methods

Since traditional window-based approaches typically halve the sending rate in case of congestion, the effect for real-time streams is quite severe. A solution to this is a rate-based approach to congestion control. Instead of the window-based transmission model, the sending rate is controlled by an estimation of the available bandwidth. By using the estimation,

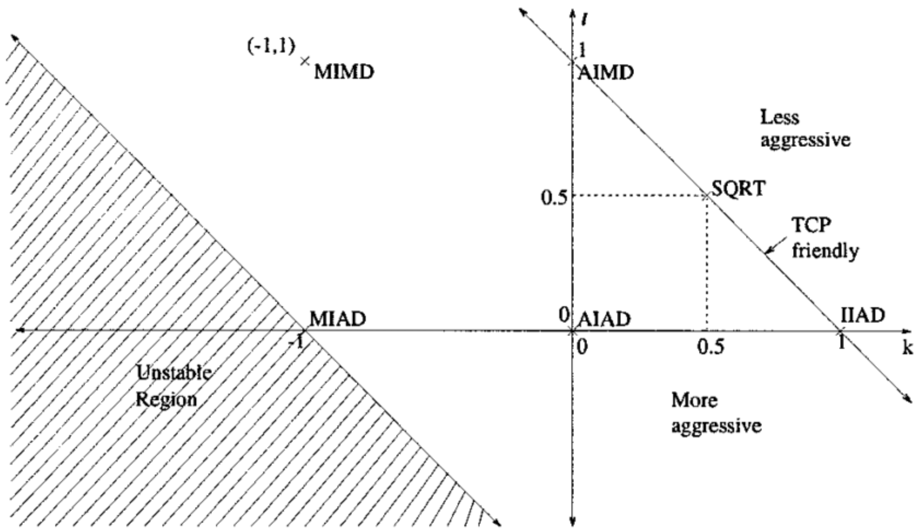

Figure 2.1: The (k, l) space of binomial control algorithms, with $k + l = 1$ line showing the set of TCP-friendly rules

the sender attempts to keep the sending rate steadier. The tradeoff is that the sender does not increase its sending rate as aggressively as TCP.

Various approaches have been taken to create a mechanism or a protocol, for providing a smooth transmission rate which multimedia streams require. An early work into providing multimedia streaming in wireline networks is Rate Adaptation Protocol (RAP) [24]. It behaves similarly to TCP as it implements an AIMD algorithm, however it abandons the window-based congestion control.

Rate-based approaches attempt to remove abrupt changes in the sending rate whilst still behaving as a good network citizen i.e. not negatively affecting TCP performance. In equation-based control, an equation is used to limit the sender transmission rate. The primary goal of equation-based approach is to avoid aggressively finding and using available bandwidth and instead maintain a steady sending rate while still being responsive to congestion [25].

TCP-Friendly Rate Control (TFRC) was developed [25], and later standardized [12], as an equation-based congestion control method for unicast traffic. In this scheme, receiver reports feedback to the sender at least once per round-trip time (RTT) if it has received packets in that interval. Accordingly, sender will reduce its sending rate in case it has not received feedback after several RTTs. Ultimately it will stop sending altogether. Adhering to these policies results in a mechanism that provides smoother throughput for multimedia applications in wired networks.

For the control equation, TFRC uses the refined TCP throughput model [26]:

$$B(p) = \frac{\ell}{RTT\sqrt{\frac{2p}{3}} + T_0(3\sqrt{\frac{3p}{8}})p(1+32p^2)} \tag{2.3}$$

where $B(p)$ stands for the upper bound of the sending rate in bytes/sec, ℓ is the packet size, RTT is the round-trip time, p is the steady-state loss event rate and T_0 is the TCP retransmit timeout value. A loss event rate is not exactly the same as packet loss rate; a loss event shows the number of packets lost in a single round-trip time. The loss event rate is calculated at the receiver end and sent back to the sender. Then, the sender will calculate a smoothed loss event rate to be used in the equation. This smoothing is done in order to avoid reacting violently to a single unrepresentative loss event rate reported by the receiver. Other than the loss event rate, all the other factors in the equation can be measured at the sender side.

TFRC has emerged as the dominant mechanism in literature and the standard protocol to provide smooth and predictable throughput for multimedia applications. The latest version is described in RFC 5348 [12].

Some other protocols are designed to be multimedia streaming friendly from the start and these include Datagram Congestion Control Protocol (DCCP) [27], Real-time Transport Protocol (RTP) [28]. DCCP is an unreliable transport protocol like UDP but incorporates end-to end congestion control. It implements a congestion-controlled, unreliable flow of datagram for multimedia streaming applications.

2.2 NUM Framework

The Network Utility Maximization (NUM) was proposed by F. Kelly [29] as a mathematical framework for rate control and bandwidth allocation in networks. The key issue is to how available bandwidth within the network should be shared among competing flows of (elastic) traffic. Under some conditions for the utility functions of the network users, NUM framework formulates this problem as a convex optimization problem.

Consider a communication network with L links and S sources (i.e., users). Each link l has a fixed capacity of c_l, and each source s is assumed to transmit only one flow at a rate of x_s, and has a utility function $U_s(x_s)$. We denote the set of links used in the path of source s by $\mathcal{L}(s)$, and the set of sources/flows sharing link l by $\mathcal{S}(l)$. The basic version of NUM could be described as follows:

$$
\begin{aligned}
\underset{\mathbf{x}\in\mathbb{R}^S}{\text{maximize}} \quad & \sum_{s=1}^{S} U_s(x_s) \\
\text{subject to} \quad & \sum_{s\in\mathcal{S}(l)} x_s \le c_l, \quad \forall l, \\
& \mathbf{x} \ge 0.
\end{aligned}
\tag{2.4}
$$

This means maximizing total utility of all sources over the set of nonnegative values of source rates $\mathbf{x} \in \mathbb{R}^S$, subject to linear flow constraints on all links, which have limited capacities. If the utility functions $U_s(\cdot)$ are increasing with rate and strictly concave, the above optimization problem would be a convex one with many desired characteristics; most importantly, local optimum is also a global optimum, and the duality gap is zero [30]. Zero duality gap allows for employing decomposition of the dual problem for constructing a distributed solution for NUM, where each source only needs its own utility function and each link its own capacity. This results in a standard distributed price-based rate allocation and congestion control algorithm for the network, which also provides stability and some form of fairness.

2.3 Fairness in Bandwidth Sharing

A fundamental question in network design and optimization is how available bandwidth should be shared among competing users of a network? In fact, the objective of congestion control is to provide both efficiency and some form of fairness among all users. As explained in previous section, NUM framework provides a distributed rate control solution which allows to reconcile potentially conflicting notions of fairness and efficiency.

In this section, we are going to review the concept of fairness in network resource allocation; to see what exactly fairness is; how it could be measured; and how it could be achieved among network users.

2.3.1 Meanings and Definitions of Fairness

Fairness, as referred to in data communication networks context, is generally attributed to resource sharing or allocation. The consequence of an unfair resource allocation among different network users may lead to resource starvation, wastage or redundant allocation. For different researchers, it is rather difficult to agree on a single definition of fairness since it is subjective [31]. When we consider rational individuals, each individual evaluates the share of resources they received and compare them with others in the system from their own points of view. Consequently, the definition of fairness or any effort to define fairness is influenced by the value ascribed to the resources by the designer of the system or by the individuals of the system.

Accordingly, there exist several notions of fairness in the literature. Two of the most representative interpretations of fairness are *max-min* and *proportional* fairness.

Max-Min Fairness

A feasible allocation of resource $\mathbf{x} = (x_1, x_2, ..., x_n)$ to n users is max-min fair if for each user i, x_i cannot be increased (while maintaining the feasibility) without decreasing x_j, where $x_j \leq x_i, (i \neq j)$ [32]. In other words, a system reaches max-min fairness, if it cannot increase any individual's share without decreasing another individual's share which is already less than the previous ones. Max-min fairness, which is also called *bottleneck optimality*, has been widely studied and implemented in many applications such as flow control [33], packet scheduling [34], [35], routing [36], and other wireless network applications [37], [38], [39].

It is worth mentioning that weights can also be introduced into the max-min fairness model. Weighted measures are more flexible in achieving fairness, but pose other issues such as the assignment of the weight set.

Proportional Fairness

An allocation $\mathbf{x} = (x_1, x_2, ..., x_n)$ of a resource with finite capacity of C is said to be proportionally-fair [29], if it is feasible, that is $\forall i,\ x_i \geq 0$ and $\sum_{i=1}^{n} x_i \leq C$, and if for any other feasible allocation $\mathbf{x}^*$, the aggregate of proportional changes is not positive:

$$\sum_{i=1}^{n} \frac{x_i^* - x_i}{x_i} \leq 0. \tag{2.5}$$

2.3.2 Quantifying and Measuring Fairness

One approach to quantify the degree of fairness associated with an allocation vector $\mathbf{x} = (x_1, x_2, ..., x_n) \in \mathbb{R}_+^n$ is through a fairness measure, which is a function f that maps $\mathbf{x}$ into a real number. Various fairness measures have been proposed throughout the years, e.g., in [40], [41], [42], [43]. These range from simple ones, e.g., the ratio between the smallest and the largest entries of $\mathbf{x}$, to more sophisticated functions, e.g., *Jain's index* [44] and the *entropy function*. Some of these fairness measures map $\mathbf{x}$ to normalized ranges between

0 and 1, where 0 denotes the minimum fairness, 1 denotes the maximum fairness (often corresponding to an $\mathbf{x}$ where all x_i are the same) and a larger value indicates more fairness. For example, min-max ratio [1] is given by the maximum ratio of any two user's resource allocation, while Jain's index computes a normalized square mean.

An alternative approach that has gained attention in the networking research community since [29], [45] is the optimization-theoretic approach of α-fairness and the associated utility maximization. Given a set of feasible allocations, a maximizer of the α-fair utility function satisfies the definition of α-fairness. Two well-known examples are as follows: a maximizer of the log utility function ($\alpha = 1$) is proportionally fair, and a maximizer of the α-fair utility function as $\alpha \to \infty$ is max-min fair.

Clearly, these two approaches for quantifying fairness are different. On the one hand, α-fair utility functions are continuous and strictly increasing in each entry of $\mathbf{x}$, thus its maximization results in Pareto optimal resource allocations. On the other hand, scale-invariant fairness measures (ones that map $\mathbf{x}$ to the same value as a normalized $\mathbf{x}$) are unaffected by the magnitude of $\mathbf{x}$, and an allocation that does not use all the resources can be as fair as one that does. As an attempt to unify these two approaches, [46] develops an axiomatic theory of fairness in netwrok resource allocation. It introduces a set of five intuitive axioms that lead to a useful family of fairness measures. These axioms are: the axiom of continuity, of homogeneity, of asymptotic saturation, of irrelevance of partition, and of monotonicity.

The development of an axiomatic theory of fairness also contributes to making connection between the basic concepts of efficiency and fairness. By removing the axiom of homogeneity, an alternative set of four axioms are proposed in [46], which allows efficiency of resource allocation be jointly captured in the fairness measure. We will use this interesting feature to design one of the key components of our proposed framework, described in Chapter 5.

2.4 Quality of Experience

Quality of Experience (QoE) is the perceptual quality of a service from the end-users' perspective. A key aspect in all forms of video delivery, including video streaming, is the need to optimize user-perceived QoE. Traditionally, QoE is obtained from subjective test, where human viewers evaluate the quality of test videos under a laboratory environment. In subjective tests as well as multimedia communications, *Mean Opinion Score* (MOS) provides a numerical measure of the perceived quality from the users' perspective. It is expressed as a single number in the range 1 to 5, where 1 corresponds to the lowest perceived quality, and 5 to the highest. Since subjective test is a costly, time consuming, and offline way of measuring QoE, objective quality models have been developed to predict QoE, in terms of MOS, based on available objective parameters, such as video coding and compression schemes and network Quality of Service (QoS) parameters [47].

In multimedia communications, the MOS provides a numerical measure of the perceived quality from the users' perspective of received media at the destination. It is expressed as a single number in the range 1 to 5, where 1 corresponds to the lowest perceived quality, and 5 to the highest.

MOS	Quality	Impairment
5	Excellent	Imperceptible
4	Good	Perceptible but not annoying
3	Fair	Slightly annoying
2	Poor	Annoying
1	Bad	Very annoying

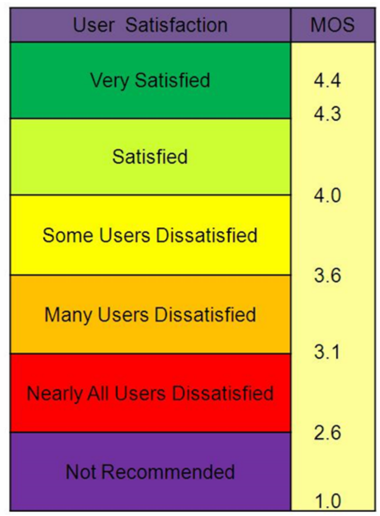

Figure 2.2: MOS Scale for Subjective Video Quality Assessment

Designing good QoE models for video streaming is an active area of research [48]. While a higher video quality is expected to improve QoE, existing literature on QoE of video streaming provides evidence that frequent switches or sudden changes in video quality or bitrate result in user's dissatisfaction and reduces QoE [49]. To this end, [50] introduced the notion of penalty for temporal variability of quality in modeling QoE. However, video freeze or stalling, which happens due to rebuffering of video player, is the most harmful factor in QoE degradation [51]: a single rebuffering event has three times the impact of a bitrate change. Another reported finding of [51] is that rebuffering incur abandonment rates six times higher than start-up latency. For a comprehensive survey of the QoE influence factors in HTTP-based adaptive video streaming see [49].

One of the well-accepted QoE models for adaptive streaming provides a prediction or estimation of MOS through a linear combination of the average quality of video segments over a streaming session, the variation of video qualities and the effect of rebuffering:

$$QoE = MOS_{est} = \alpha.\mu - \beta.\sigma - \gamma.\phi + \delta \tag{2.6}$$

where μ is the average quality of video segments, σ is the standard deviation of video qualities, ϕ is a measure of rebuffering or freezes, and α, β, γ, and δ are tunable parameters [52].

2.5 Decision Making under Uncertainty

In most real-world problems, there are many sources of uncertainty in the environment and problem description and yet an entity in charge of decision and control needs to take actions to perform a task or achieve a goal. Stochastic modeling techniques such as *Markov Decision Process* (MDP) [18], are the standard mathematical tools for decision making under uncertainty.

An MDP is a model of an agent interacting with a dynamic world. The agent takes as input the *state* of the world and makes a choice about his *action*, which in turn results in some *reward* and a probabilistic transition to a new state of the world. Formally speaking, a Markov Decision Process is defined as a tuple $\langle \mathcal{S}, \mathcal{A}, T, R, h \rangle$, where

- $\mathcal{S}$ is the finite set of states $\mathbf{s}$ of the world (environment);
- $\mathcal{A}$ is the finite set of actions;
- T is the *state transition function* that provides $P(\mathbf{s}'|\mathbf{s}, a)$, the probability of transition to a next state $\mathbf{s}'$ given that the agent starts at state $\mathbf{s}$ and takes action a;
- R is the immediate reward function which depends on the state of the environment and action of the agent: $R(\mathbf{s}, a)$ providing a real number;
- h is the time horizon of the problem, which could be either finite or infinite.

In this model, the next state and reward depend only on the current state and the action taken; not the additional previous states. This is known as the Markov property, stating that the conditional probability distribution of future states of the process (conditional on both past and present states) depends only upon the present state, not on the sequence of events that preceded it.

MDP framework has been widely used to formulate optimal decision making and control problems. When the full state information is available, dynamic programming could be utilized to find the optimal solution of MDPs. However, full state information is not always available in many real-world problems. *Partially Observable MDP* (POMDP) framework is used to tackle decision problems with imperfect or unobservable state information [53, Chpater 12].

When there are more than one active decision-maker in the environment, we are dealing with a multi-agent system [54]. *Decentralized Partially Observable Markov Decision Process* (Dec-POMDP) framework is an extension of single-agent POMDP to a multi-agent cooperative setting. In a Dec-POMDP, the dynamics of the system and the objective function depend on the actions of all agents, however each agent need to make decisions based on local information and imperfect observation of the state.

2.5.1 Partially-Observable Model: Dec-POMDP

Here we provide a formal definition of the employed modeling framework for the partially-observable case, which is an extension of single-agent POMDP to a multi-agent cooperative setting. A *Decentralized Partially Observable Markov Decision Process* (Dec-POMDP) is defined as a tuple $\langle \mathcal{N}, \mathcal{S}, \mathcal{A}, \mathcal{O}, T, O, R, h, I \rangle$, where

- $\mathcal{N} = \{1, 2, \ldots, N\}$ is the set of N agents.
- $\mathcal{S}$ is the finite set of states $\mathbf{s}$ in which the environment can be.
- $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_N$ is the finite set of joint actions of all agents $\mathbf{a} = \langle a_1, \ldots, a_N \rangle$, where an individual action of an agent $n \in \mathcal{N}$ is denoted by $a_n \in \mathcal{A}_n$.
- $\mathcal{O} = \mathcal{O}_1 \times \cdots \times \mathcal{O}_N$ is the finite set of joint observations $\mathbf{o} = \langle o_1, \ldots, o_N \rangle$, where an individual observation of an agent $n \in \mathcal{N}$ is denoted by $o_n \in \mathcal{O}_n$.
- T is the transition function that provides $P(\mathbf{s}'|\mathbf{s}, \mathbf{a})$, the probability of transition to a next state $\mathbf{s}'$ given that joint action $\mathbf{a}$ is executed at state $\mathbf{s}$.
- O is the observation function that specifies $P(\mathbf{o}|\mathbf{a}, \mathbf{s}')$, the probability that the agents receive joint observation $\mathbf{o}$ of state $\mathbf{s}'$, when they reached this state through joint action $\mathbf{a}$.
- R is the common immediate reward function which depends on the state of the environment and actions of all agents. $R(\mathbf{s}, \mathbf{a})$ specifies a real number as the common reward for all agents.
- h is the time horizon of the problem, which could be either finite or infinite.
- $I \in \mathcal{P}(\mathcal{S})$ is the initial probability distribution of the state, where $\mathcal{P}(\cdot)$ denotes the set of probability distributions over its argument.

2.5.2 Optimal Solution of Dec-POMDP

A policy in a fully observable MDP is a mapping from states to actions. The policy of a user is comprised of a sequence of actions selected by the user at every time step based on the state of the environment. In selecting the actions, the agent can ignore the history because of the Markov property. In a POMDP [20], the agent does not observe the state, but it can compute a *belief* that summarizes the history and works as a Markovian signal.

In a Dec-POMDP [55], however, each agent will only have access to its individual actions and observations during execution and there is no method known to summarize this individual history. It is not possible to maintain and update an individual belief in the same way as in a POMDP, because the transition and observation functions are specified in terms of joint actions and observations. The consequence of this lack of access to a Markovian signal is that planning for Dec-POMDPs involves searching the space of tuples of individual policies that map full-length individual histories to actions.

Solving a Dec-POMDP is a really challenging task. In fact, it is known that the problem of finding the optimal solution for a finite-horizon Dec-POMDP with even only two agents is NEXP-complete [56]. In practice, this means that solving a Dec-POMDP takes doubly exponential time in the worst case. Moreover, efficient approximation of Dec-POMDP is not easily possible, and even finding an ϵ-approximate solution is NEXP-complete [57].

Since the number of joint policies in a Dec-POMDP grows exponentially with the number of possible observations, a brute force search would only be suitable for very small problems. Therefore, so much effort has been spent by researchers during last decade to create efficient methods for finding exact or approximate solution of Dec-POMDP. [58] provides a recent survey of the existing methods, including approaches based on dynamic programming [59] [60] [61] [62], as well as heuristic search algorithms [63] [64] [65].

For our work, we use the *Joint Equilibrium based Search for Policies* (JESP) [66], which is guaranteed to find a locally optimal joint policy. It relies on a procedure called *alternating maximization*, that computes a maximizing policy for one agent at a time, while keeping the policies of the other agents fixed. This process is repeated until the joint policy converges to a Nash equilibrium: a tuple of policies such that for each agent's policy is a best response to the policies employed by the other agents. JESP uses a dynamic programming approach to compute the best-response policy for a selected agent, using a reformulation of the problem as an augmented POMDP by fixing the other agent's policies.

Alternating Maximization

One of the most practical methods introduced for solving Dec-POMDP is *Joint Equilibrium based Search for Policies* (JESP) [66], which is guaranteed to find a locally optimal joint policy. It relies on a procedure called *alternating maximization*, that computes a maximizing policy for one agent at a time, while keeping the policies of the other agents fixed. This process is repeated until the joint policy converges to a Nash equilibrium: a tuple of policies such that for each agent's policy is a best response to the policies employed by the other agents. JESP uses a dynamic programming approach to compute the best-response policy for a selected agent, using a reformulation of the problem as an augmented POMDP by fixing the other agent's policies.

Optimal Value Function

Another approach for solving Dec-POMDP is more in line with conventional methods for single agent MDPs and POMDPs: identifying an optimal value function Q^* that can be used to derive an optimal policy. Even though computation of Q^* itself is intractable, the insight it provides is valuable [55]. In particular, it has a clear relation with the other two dominant approaches to solving Dec-POMDPs: the backward and the forward approach which will be explained in the following.

Backward Approach: Dynamic Programming

The core idea of Dynamic Programming (DP) for Dec-POMDP [59] is to incrementally construct sets of longer sub-tree policies for the agents. Unfortunately, these sets grow doubly exponentially with the horizon of the problem. To counter this source of intractability, *pruning* dominated sub-tree policies is required. DP iterates over agents until no further pruning is possible, a procedure known as *iterated elimination of dominated policies* [67]. In practice, the pruning step in DP often is not able to sufficiently reduce the maintained sets to make the approach tractable for larger problems. Therefore, a number of variants of this method have also been proposed in order to improve the pruning process. Point-based DP [60], memory-bounded DP [61], and DP-LPC [62] are some of the notable variations of this approach.

Forward Approach: Heuristic Search

We pointed out that computing Q^* is not tractable for Dec-POMDP. The basic idea to overcome this problem in *forward approach* is to use an approximation $\hat{Q}$ instead. One popular method in this category, known as MAA* [63], approximates the optimal value function with the value function of the 'underlying MDP': the MDP with the same transition and reward function as the Dec-POMDP. In a slightly different variation of this method, the value function of the 'underlying POMDP'is used as an approximation. A generalized framework for MAA*, called GMAA*, was proposed by [64], and subsequently improved for faster planning leading to GMAA*-ICE method [65].

2.5.3 Fully Observable Model: MMDP

Unlike Dec-POMDP, in a *Multi-agent Markov Decision Process* (MMDP), each agent is able to observe the true state of the environment, making the problem fully observable. MMDP is basically modeling a group of agents who are collectively controlling a process and share a common objective or reward. It was first studied by [68] in the context of coordination problem among multiple agents. Figure 2.3 captures the relation between different frameworks for decision making under uncertainty. An MMDP is formally defined as a tuple $\langle \mathcal{N}, \mathcal{S}, \mathcal{A}, T, R, h \rangle$, where

- $\mathcal{N} = \{1, 2, \ldots, N\}$ is the set of N agents.
- $\mathcal{S}$ is the finite set of states $\mathbf{s}$ in which the environment can be. It is completely observable by individual agents.
- $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_N$ is the finite set of joint actions of all agents $\mathbf{a} = \langle a_1, \ldots, a_N \rangle$, where an individual action of an agent $n \in \mathcal{N}$ is denoted by $a_n \in \mathcal{A}_n$.
- T is the transition function that provides $P(\mathbf{s}'|\mathbf{s}, \mathbf{a})$, the probability of transition to a next state $\mathbf{s}'$ given that joint action $\mathbf{a}$ is executed at state $\mathbf{s}$.
- R is the common reward function which depends on the state of the environment and actions of all agents. $R(\mathbf{s}, \mathbf{a})$ specifies a real number as the common reward for all agents.
- h is the time horizon of the problem, which could be either finite or infinite.

At each time step, also known as *stage*, the agents simultaneously take an action. The resulting joint action provides a common reward based on the current state. It also causes a stochastic transition to the next state, of which a joint observation is emitted by the environment and each agent observes its own component.

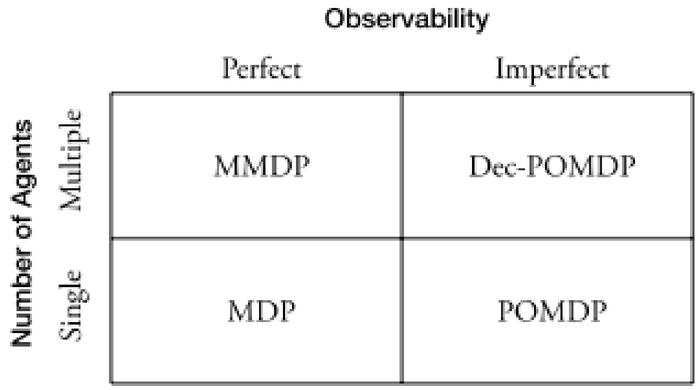

Figure 2.3: Different frameworks for decision making under uncertainty

2.5.4 Optimal Solution of MMDP

Finding a solution for MMDP is as computationally complex as solving MDP; they are both P-Complete [58]. Taking the joint action space of an MMDP to be the set of basic actions, it could be viewed as a standard single-agent MDP [69]. Specifically, since there is a single reward function, the agents do not have competing interests; so any course of action is equally good for all. Therefore, the optimal joint policies over joint action space could be computed by solving the standard MDP using an algorithm like value iteration [18], revised by adding coordination mechanisms to ensure agent policies are consistent with each other. This approach to solving an MMDP, assuming complete knowledge or availability of the model, is often called *planning*. In many practical cases, the transition function of MMDP is not known, and we need to resort to model-free reinforcement *learning* approach for finding the optimal policies.

2.6 SDN and Network-Assisted Rate Adaptation

Software-Defined Networking (SDN) is an emerging networking paradigm with the promise of changing the limitations of current network infrastructure and providing more flexible and programmable networks [70] and [71]. Traditionally, a network element (NE) such as a router does two main things when it receives a packet: first, based on the packet's header, the NE decides how to route the packet, using some type of routing algorithm. Second, it forwards the packet to the next appropriate NE (hop) based on the route. SDN, on the other hand, physically separates a network's control plane from its forwarding plane. Control plane refers to the intelligence that decides how the packets should move throughout the network (routing algorithm, among others), while forwarding plane refers to the physical forwarding of the incoming packets to the next hop in the network. Under SDN, NEs no longer make the routing decisions, but simply forward packets according to instructions received from the control plane, which now monitors and controls everything. This leads to many advantages and features, the discussion of which is also beyond the scope of this paper. We suffice to mention that the feature of SDN we use in our work is its ability to provide a centralized global network view: all element and link metrics (delay, bandwidth, connectivity, topology, etc.) can be measured dynamically [72]. This centralized global network view has already been taken advantage of by existing literature for wireless networks. For instance, [73] proposes a Software-Defined Mobile Networks (SDMN) architecture with programmability, flexibility, and openness features backed by experimental testing. In a more recent work, [74] addresses the challenges of implementing intelligent content delivery on SDN-enabled wireless access systems such as LTE and Wi-Fi, by dynamically controlling network traffic over WANs from edge nodes of wireless networks.

One approach to address the challenges of congestion control in general, and multimedia rate adaptation in particular, is to leverage the logically centralized control plane offered by SDN. In this approach, which has gained a lot of interest over the past few years both in the academia and industry, centralized SDN-based dynamic resource allocation schemes are either replacing the distributed rate control and adaptation protocols, or becoming an important source of assistance in the process of rate adaptation. The concept of network-assisted rate adaptation is one of the building blocks of the methods proposed in this thesis.

Chapter 3

Related Work

In this chapter, we will review the state of the art in bandwidth allocation for video streaming applications. We categorize the most recent related work in this area into three groups. First MDP-based rate adaptation algorithms will be reviewed. We consider both server-side and client-side adaptation schemes. The second group consists of a few published works in the area of fair bandwidth allocation. We will distinguish between rate-fairness and QoE-fairness. Finally, network-assisted rate adaptation mechanisms are covered in the third section.

3.1 MDP-based Approaches

Markov Decision Process has been employed by many researchers over the past few years to formulate the optimal decision making problem in video rate adaptation. It provides a foresighted optimization framework under stochastic uncertainties posed by the nature of communication networks. Before the advent and popularity of HAS such as DASH, server-side rate adaptation problem was at the center of published researches. In [16], the authors propose a quality-centric congestion control scheme for multimedia streaming over wired IP networks, using a finite horizon MDP to determine the optimal congestion control policy that maximizes the long-term video quality, while adhering to some form of TCP-friendliness constraint. This work assumes full observability of the MDP state, which is defined to be comprised of the expected TCP window size (representing the network state) and the number of packets in all packet classes (representing the application state). [75] takes a partially observable approach in formulating the congestion control problem for conversational multimedia services by using a POMDP model, whose state is defined to include the source rate, which is known by the agent, and the packet loss rate, which cannot be directly observed. Users need to infer the congestion status of the bottleneck links using observations and QoE feedbacks, which are provided in terms of MOS. [76] proposes a cross-layer control mechanism, formulated as an MDP, to stream scalable videos to mobile receivers. Its goal is to maximize the quality of the received video while accounting for the variations of the characteristics of the transmitted content and of the channel, by filtering the scalability layers of the encoded video.

MDP-based optimization has also been applied to the client-side rate adaptation area, including HTTP-based adaptive streaming. [77] designed a Q-learning-based HAS client to dynamically learn the best actions corresponding to the actual network environment. [8] investigated HAS in vehicular environment using an MDP formulation of the problem and demonstrated its superior performance in terms of reducing playback deadline miss compared to non-MDP solutions. MDP-based rate adaptation for DASH was also studied in [9], aiming to maximize the users' QoE under time-varying channel conditions. This work takes into account key factors impacting the visual quality, including video playback quality, video rate switching frequency and amplitude, buffer overflow/underflow, and buffer occupancy. Another online learning adaptation strategy for DASH clients was proposed by [6] based on an MDP optimization. The authors introduced a penalty function into the reward function to penalize the system for rebuffering events as well as moving away from a safe buffer level.

3.2 Fairness-aware Approaches

Flow rate fairness was, and maybe still is, the predominant way of defining fairness in network resource allocation. A decade ago, [78] pointed out the shortcomings of this notion, which turns out to be even impractical. This work addressed the related issues of fairness and resource accountability of the Internet, in an attempt to dismantle the so-called "religion" of flow rate fairness. Despite such attempts, the majority of the research work in the area of fair allocation of network resources, does focus on rate-fairness. For instance, [79] proposes a rate adaptation scheme, called FESTIVE, which tries to strike an appropriate balance among three performance metrics for HAS video streaming: fairness, efficiency, and stability. However, the notion of fairness employed in FESTIVE is all about equal share of bandwidth and enjoying the same amount of bitrate for different flows.

[80] is another recent work aimed at realizing trade-offs among mean, variability, and fairness in resource allocation, which introduces a generalization of NUM framework to incorporate the detrimental impact of temporal variability in a user's QoE. It's main contribution is the development of an online algorithm, which asymptotically solves the

variability-aware resource allocation problem. However, utility functions have been used in the formulation of the optimization problem, as proxies for the users' QoE. Under the assumption that these utility functions are concave and differentiable, the obtained resource allocation would be α-fair in QoE.

In [7], the authors propose a novel rate adaptation algorithm called FINEAS (Fair In-Network Enhanced Adaptive Streaming), for achieving QoE-fairness in a multi-user setting. They use an in-network system of coordination proxies to facilitate fair resource sharing among clients. As a result, QoE-fairness is achieved without explicit communication among clients and thus no significant overhead is introduced into the network.

In a highly cited paper, [81] reported that unfairness is mainly caused by the quality adaptation algorithms themselves, since they are not designed to explicitly cope with a multi-user scenario. It is shown that a relevant cause of unfairness is the temporal overlap of the ON-OFF periods of different clients, since this can lead to wrong bandwidth estimations. Based on this consideration, most of the algorithms designed to improve fairness in HAS focus on the modification of the time interval at which clients request a new segment.

While many HAS system designs aim at sharing the network bandwidth in a rate-fair manner, some recent research works are focusing on achieving a QoE-fair allocation of resources. [82] proposed a price-based controller for HAS clients that is able to reach quality fairness while preserving the main characteristics of HAS systems and with a limited support from the network devices. In [83] a new SDN-based dynamic resource allocation and management architecture for DASH is proposed, which aims to alleviate quality instability and unfair bandwidth sharing issues and improve the per-client QoE.

In [84], the authors aim at improving QoE and fairness in adaptive video streaming over wireless LTE networks. The share of radio resources is optimized according to video content characteristics, play-out buffer levels and channel conditions to deliver fair video quality and to achieve asymptotically fair play-out buffer levels among HAS clients competing for the same wireless resources in an LTE cell.

3.3 Network-assisted Approaches

One of the early works in the area of network-assisted rate adaptation is Layered Internet Video Adaptation (LIVA) [85], in which network nodes feed back virtual congestion levels to video senders to both assist bandwidth sharing and transit-loss protection. The video senders respond to such feedback by adapting the rates of encoded scalable bitstreams based on their respective video rate-distortion characteristics. Later, [86] proposed a coordinated

Internet video control plane: motivated by measurement-driven insights, they made a case for a video control plane that can use a global view of client and network conditions to dynamically optimize the video delivery in order to provide a high quality viewing experience despite an unreliable delivery infrastructure. Their analysis shows that such a control plane can potentially improve the rebuffering ratio by up to two times in the average case and by more than one order of magnitude under stress.

With the prevalence of HTTP-based adaptive streaming (HAS) and DASH [4] over the past few years, network-assisted rate adaptation schemes for DASH were also proposed and promoted both by the academy and industry. The idea of in-network quality optimization for DASH was introduced by [87]: several centralized and distributed algorithms and heuristics were proposed that allow nodes inside the network to steer the HAS client's quality selection process. The algorithms are able to enforce management policies by limiting the set of available qualities for specific clients. In another work, [88] proposed an SDN-based QoE optimization for HAS to tailor the bandwidth allocation to both content complexity of the requested video and playout buffer status of individual clients, instead of allocating bandwidth equally among competing flows.

On the industry side, MPEG is working on Server and Network Assisted DASH (SAND) [89], which provides asynchronous network-to-client and network-to-network communication of quality-related assisting information.

Most implementations of DASH players naively estimate bandwidth from a one-sided client perspective, without taking into account other devices in the network. This behavior results in unfairness and could potentially lower QoE for all clients. To counter that, [90] proposed an OpenFlow-assisted QoE fairness framework that aims to fairly maximize the QoE of multiple competing clients in a shared network environment. By leveraging a SDN's OpenFlow, the authors provide a control plane that orchestrates this functionality. Using a similar approach, [91] proposed an SDN application to monitor network conditions of streaming flow in real time and dynamically change routing paths using multi-protocol label switching (MPLS) traffic engineering (TE) to provide reliable video watching experience.

Despite the relatively rich literature on any of the three preceding topics, an all-inclusive approach to the problem of bandwidth allocation for video streaming which i) formulates it as a foresighted optimization using MDP-based models, ii) is aware of QoE-fairness among concurrent users, iii) takes advantage of the network assistance for a more efficient and fair resource allocation, and iv) most importantly, takes into account the multi-agent nature of the problem, is missing int the state-of-the-art.

Chapter 4

Bandwidth Allocation using Sigmoidal Utility Functions: NUM-SP

Network bandwidth allocation is crucial in network design, deployment, and maintenance. It deals with this fundamental question: how much bandwidth should be allocated to each user and/or flow on the network, such that we provide fairness to users while also making efficient use of network resources? Typically, the answer comes after modeling the question as an optimization problem and solving it. The most common solutions are that of, or similar to, Network Utility Maximization (NUM) [29], [92]. In NUM, users' level of satisfaction is modeled by utility functions and maximized subject to network capacity constraints, leading to a price-based distributed rate allocation. These types of methods have gained much interest during the past two decades. For example, it has been shown that the well-known TCP congestion control algorithm is actually implicitly solving a utility maximization problem [92].

Unfortunately, a common assumption behind these methods is that the utility functions for traffic flows are concave, which is critical for analytical tractability and convexity of the NUM optimization problem. However, the utility function of multimedia flows such as video streaming is not concave, but sigmoidal. Shenker [93] was the first to suggest sigmoidal functions to model end-user's utility for delay- and rate-sensitive multimedia applications. Sigmoidal functions have also been used in a number of studies [94], [95] as a model of user satisfaction or Quality of Experience (QoE) for multimedia traffic. More recently, it has been shown that the video streaming QoE of popular video services such as YouTube is indeed a sigmoidal function of bitrate [96]. This means that the optimality and applicability of current rate allocation techniques for multimedia flows are doubtful,

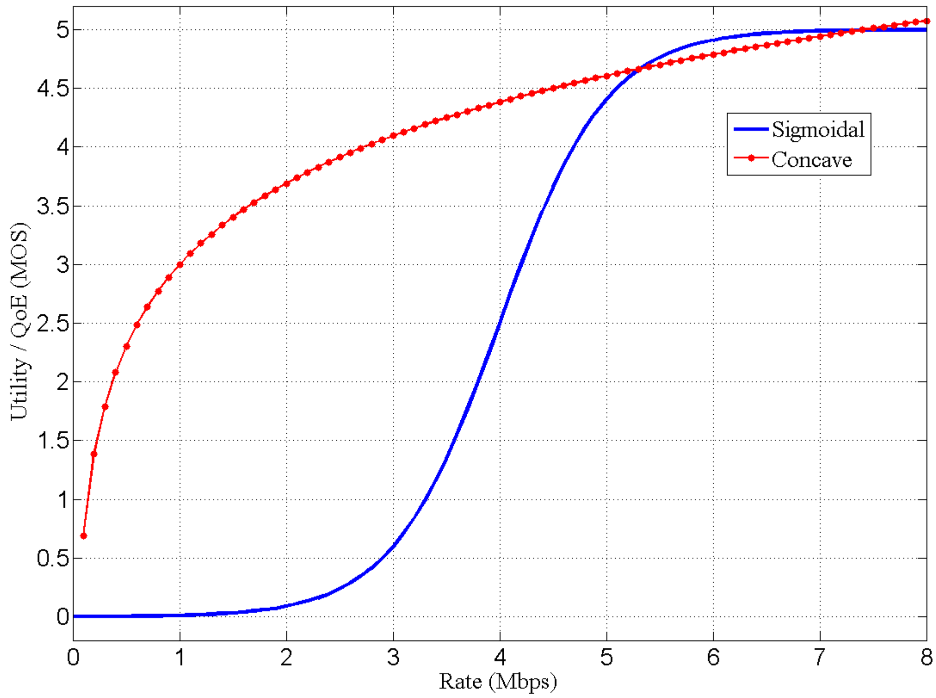

Figure 4.1: Sigmoidal vs. concave utility functions

because they use concave functions.

A single-sigmoidal is a shape that has one convex region followed by a concave one with a single inflection point separating the two regions. The intuition behind this shape is that low values of rate offer very low degree of user satisfaction, and as data rate increases, user satisfaction increases rapidly up to a turning point where the slope of increase starts to decline until a saturation appears and user satisfaction reaches maximum. Figure 4.1 shows a single-sigmoidal utility function versus a concave one.

In this chapter, we present an optimization framework, called *Network Utility Maximization with Sigmoidal Programming* (NUM-SP), for maximizing sigmoidal utility functions subject to network constraints. We formulate the problem of network bandwidth allocation of video traffic as a NUM with sigmoidal utility. This is a nonconvex optimization problem and is NP-hard to solve [97], [98]. Using recent mathematical findings on bounding the duality gap for optimization problems with separable nonconvex objective functions [99], we propose an approximate solution of the sigmoidal NUM problem that guarantees ϵ-suboptimalily. We also implemented our proposed NUM-SP, using optimization software packages in Julia [100], to compare its resulting flow assignments to two existing representative rate allocation methods: *Proportional Fair* and *Max-Min Fair*.

Finally, we present the results of extensive simulations on a range of realistic network topologies, and we show that our sigmoidal programming method outperforms the above methods in terms of average utility of users, fairness, number of satisfied users, and network capacity usage.

4.1 Background: NUM and Nonconvexity

4.1.1 NUM Framework

Consider a communication network with L links and S sources (i.e., users). Each link l has a fixed capacity of c_l, and each source s is assumed to transmit only one flow at a rate of x_s, and has a utility function $U_s(x_s)$. We denote the set of links used in the path of source s by $\mathcal{L}(s)$, and the set of sources/flows sharing link l by $\mathcal{S}(l)$. The basic version of NUM could be described as follows:

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^S}{\text{maximize}} \quad & \sum_{s=1}^{S} U_s(x_s) \\ \text{subject to} \quad & \sum_{s\in\mathcal{S}(l)} x_s \le c_l, \quad \forall l, \\ & \mathbf{x} \ge 0. \end{aligned} \tag{4.1}$$

This means maximizing total utility of all network users over the set of nonnegative values of source rates $\mathbf{x} \in \mathbb{R}^S$, subject to linear flow constraints on all links, which have limited capacities.

There are a number of basic assumptions for utility functions in NUM: they are assumed to be functions of the allocated rate (rate dependency) and only its own rate (locality), continuous and twice differentiable (smoothness), increasing with rate (monotonicity), and strictly concave. This *concavity* is critical for NUM to be a convex optimization problem and hence mathematically tractable: a local optimum is also a global optimum, and the duality gap is zero [30]. Zero duality gap implies that the minimized value of the Lagrange dual problem, which provides a decomposition structure, is equal to the maximized total utility in the primal problem (4.1). Utilizing the decomposed dual problem, a distributed solution could be constructed for NUM, where each source only needs its own utility function and each link its own capacity. This results in a distributed rate allocation and congestion control algorithm for the network.

In addition to analytical tractability, another justifications of the concavity assumption is law of diminishing marginal utility in economics, which states that the first unit of consumption of goods or services yields more utility than the second and subsequent units, with a continuing reduction for greater amounts. Similarly, following the same law, the utility function of network users should become concave as the allocated rate increases. However, experience shows that while the law *eventually* will be effective, for video streaming it doesn't do so through the entire domain [96].

In his seminal paper, Shenker [93] differentiated inelastic and elastic network traffic. Utility functions for elastic traffic, which include traditional data like file transfer and e-mail, were modeled as strictly concave, based on the intuition that they have decreasing marginal improvement for incremental rate increases. Some other applications, however, are not as delay-tolerant as elastic traffic and have an intrinsic bandwidth requirement. Multimedia streaming traffic falls into this category, which cannot be accurately modeled by concave utility functions. Shenker distinguished three different types of inelastic traffic: hard real-time, delay-adaptive, and rate-adaptive; he also proposed three utility forms for these categories. If elastic traffic is considered to be positioned at one end of the delay tolerance spectrum, at the end are applications with hard real-time requirements. Traditional telephony, and other applications that expect circuit-switched service, are examples of hard real-time applications. The utility curve for such applications looks like a step function as depicted in Fig. 4.2. Applications with hard real-time requirements would function much better in a network using admission control to ensure that the bandwidth shares never fall below the critical level [93].

Unlike traditional telephony, multimedia streaming over best-effort networks like the Internet are rather tolerant of occasional violations of delay bounds and packet losses. So their utility would not react as abruptly to rate changes. However, since multimedia applications have an intrinsic bandwidth requirement, independent of the network conditions, their utility would degrade as soon as the bandwidth share falls below the intrinsic data generation rate. For these delay-adaptive multimedia applications, the utility function has a sigmoidal form as shown in Fig. 4.2.

There is still another class of real-time or inelatic traffic: rate-adaptive applications, such as MPEG-DASH, which adjust their rate in response to fluctuations of available bandwidth. This adjustment keeps the delays moderate and avoids the streaming flow getting stalled, as long as the bandwidth share is not unreasonably low. Thus, the utility of the application and its user's QoE depends completely on the quality of the audiovisual signal. At large bandwidths, the marginal utility of additional bandwidth is very slight because the audio/video quality is already high and beyond human perception. Moreover, at very small bandwidths, the marginal utility is again very slight because the signal quality

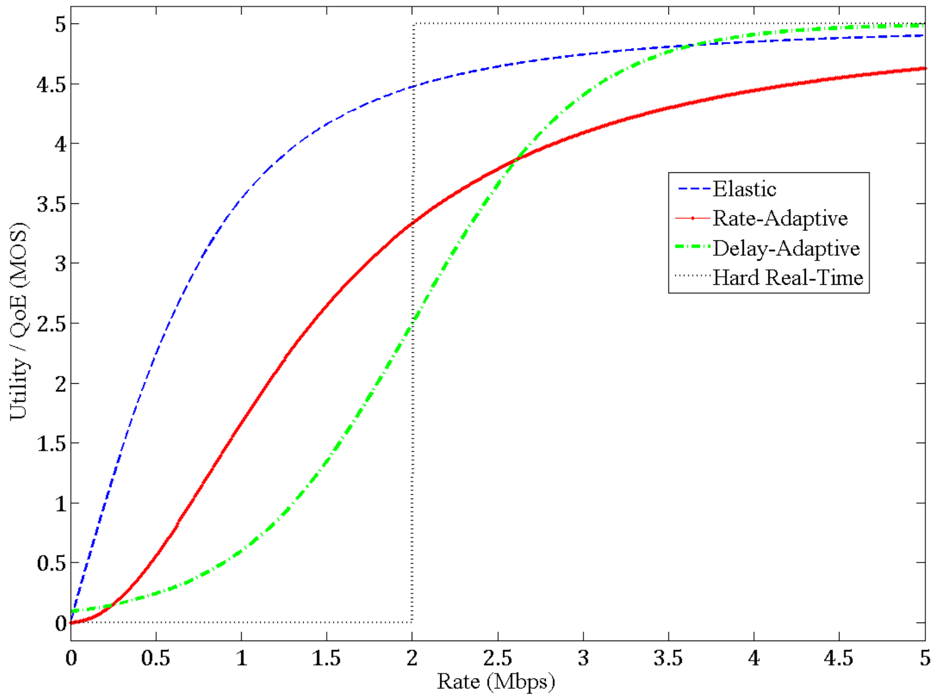

Figure 4.2: Utility functions for different classes of network flows

is unbearably low and perhaps imperceivable by the user [101]. Similar to the regular (delay-adaptive) multimedia utility functions, these utility functions are not concave in a neighborhood around zero; however, as the rate increases beyond the inflection point, which in this case is tied to the bandwidth need of the minimally acceptable quality, the curve becomes concave, as shown in Fig. 4.2. Recent experimental studies, such as [96], also confirm that the utility or QoE for YouTube video streaming service is actually a sigmoidal function of the video bitrate.

4.1.2 Nonconvex Network Optimization

Convexity is often viewed as the "watershed" between easy and hard optimization problems [95]. However, there has been much progress on nonconvex optimization over the past decade. Chiang [102] categorizes the proposed approaches into three alternatives for tackling nonconvex optimization problems in communication networks: 1) going "through" nonconvexity: solving the difficult nonconvex problem by for example, using successive convex relaxations, utilizing special structures in the problem, or leveraging smarter branch and bound methods; 2) going "around" nonconvexity: avoiding the nonconvexity by for example, changing variables or making approximations to convert the problem into a convex one, or determining conditions under which the problem is convex; 3) going "above" nonconvexity: reformulating the problem in the first place to make it more solvable or at least approximately solvable, also known as *design for optimizability*.

While some nonconvex optimization problems, like power control in wireless networks or spectrum management in Digital Subscriber Line technologies, are tackled using the "around" approach, and the nonconvex routing problem of the Internet is approximated using the "above" approach, nonconvex NUM problem appears to be manageable only by taking the "through" approach, which involves facing the difficulties of nonconvex problems. For more details about these three approaches, see [102].

One of the earliest attempts to solve the nonconvex NUM is the sum-of-square method [97]. This centralized computational method is empirically found to compute the globally optimal rate allocation efficiently. However, since this approach was not amenable to distributed algorithms required for conventional IP networks, it did not gain much attention at that time. The lack of theoretical guarantees on its optimality is another drawback, which hinders its application even in centralized settings. Finally, this method only works for polynomial utility functions, while we are using exponential sigmoidal utility functions. So we can neither use nor compare our method to this method.

Later, Chiang *et al.* studied distributed rate allocation for inelastic traffic [95]. For NUM with sigmoidal utility functions, they found a set of necessary and sufficient conditions under which the standard distributed price-based rate allocation algorithm would converge to the optimal solution despite the fact that the problem formulation is nonconvex. Note that relaxing the critical assumption of concave utility functions results in a nonconvex optimization problem, where a local optimum may not be a global optimum and the duality gap could be strictly positive. Therefore, the standard distributed rate allocation algorithm that solves the dual problem may produce infeasible or suboptimal rate allocations. The authors prove that continuity of price-based rate allocation with respect to the dual variable at all of the optimal prices is a sufficient condition for global convergence.

Authors in [94] extend the NUM framework and show that for sigmoidal utility functions, the optimal rate allocation is discontinuous at only one point. This discontinuity point can cause oscillations when trying to solve the problem using a gradient iterative method. A set of conditions is also proposed with at least one of them to hold when a user causes oscillations. To resolve these oscillations, they stop the user causing the oscillation from transmitting, allowing the rest of the users to stabilize. This method has been shown to approach optimal solution as the number of users tends to infinity. However, since it is excluding some users from being allocated resources, its fairness is questionable [103].

Recently, [98] addressed the problem of maximizing a sum of sigmoidal functions over a convex constraint set. It defines the *Sigmoidal Programming* problem and shows that the general problem is NP-hard. Built upon the mathematical results of [99], they propose an approximation algorithm using a branch and bound method to find a globally optimal approximate solution to the sigmoidal programming problem. We adopt this approach in developing an efficient solution method for our NUM problem with sigmoidal utilities.

4.1.3 Fair Resource Allocation

Generally, fairness does not simply mean allocating the same share to all users. Several notions of fairness have been proposed in the literature [46], with the two most representative ones being "max-min fair" and "proportional fair".

Max-min fair: Consider a resource allocation problem, with a feasible solution vector $\mathbf{x} \in \mathbb{R}^n_+$, where x_i is the resource allocated to user i. This allocation vector is defined to be max–min fair, iff no x_i can be increased without decreasing some other's share which is already smaller than or equal to x_i. Max-min fair allocation maximizes the minimum share and can be viewed as giving the maximum protection to the user with minimum allocated resource. Depending on the problem, a max-min fair allocation may or may not exist; however, in case of existence, it is unique. For network bandwidth allocation problem, algorithms have been developed for achieving max–min fair rates.

Proportional fair: The notion of max-min fairness puts emphasis on maintaining high values for the smallest. This might lead to resource utilization inefficiency. An alternative definition, known as proportional fairness, proposed by Kelly [29], considers an allocation vector $\mathbf{x}$ to be proportionally fair, iff it is feasible and for any other feasible vector $\mathbf{y}$, the aggregate of the proportional changes is non-positive; i.e.,

$$\sum_{i=1}^{n} \frac{y_i - x_i}{x_i} \leq 0. \tag{4.2}$$

There exists one unique proportionally fair allocation, which is obtained by maximizing the sum of logarithmic utility functions of users over the set of feasible allocations. It is shown that TCP's rate control roughly corresponds to the optimal solution of this logarithmic utility maximization problem, and therefore TCP fairness is close to proportional fairness [29].

In addition to the above two *qualitative* definitions of fairness, we can also *quantify* the degree of fairness associated with an allocation vector $\mathbf{x}$, using a fairness measure, which is a mapping from allocation vectors to real numbers. Various fairness measures have been proposed across different scientific disciplines, ranging from the simple ratio between the smallest and the largest entries, to more sophisticated functions like Entropy and Jain's index [44], the latter very common in network resource allocation, and defined as:

$$J(\mathbf{x}) = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n.\sum_{i=1}^{n} x_i^2} \quad \in [0,1]. \tag{4.3}$$

Jain's index, also simply known as fairness index, is a bounded and continuous function which is independent of population size and scaling of allocated values. Due to these desirable properties and its rather simple formula, Jain's index is the most widely used fairness measure for resource allocation methods.

4.2 Proposed Approach: NUM-SP

4.2.1 Problem Formulation

Consider the following optimization problem:

$$\begin{aligned} \underset{\mathbf{x}}{\text{maximize}} \quad & f(\mathbf{x}) = \sum_{i=1}^{n} f_i(x_i) \\ \text{subject to} \quad & \mathbf{x} \in \mathcal{C}. \end{aligned} \tag{4.4}$$

where $f_i(x) : [l, u] \to \mathbb{R}$ is a sigmoidal function for each i, and the variable $\mathbf{x} \in \mathbb{R}^n$ is constrained to lie in a nonempty bounded closed convex set $\mathcal{C}$. We call a Lipschitz [30] continuous function to be sigmoidal if it is either convex, concave, or convex for $x \leq z$ and concave for $x \geq z$ for some parameter $z \in \mathbb{R}$, which identifies the inflection point of the sigmoidal function. The above problem is called Sigmoidal Programming (SP) problem. The class of sigmoidal functions includes "s-shaped" graphs, such as the logistic function:

$$U(x) = \frac{c}{1 + e^{-a(x-b)}} \tag{4.5}$$

where c is the maximum value of the utility function, b is the inflection or turning point of the sigmoid, and a is the slope of the sigmoidal curve at turning point.

Now consider a network with S concurrent sources of multimedia flows (e.g. video streaming). We assume that all sources/users share a particular sigmoidal utility function $\forall s \ U_s(x_s) = U(x_s)$, where $U(\cdot)$ is of the form of Equation (4.5). The goal is to maximize total utility of all users. Network constraints could be described in terms of the routing

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:Topology xmlns:ns2="/XMLBinding">
    <NetworkElements>
        <NetworkElement>
            <Name>R1</Name>
            <NetworkElementId>R1</NetworkElementId>
            <Ports>
                <Port>
                    <PortID>tP1</PortID>
                    <Index>0</Index>
                    <Bandwidth>2000</Bandwidth>
                </Port>
                <Port>
                    <PortID>tP2</PortID>
                    <Index>1</Index>
                    <Bandwidth>2000</Bandwidth>
                </Port>
                <Port>
                    <PortID>tP3</PortID>
                    <Index>2</Index>
                    <Bandwidth>2000</Bandwidth>
                </Port>
            </Ports>
        </NetworkElement>
```

Figure 4.3: A segment of XML file describing network topology of the tree-shaped CDN

matrix $\mathbf{R}$, vector of link capacities $\mathbf{c}$, and non-negativeness of flow assignment variables, leading to the following Sigmoidal Programming variant of NUM, which we call NUM-SP:

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^S}{\text{maximize}} \quad & \sum_{s=1}^{S} U(x_s) \\ \text{subject to} \quad & \mathbf{R}\,\mathbf{x} \leq \mathbf{c}, \\ & \mathbf{x} \geq 0. \end{aligned} \tag{4.6}$$

4.2.2 Proposed Solution

We propose a centralized solution of the above problem, as opposed to the distributed ones typically used in existing work, because with the recent advent of Software-Defined Networking (SDN), it has become realistic to assume that a global view of the network is available, for example at the SDN's control plane, for the same network service provider. In networks that span multiple providers, each provider can implement our solution based

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:Service xmlns:ns2="/services/XMLBinding">
    <Name>SigTest2</Name>
    <UID>6f5a26ae-f5dc-4fbc-9b97-eb81f8c60e6e</UID>
    <Flows>
        <Flow>
            <Name>f-0</Name>
            <UID>f-0-SigTest2</UID>
            <RequestedBandwidth>300.0</RequestedBandwidth>
            <IngressNode>R1</IngressNode>
            <EgressNode>C1</EgressNode>
            <Weight>1.0</Weight>
            <MinimumBandwidth>200</MinimumBandwidth>
            <MaximumBandwidth>500</MaximumBandwidth>
        </Flow>
        <Flow>
            <Name>f-1</Name>
            <UID>f-1-SigTest2</UID>
            <RequestedBandwidth>300.0</RequestedBandwidth>
            <IngressNode>R1</IngressNode>
            <EgressNode>C3</EgressNode>
            <Weight>1.0</Weight>
            <MinimumBandwidth>200</MinimumBandwidth>
            <MaximumBandwidth>500</MaximumBandwidth>
        </Flow>
```


Figure 4.4: A segment of XML file describing services (streaming flows) over the CDN

on incoming and outgoing flows to/from its own network.

Our solution is adopted from the algorithm in [98], which employs the well-known branch and bound method [104] for nonconvex optimization problems and is in turn, based on the mathematical results on bounding the duality gap for problems with separable objective functions [99]. The branch and bound method is a recursive procedure for finding the global solution to an optimization problem restricted to a bounded rectangle. The method works by partitioning the rectangle into smaller rectangles, and computing upper and lower bounds on the value of the optimization problem restricted to those small regions.

The algorithm proposed by [98] computes upper and lower bounds for the sigmoidal programming problem by relaxing it to a tractable convex program. These upper and lower bounds are used as the basis of a recursion that eventually converges to the global solution of the problem. For a detailed description of the algorithm and the mathematical proof for its ϵ-suboptimality see [98] and [99], respectively. The only requirement of this algorithm is the ability to construct a concave upper bound on the objective functions to be maximized on any rectangle. Therefore, this algorithm could be applied to sum of sigmoidal functions, such as Equation (4.4).

An implementation of this algorithm is provided as an open-source Julia package [105]. This package models the sub-problems created at each node in the branch and bound tree using the Julia for Mathematical Optimization (JuMP) modeling language, making it easy to substitute different solvers. In our case, all the convex sub-problems are simply linear programs, which could be efficiently solved by GNU Linear Programming Kit (GLPK). We used this package to implement our solution for NUM-SP.

We model the user's QoE or utility function as a sigmoid described by Equation (4.5) with the following parameters: $a = 1$, which makes a unity slope around the inflection point of the utility function; b =450 kbps, which is the average minimum acceptable bitrate of video streams on different devices and as mentioned before, defines the inflection point of the sigmoidal function; and $c = 5$, which sets the maximum value of the utility to the scale of Mean Opinion Score (MOS) for QoE.

We also describe network topology, link capacities, and demanded services (multimedia streams) in XML files and feed them into our Julia code, which constructs the NUM-SP and solves it using solvers implemented in the Sigmoidal Programming package. Samples of these XML files are shown in Figures 4.3 and 4.4. As shown in the XML segment, for each flow, we define a minimum and maximum bandwidth required for the streaming to proceed, which are set to 200 kbps and 500 kbps, respectively.

4.3 Performance Evaluation

We constructed a wide range of video streaming scenarios over a number of realistic network topologies. In [106], we showed evaluation results using a backbone network topology of an actual Telco company. But since major video streaming platforms, like Netflix and YouTube, utilize Content Delivery Networks (CDN) for fast and efficient video delivery, here we use CDN topologies for our evaluations.

According to [87], typical access networks use a logical tree for their content delivery, although the underlying physical network is not a tree due to replication concerns. CDNs have three type of nodes: servers, proxies, and clients, which are connected in a logical tree topology in ℓ levels and with k branches at each level. Using a "Bottleneck Factor" (BF), the bandwidth of upper-level links are calculated based on the full load bandwidth requirement of all child branches multiplied by BF. Figure 4.5 shows the topology of a sample tree-shaped CDN, including one video delivery server (reflector), three proxy servers, and nine edge servers.

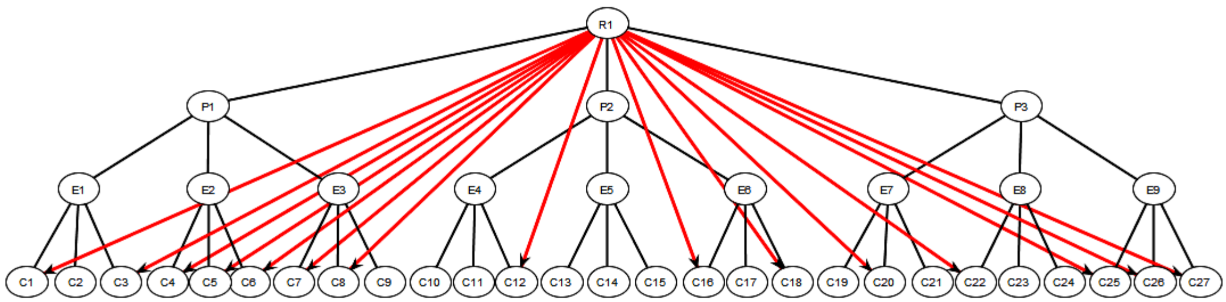

Figure 4.5: A CDN with tree-shaped topology: flows (represented by arrows), streaming from the delivery server (reflector) to clients

Table 4.1: Benchmark CDNs

Topology	**Reflector**	**Proxy Server**	**Edge Server**	**End-User**
CDN-A	1	5	25	125
CDN-B	1	10	100	1000

We consider two benchmark delivery networks: CDN-A as a small size delivery network with 125 end-users (clients), and CDN-B with a larger size of 1000 clients. Both networks have $\ell = 4$ levels, with $k = 5$ branches at each level for CDN-A, and $k = 10$ for CDN-B. Table 4.1 lists the number of delivery servers (reflectors), proxy servers, and edge servers for each of these benchmark CDNs. The capacity of links from edge servers to clients is assumed to be equal to $500kbps$, to allow for acceptable video quality which is assumed to require $450kbps$, as mentioned before. Using a bottleneck factor of $BF = 0.8$, we set the capacities of other links of the CDNs at upper levels.

Table 4.2: Simulation Scenarios

Topology /Scenario	**Flows**	**Overloaded Proxies**	**Overloaded Edge Servers**
CDN-A S1	80	1	5
CDN-A S2	80	2	5
CDN-A S3	90	3	9
CDN-B S1	720	4	16
CDN-B S2	760	8	24

Our delivery networks are providing service to a number of concurrent streaming flows from the reflector to a random subset of the clients. In order to simulate different distributions of active clients, we design several streaming scenarios/services as listed in Table 4.2. For instance, while the number of concurrent flows and also overloaded edge servers in *S1* and *S2* scenarios for CDN-A are equal, they are representing different geographical distribution of overloaded edge servers, and hence different number of overloaded proxy servers. While the scenarios are designed to represent under-provisioned situations in terms of having overloaded links and proxy servers, the ratio of active clients (number of streaming flows) to all network clients is still less than the bottleneck factor used for network provisioning.

For each of these scenarios, we solve the NUM problem for three methods: our proposed NUM-SP method, Proportional Fair and Max-Min Fair. We use the resulting flow assignments from each of these optimization methods to calculate the level of users satisfaction, according to the utility functions described in Equation (4.5). Finally, we compare the methods in terms of the following metrics: 1) average utility of all users, 2) Jain's fairness index, 3) number of satisfied ($QoE \geq 3.5$) users, and 4) network capacity usage.

As shown in Figures 4.6 and 4.7, our NUM-SP method has consistently improved the average utility of network users, and thus is more efficient for bandwidth allocation. Specifically, we observe a minimum of 60% QoE improvement compared to the best contender of the other two methods. This is actually a direct outcome of our method taking into account the sigmoidal shape of the utility function, whereas Proportional Fair and Max-Min Fair do not do so. The percentage of satisfied clients (PSC) is a closely related metric that further validates the improvement. While PSC of other methods drops to almost 10% in some cases (*e.g.*, CDN-B *S2*), our method maintains a minimum PSC of 50%, even in seriously under-provisioned scenarios.

Jain's fairness index results show that NUM-SP is significantly fairer than the other two methods, with improvements of at least 45%. The reason is that our method does not allocate more bandwidth to users who have already reached the upper end of the sigmoid with a satisfactory QoE, since it realizes that increasing their bandwidth share does not significantly improve their utility. This is not the case for the other methods, whose utility is concave and increasing with rate.

Aside from the improvements in utility and fairness, we are also interested in total network capacity usage, from the network operator's point of view. Evaluation results indicate that, despite its superiority in providing higher average utility and fairness, NUM-SP is actually consuming slightly less network resources. Our other experiments show that even for well-provisioned scenarios, where all methods are able to achieve high average utilities and fairness, NUM-SP offers significant (up to 20%) advantage in network capacity usage. This is mainly because NUM-SP does not waste bandwidth by allocating it to users

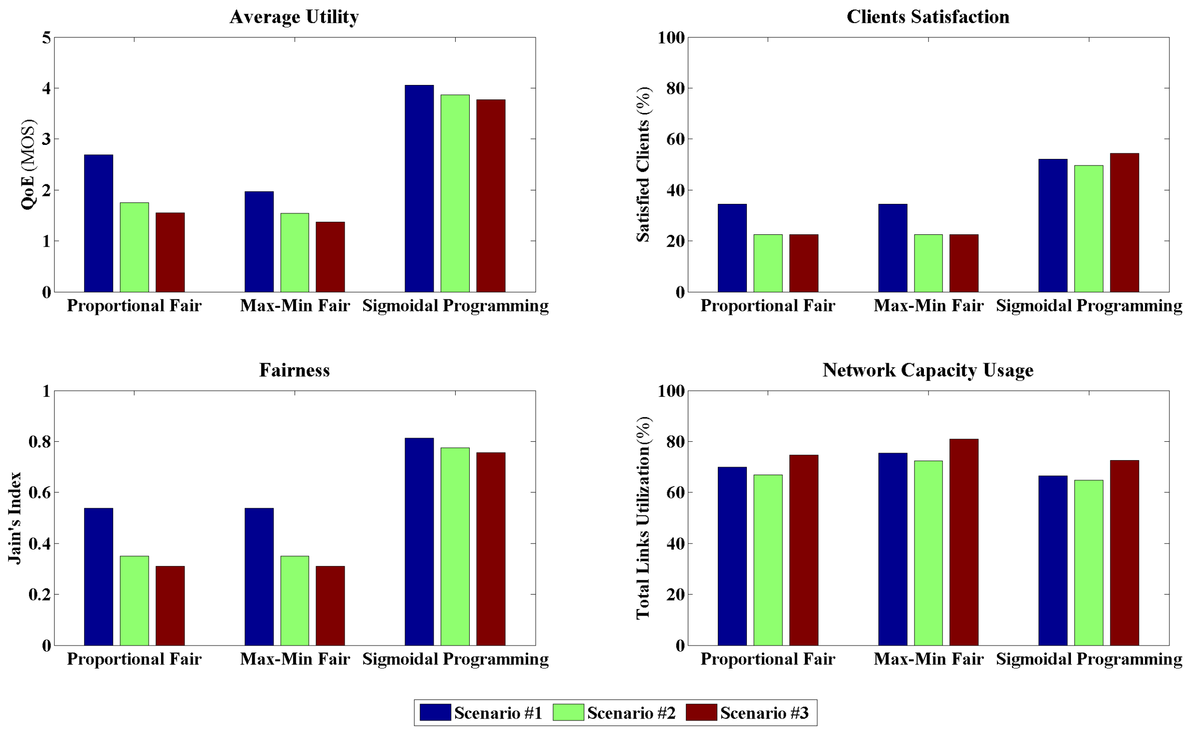

Figure 4.6: Comparison of different methods in terms of performance metrics for CDN-A network topology

whose final share of the bandwidth would remain in the lower part of the sigmoid and hence their utility/QoE would not improve. In addition, if a user is already in the upper part of the sigmoid, enjoying a satisfactory level of QoE, our method would not allocate more bandwidth to him, whereas other methods maximizing concave utilities would do that.

4.4 Summary

Although the bulk of today's Internet traffic is multimedia, especially on-demand video streaming, conventional rate allocation algorithms assume a concave utility function which does not apply to multimedia flows, which have a sigmoidal utility function. In this chapter, we proposed a centralized solution for video flow utility maximization, using sigmoidal programming.

Utilizing some recent mathematical findings for minimizing a sum of nonconvex functions over a compact domain, we developed a solution method for the NUM-SP problem and, despite its NP-hardness, we obtained approximate solutions guaranteeing ϵ-suboptimality. Our experiments showed both the feasibility and the superiority of our

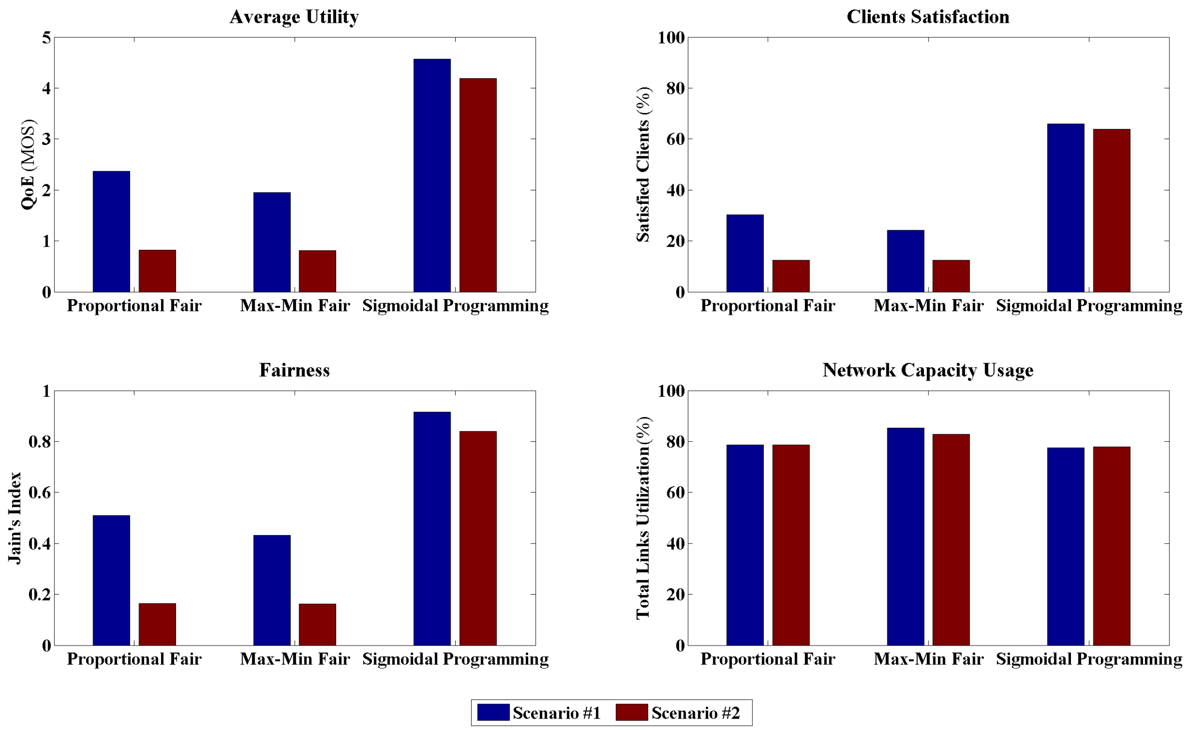

Figure 4.7: Comparison of different methods in terms of performance metrics for CDN-B network topology

solution compared to other representative optimization methods, in terms of utility and user satisfaction, fairness, and network capacity usage.

It is worth mentioning that our work is not restricted to TCP- or UDP-based video streaming schemes. In fact, the only assumption in NUM-SP is the sigmoidal utility function; therefore, it should work for any kind of multimedia streaming.

Numerical results show that our proposed sigmoidal programming method outperforms existing rate allocation algorithms, such as proportional fair and max-min fair, in terms of efficiency (total or average utility of network users) and fairness (Jain's index), while utilizing similar or less amounts of network resources. Specifically, we observed improvements of at least 60% in average utility (QoE) and 45% in fairness, compared to other representative optimization methods.

To conclude, we would like to mention that although one could also think of using a staircase utility function consisting of multiple sigmoid functions connected one after the other, and each representing a different video quality (e.g., 720p, 1080p, 4K, etc.), this scheme is not appropriate because it does not match the real-world observations reported in the literature, which have found that even with multi-quality video, such as those used in YouTube, the video's utility function is still a single sigmoid [96].

Chapter 5

Addressing Fairness Explicitly: From NUM to SUM

Having reviewed NUM framework in Chapter 2 and having proposed NUM-SP as a new variant of NUM for multimedia streaming in previous chapter, we now introduce the novel concept of *Social Utility Maximization* (SUM) as a replacement for NUM, with explicit handling of the fairness issue in bandwidth allocation.

We already know that depending on the type of utility functions used in formulating the NUM problem, the solution would implicitly satisfy some form of fairness; e.g. proportional fairness for logarithmic utility functions. However, such notions of fairness are not as intuitive as desired in many applications, and yet there is no guaranteed way to achieve them for other types of utility functions. SUM framework incorporates a desired fairness measure directly into the objective function of the optimization problem so that its solution provides a socially optimal distribution of the network resources.

We first present the key conceptual components of the SUM framework, and then elaborate on the technical details.

5.1 Core Components

5.1.1 Ultimate Utility: QoE

The ultimate goal of bitrate adaptation is to improve the QoE of users in order to achieve higher long-term user engagement. So in formulating our multi-agent decision process model, we assume that the utility function of the agents is their QoE. Since QoE is a subjective measure of quality which is not readily available in real-time, an automatic prediction method is required to map the network's Quality of Service to user's QoE.

Different QoE models could be plugged into the SUM framework. In Section 2.4, we provided a brief review of some QoE models for video streaming applications.

5.1.2 Social Welfare: Efficiency and Fairness

When different network users selfishly compete for the shared resources like bandwidth, their behavior could affect the entire system and result in a damage to everyone's interest. This problem known as *Price of Anarchy* [107] is a widely studied concept in economics and game theory that measures how the efficiency of a system degrades due to selfish behavior of its agents. To avoid such degradations, a multi-agent system could be designed in a way to promote explicit or implicit cooperation among its agents. This could be done by providing incentives to the agents for behaving cooperatively. As a matter of fact, there is strong scientific evidence that shows intelligent selfish agents are capable of learning to cooperate [108].

We assume that all users of the network are aware of the benefits of a cooperative approach, and are programmed to behave altruistically, as opposed to selfishly. This assumption does not violate the independent operation of the agents, nor does it impose any type of communications or information exchange among the users. It simply assumes that the geographically distributed users of the network not only share common resources, but also share a common objective function, called *social welfare*, which captures both *efficiency* and *fairness*. Here, efficiency refers to maximizing total utility of all users. We also use Jain's index as a measure of fairness of utilities.

We define the social welfare function as a weighted sum of the network's total utility and some fairness measure of the distribution of utility, namely Jain's index. We will elaborate on technical details in the subsequent section.

5.1.3 Foresighted Optimization under Uncertainty

Most existing rate adaptation methods, whether buffer-based or throughput-based have a myopic and reactive behavior based on the current network condition. Instead, we propose to employ MDP-based models to perform a foresighted optimization over a finite horizon, so that the rate adaptation scheme would be capable of taking into account the future impacts of its rate adjustments on user's QoE. Specifically, the proposed foresighted optimization framework attempts to avoid video stalling and also minimize video quality switches, in order to maximize QoE.

Modeling the inherent uncertainty of the network condition is another aspect of the proposed optimization method. Even in control-theoretic approaches [109] to dynamic rate adaptation, which leverage Model Predictive Control for a foresighted optimization, a *deterministic* model-based approach is taken. We emphasize that this is a problem of decision-making under *uncertainty* as described in Section 2.5.

In our proposed rate adaptation framework, we will develop a foresighted optimization under uncertainty using the MDP-based decision process models, that tries to maximize the long-term expected QoE of the user.

5.1.4 Common Interest Game in a Multi-Agent System

Although video streaming is obviously a multi-agent problem with more than one utility-maximizing decision-maker, competing for obtaining larger piece of the shared cake, there are very few works in the literature that are based on models of multi-agent systems. As one of the key components of our proposed modeling framework, we employ multi-agent models and decision processes for capturing the strategic interactions among different network users in a rate-adaptive video streaming scenario.

Common interest or cooperative games are special types of multi-agent systems in which the individual utility of the agents coincides with the joint utility of the group [67]. Since the agents share a common objective function, they are all open to cooperation, leading to a grand-coalition [110]. We try to leverage this cooperative behavior to maximize the social welfare function to achieve both efficiency and fairness at the same time. This results in a *coordinated* resource allocation where Price of Anarchy is minimized and public good is provided.

5.2 Definition of Social Utility

We assume that all users of the network are programmed to behave altruistically, as opposed to selfishly, although they all operate independently without any type of communications or information exchange. In other words, the geographically distributed users of the network not only share common resources, but also share a common objective function, called *social welfare* or *social utility*, which captures both *efficiency* and *fairness*.

By efficiency, we mean maximizing total utility of all users. Fairness, on the other hand, could have many different interpretations and criteria [111]. Various fairness measures have been proposed across different scientific disciplines, ranging from the simple ratio between the smallest and the largest entries, to more sophisticated functions like Jain's index [44], which is very popular in network resource allocation. An axiomatic theory of fairness was constructed by [46] in an attempt to formalize the notion of fairness. This work showed that all fairness measures satisfying five basic axioms, form a unique family of functions $f_\beta(\cdot)$ parametrized by β, where the Jain's index corresponds to the special case of $\beta = -1$.

We define the social welfare function as a weighted sum of the network's total utility and some fairness measure of the distribution of utility, namely Jain's index. Let $\mathbf{u} = [u_1, u_2, \ldots, u_N]^T$ be the vector of utilities (*i.e.* QoE's) achieved for all N users of the network. We apply Jain's fairness function to the allocation vector and combine it with the sum $\sum_{n=1}^{N} u_n$ as a measure of efficiency in order to construct a scalar metric for maximization of both objectives:

$$\Phi(\mathbf{u}) = \log\left(\sum_{n=1}^{N} u_n\right) + \lambda \log\left(J(\mathbf{u})\right) \tag{5.1}$$

where

$$J(\mathbf{x}) = \frac{\left(\sum_{i=1}^{N} x_i\right)^2}{N.\sum_{i=1}^{N} x_i^2} \in [0,1] \tag{5.2}$$

is the well-known Jain's index for distributive fairness [44] and λ serves as the relative importance of our two objectives.

It could be shown [46] that there is an upper bound for λ in order for the welfare function to preserve the common sense of Pareto dominance. It turns out that λ should be less than or equal to $\bar{\lambda} = \left|\frac{\beta}{1-\beta}\right|$, which would be equal to $1/2$ for the case of Jain's index. Replacing $\lambda = 1/2$ in Equation 5.1, our social welfare function could be written as

$$\Phi(\mathbf{u}) = \log\left(\frac{\left(\sum_{n=1}^{N} u_n\right)^2}{\sqrt{N \cdot \sum_{n=1}^{N} u_n^2}}\right). \tag{5.3}$$

Chapter 6

SUM with Partial Observability of Congestion: Dec-POMDP

After introducing the framework of SUM in previous chapter, we now turn to the application of SUM to the problem of bandwidth sharing in a multi-user video streaming scenario. Video traffic on the Internet has been growing at a rapid pace during past decade, and an increasing fraction of this video traffic comes from video streaming services, such as live streaming, video-on-demand and over-the-top (OTT) video services, usually used concurrently by several users who share common network resources. Hence, a fair and efficient video rate allocation model is required to 1) prevent congestion, and 2) provide a balanced video quality to all end users. Since there is no centralized authority for resource allocation in the Internet, a distributed solution for video rate adaptation is needed for congestion avoidance and bandwidth sharing among multiple video streams. The design of such a solution is a challenging problem in today's video delivery industry.

The earliest attempt to providing such a solution was TCP-Friendly Rate Control (TFRC) [12], in which the video sender infers the network condition from the estimated packet loss rates and delay metrics reported by the receiver via feedback packets. Naturally, this can only *react* to network congestion or packet loss and lacks a foresighted behavior. Some network-assisted approaches [85] have been proposed to fix these issues and improve agility in response to abrupt changes in traffic or network conditions. Although TFRC tries to provide fair bandwidth sharing to flows of different protocols, most existing congestion control solutions fail to provide a fair allocation of network bandwidth among competing video streams. Generally, fairness is not explicitly taken into account as an objective. Even when fairness is considered [79], it is about fair distribution of throughput, while the end users' quality-fairness, which is more desirable and has the ultimate impact on the human user, is typically ignored.

In general, existing approaches suffer from one or more of the following four shortcomings: 1) are generic protocols that are application-agnostic and do not take into account video quality, 2) do not capture the multi-agent nature of the problem, even though the problem clearly involves more than one utility-maximizing decision-maker interacting with each other, 3) do myopic adaptation only based on instantaneous rates, and 4) do not explicitly or correctly address fairness.

In this chapter, we address all of the above shortcomings and target a quality-driven fairness-aware end-to-end congestion avoidance and bandwidth sharing mechanism. We propose a decision-theoretic model, called *Decentralized Partially Observable Markov Decision Process* (Dec-POMDP), to formulate the interaction of multiple concurrent video streaming sessions over the Internet.

Aiming at maximizing the perceived quality of end-users while maintaining fairness in network bandwidth allocation, we employ a QoE model and introduce a *social welfare* function by combining the main objectives of *efficiency* and *fairness*. The solution of the proposed multi-agent decision process provides an optimal policy for all network users to adapt their streaming rates in the best interests of the entire network, leading to an optimum fair distribution of QoE among users. We evaluate the performance of this rate adaptation scheme through simulations, showing its advantages over the TFRC.

6.1 Problem Description

We consider the problem of network bandwidth sharing by several video streaming sessions over the Internet, seeking a fair and efficient distribution of Quality of Experience (QoE) from the media consumers' perspective. All network users are supposed to share a common altruistic goal to maximize some notion of social welfare. A dynamic rate adaptation scheme is also assumed to be implemented by the media servers.

6.1.1 Network Model

Consider N concurrent video streaming sessions over the Internet, sharing the bandwidth of the network. Each session is composed of a sender node (media streaming server) and a receiver node (client) that establish an end-to-end transport layer connection, equipped with TFRC protocol [12] to stream a multimedia content. We assume that the network has a single bottleneck link that causes packet loss once congested. Since the sender-side of each session cannot observe the traffic generated by other sessions, it is only able to infer the congestion status based on the feedback information received from the network

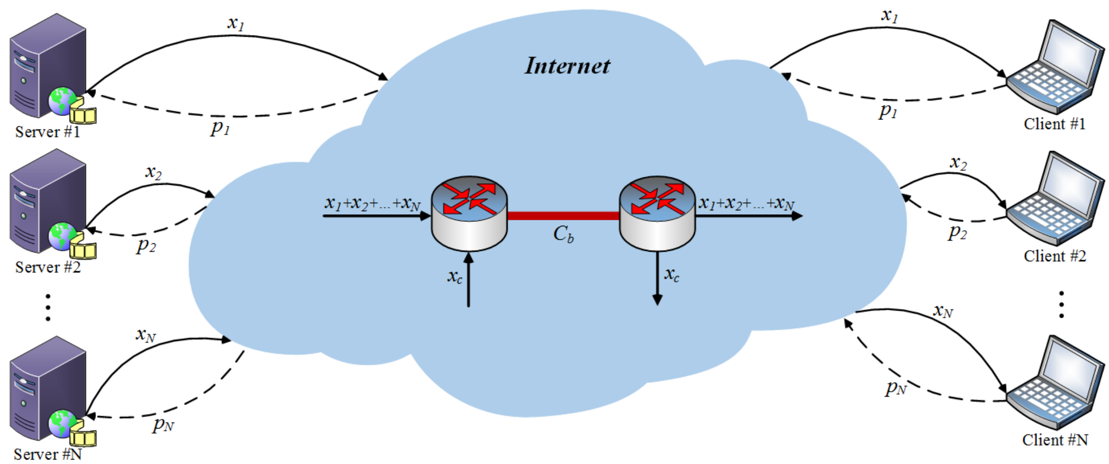

Figure 6.1: Schematic diagram of the problem showing N concurrent video streaming sessions

or receiver-side. The major source of information about network congestion level is the receiver's estimate of packet loss rate, which is included in TFRC feedback packets.

As depicted in Figure 6.1, each session n chooses its transport-layer sending rate x_n at sender-side and receives an estimate of the packet loss rate p_n from the receiver-side. The bottleneck link of the network, with a capacity of C_b, should handle the total sum of sending rates plus a time-varying cross-traffic x_c. As the total traffic on bottleneck link comes close to its capacity, all sessions would experience higher rates of packet loss. Therefore, a distributed rate control and congestion avoidance scheme, capable of dealing with this dynamic and partially observable environment is required.

6.1.2 Utility Model: QoE

The utility of users is their ultimate Quality of Experience (QoE). Since QoE is a subjective measure of quality which is not readily available in real-time, an automatic prediction method is required to map the network's Quality of Service to user's QoE. We first [112] adopted the G.1070 opinion model recommended by ITU-T [113] as the video quality model for predicting the subjective quality measured in MOS (see Section 2.4). However, due to limited scope of G.1070 to video sequences of up to VGA resolution, we later employed G.1071 recommendation [114] as our QoE model for video streaming. ITU-T G.1071 provides models which deliver estimates of the impact of typical IP network impairments on the quality experienced by the end user in multimedia mobile streaming and Internet protocol television (IPTV) applications over transport formats such as RTP over UDP, which is the case for the streaming scenario that we study in this chapter. Moreover, this

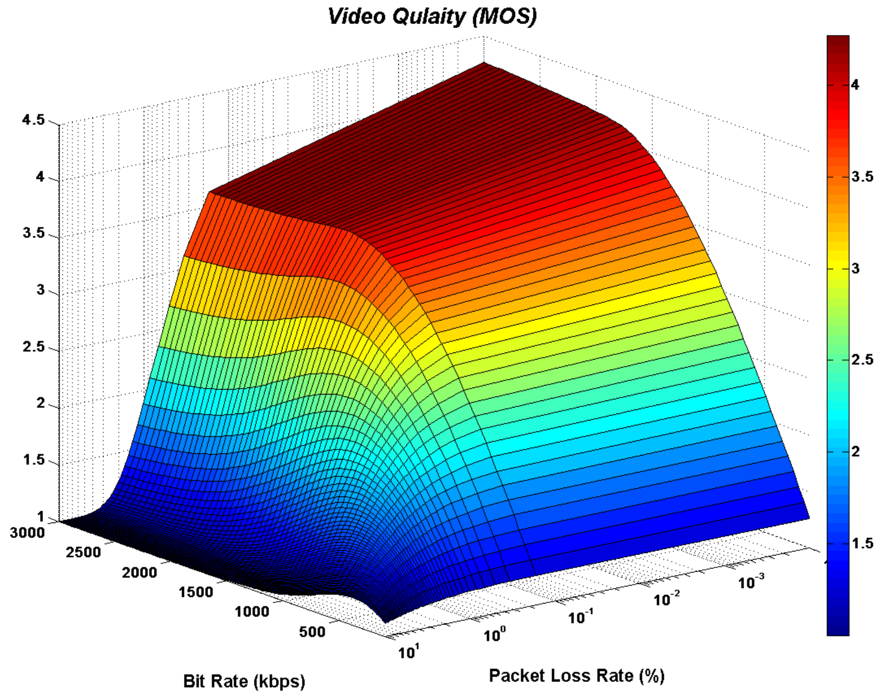

Figure 6.2: Video Quality based on ITU-T Recommendation G.1070

Recommendation addresses two application areas: lower resolution, including services such as mobile TV, and higher resolution, including services such as IPTV.

There are several factors affecting the perceived quality of a video: codec type and specifications, spatial resolution, key frame interval, delay, frame rate, bit rate and packet loss rate. Among these, we assume that all are fixed (or have negligible impact on video quality) during a streaming session except the last two: encoding bit rate (r) and packet loss rate (p). Under such conditions, we first describe the QoE model of G.1070, and then move on to the higher resolution model, recommended by G.1071.

According to G.1070 and under the above conditions, the utility (u) or QoE of the user could be summarized as:

$$u = 1 + \left[c_1 \left(1 - \frac{1}{1 + (\frac{r}{c_2})^{c_3}}\right)\right] e^{-\frac{p}{d}} \quad \in [1, 5] \tag{6.1}$$

where c_1, c_2, and c_3 are codec-dependent constants and d is the degree of robustness against packet loss (p) and is itself a function of bit rate (r), frame rate and a number of codec-dependent constants.

A visualization of the above QoE model and its variations with respect to bit rate and packet loss is shown in Figure 6.2. It is clearly observed that a significant degradation of

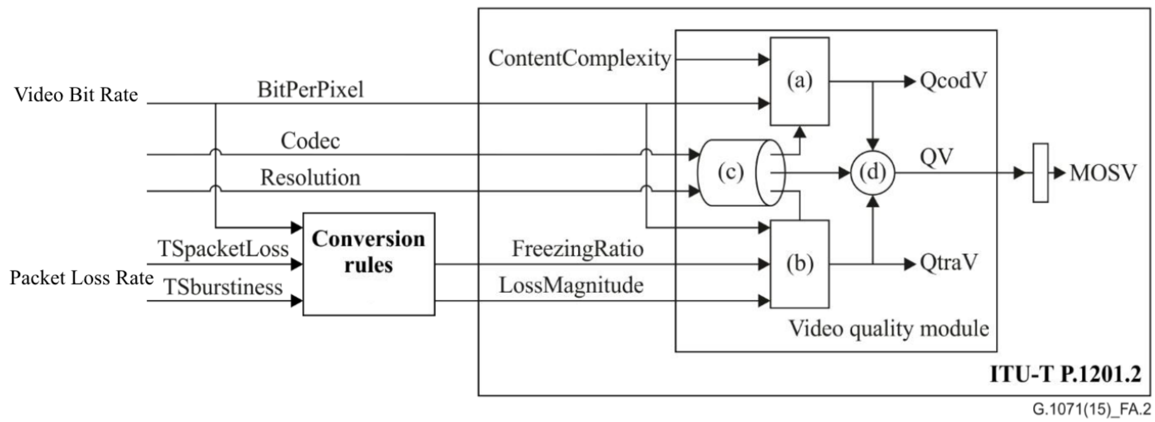

Figure 6.3: Block Diagram of Higher Resolution Model of ITU-T Recommendation G.1071

video quality occurs at packet loss rates above 0.05%, even with a pretty high source bit rate. Therefore, a QoE-aware rate adaptation scheme at the sender-side should not only try to maximize its bandwidth utilization, but also be able to pro-actively avoid congestion leading to higher packet loss rates.

The higher resolution QoE model of G.1071, is much more complicated compared to the above model. It covers video sizes of up to 1080p with coded video bitrates of up to 30 Mbps, using H.264 codec. The QoE model is also valid for packet loss rates of up to 2%. A block diagram of the video module of this QoE model is depicted in Figure 6.3. The actual inputs to the model are video bitrate and packet loss rate, as before. We fix the codec and resolution inputs to H.264 and HD 720p, respectively, and use the corresponding coefficient sets: a, b, c, and d.

The model's output, $MOSV$, is a combination of coding quality, Q_{codV}, and transmission quality, Q_{traV}, which are mainly dependent on video bitrate and packet loss rate, respectively. In using this QoE model, we assume that only uniform packet loss is present and disregard burstiness. We also assume that a packet loss concealment scheme is used to repair erroneous frames, hence *LossMagnitude*, not *FreezingRatio* would be the dominant factor due to packet loss rate. Annex A of [114] contains all the details of this QoE model.

6.1.3 Rate Adaptation Mechanism

State-of-the-art adaptive video streaming [3] [4] embed the rate adaptation algorithm inside the client application. This allows the client to independently choose the playback quality without any need for intelligent components inside the network. However, both industry [115] and academia [87] are showing interest in server-side or network-assisted adaptive streaming. In this study, we are considering a *server-based adaptive streaming*.

The adaptation mechanism is to be done jointly at both the application and the transport layers. Video source rate control would be carried out by the application layer, while a TFRC-like congestion control scheme is performed at the transport layer to take care of the packet loss rate. In TFRC, the sending rate x reacts to variation of packet loss rate p using the following formula:

$$x = g(p) = \frac{\ell}{RTT} \sqrt{\frac{3}{2p}} \tag{6.2}$$

where ℓ represents the packet size and RTT is the round-trip time.

Similar to TFRC, our server-based adaptation mechanism includes a rate switching scheme at transport layer, which uses a mapping from observed packet loss rates to optimal sending rates. But unlike TFRC, this adaptation is not tied to instantaneous values; rather it takes into account the history of observations and tries to do a foresighted optimization.

At the application layer, the source rate r (the target rate of the live video encoder or the source rate of pre-encoded videos) is adjusted according to the chosen sending rate using the sending buffer level:

$$r = x + \delta \left(B - \frac{B_u + B_l}{2} \right) \tag{6.3}$$

where B is the current buffer level, B_u and B_l are the upper and lower limits of the buffer and δ is the adjustment rate. The buffer's input and output rates are r and x, respectively. Using this rate adjustment method, the source rate would closely follow the sending rate, especially at steady state. Therefore, we might use them interchangeably in some approximations later on.

6.1.4 Objective Function: Social Welfare

As elaborated in Chapter 5, we assume that all users of the network are programmed to maximize a common objective function, which is the social welfare, defined by Equations 5.1 and 5.3, which captures both efficiency and fairness of bandwidth allocation.

6.2 Proposed Model: Dec-POMDP

Our proposed Dec-POMDP model is specified by the following components:

- **Agents**

  The dynamic interaction is taking place among a finite number of video streaming sessions, regarded as agents or users, indexed by $n \in \mathcal{N} = \{1, 2, \ldots, N\}$. Each streaming session comprises a sender and a receiver node in the network. However, since we are considering a sender-based adaptive streaming, the agents of the model are actually the sending entities of the video streaming process. The receiver nodes help provide the observations to the senders by sending back TFRC-like feedback packets. This multi-agent decision process runs over the course of a sequence of discrete time steps indexed by $k = 0, 1, 2, ...$ We assume a fixed *Round-Trip Time* (RTT) for the network, and set the time step of Dec-POMDP to be equal to one RTT.

- **Actions**

  We take the *sending rate* (x) of each video stream at the transport layer as the action of the agent. Note that the *source encoding rate* (r) of the video streams are closely tied to the chosen sending rate. We assume that in live video streaming, the target encoding rate of the video stream is adjusted at the application layer, according to the sending rate, using the sending buffer level. For the case of on-demand streaming, the video is supposed to be pre-encoded at corresponding source rates as well.

  Formally speaking, the action taken by user n at a given time step k is to choose the appropriate sending rate $a_n^k = x_n^k \in \mathcal{A}_n$, where $\mathcal{A}_n$ is the set of discrete pre-defined sending rates available for user n. The joint action of all users would be denoted by $\mathbf{a}^k = \langle a_1^k, \ldots, a_N^k \rangle \in \mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_N$.

- **States**

  The state of the proposed Dec-POMDP is set to be the unobservable *congestion level* of the network. As mentioned before, we assume that there is a single bottleneck link within the network that determines the congestion status of the network for the video streaming scenario under study.

  The congestion level, denoted by C_g, takes values in the interval of $[0, 1]$ and is defined as the ratio of total traffic on the bottleneck link to its capacity:

$$C_g = \min\{\frac{x_c + \sum_{n=1}^{N} x_n}{C_b}, 1\}, \tag{6.4}$$

  where C_b is the capacity of the bottleneck link, x_n's are the sending rates of the concurrent video streaming sessions, and x_c is the rate of the cross-traffic on the bottleneck link.

  Since the state space is a finite set in the formalism of MDP, we define the state of our model at time step k, as $\mathbf{s}^k = C_g^k \in \mathcal{S}$, where $\mathcal{S}$ is a discretization of the interval of $[0, 1]$ into a finite number of possibilities.

It is clear that the congestion level of the network is not directly observed by the users. Therefore, we are dealing with a Partially Observable MDP (POMDP) from the viewpoint of each user, where it has to infer the state of the decision process using its limited observations.

- **Observations**

  As mentioned above, each user observes neither the traffic generated by other users, nor the congestion level of the network, which is impacted by the overall traffic transmitted over the bottleneck link. The only available piece of information about the network condition is the TFRC-like feedback packets received per RTT. By means of each feedback packet, the sender observes the receiver's estimate of the *Packet Loss Rate (PLR)*, denoted by $\hat{p}$, which is an indication of how congested the bottleneck router of the network is.

  We define the observation of user n at time step k as $o_n^k = \hat{p}_n^k \in \mathcal{O}_n$, where $\mathcal{O}_n$ is a discretized set of possible values for PLR estimates.

- **Transition Function**

  In order to specify the transition function: $T : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$, which provides a probability distribution of the new state given the current state and the joint action of the users, we need a probabilistic model for the cross-traffic over the bottleneck link. If the joint action of the users at time step k is $\mathbf{a} = \langle x_1^k, x_2^k, \ldots, x_N^k \rangle$, the probability that the new state C_g^k falls into the interval $[\alpha, \beta]$, denoted by:

$$P\left(\alpha \le C_g^k \le \beta\right) = P(\alpha \le \frac{x_c^k + \sum_{n=1}^N x_n^k}{C_b} \le \beta), \tag{6.5}$$

  translates to the probability that the cross-traffic falls into a corresponding interval:

$$P\left(\alpha \cdot C_b - \sum_{n=1}^{N} x_n^k \le \ x_c^k \ \le \beta \cdot C_b - \sum_{n=1}^{N} x_n^k\right) = P\left(\tilde{\alpha} \le x_c^k \le \tilde{\beta}\right). \tag{6.6}$$

  We use the results of [116] for modeling the Internet backbone traffic at the flow level. According to their model, since the total cross-traffic is the result of a number of flows with independent rates, the central limit theorem asserts that the distribution of the cross-traffic tends to be Gaussian at high loads, which is typical of backbone links. The mean and variance of the rate of the cross-traffic are also calculated in this model in terms of the average size and duration of the contributing flows. Having specified the probability distribution function of the cross-traffic, we can calculate the transition probabilities for each joint action using Equation (6.6).

The transition probabilities depend on not only the joint action of the users, but also on the current state of the network. The auto-correlation function of the cross-traffic stochastic process induces this dependency since it restrains abrupt changes in the rate of the cross-traffic and accordingly the congestion level.

- **Observation Function**
  The observation function: $O : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{O})$ determines the probability of each joint observation $\mathbf{o}$ if a particular joint action $\mathbf{a}$ is taken that leads to a new state $\mathbf{s}'$. We assume that the probability of observing a certain rate of packet loss by each user is solely dependent on the network's state (congestion level) and independent of the users' actions (sending rates). This is justified by noting that the congestion level itself is caused by and derived from total sending rate generated by users plus cross-traffic rate. We also assume that the observation of each agent is statistically independent of other's observations. Therefore the probability of each joint observation would be equal to the product of individual probabilities.

- **Reward Function**
  The common goal of all agents is to achieve and maintain an optimal allocation of the network bandwidth leading to an efficient and fair distribution of end-users' QoE. Using the social welfare function defined in Equation (5.3), we can specify the common reward function of our Dec-POMDP model as:

$$R = \mathbb{E}\left[\sum_{k=1}^{h} \gamma^k \; \Phi(\mathbf{u}^k)\right] \tag{6.7}$$

  where γ is the discount factor (set to one in this model) and $\mathbf{u}^k = [u_1^k, u_2^k, \ldots, u_N^k]^T$ is the vector of utilities (*i.e.* QoE's) achieved for all N users at time step k as defined by Equation (6.1). Note that the QoE of the n^{th} user u_n^k is in turn a function of its source rate r_n^k and experienced packet loss rate p_n^k. Since the video source rate r_n^k is adjusted to the sending rate x_n^k, and the packet loss rate p_n^k is determined by the network congestion level C_g^k, the common reward function of Equation (7.3) would reduce to a function of joint action and state: $R(\mathbf{a}, \mathbf{s})$.

- **Horizon**
  The choice of time horizon in sequential decision problems is very critical. On one hand, it has to be large enough in order to fulfill the objective of foresighted (as opposed to myopic) optimization. On the other hand, it has to be chosen small enough for the problem to remain computationally solvable. We chose $h = 5$ for our model.

- **Initial State Distribution**

  We assume that network congestion is not at the extreme low or extreme high levels at the beginning of the video streaming service, and is uniformly distributed over the rest of the interval.

6.3 Optimal Solution of Dec-POMDP

6.3.1 Planning: JESP

The challenge of finding the optimal policies for a decision process, when the transition probabilities are available, is called *planning*. A policy is a mapping from states (in fully observable models) or beliefs (in partially observable models) to actions. The policy of a user is comprised of a sequence of actions selected by the user at every time step based on the state of the environment.

As mentioned in Chapter 2, solving a Dec-POMDP is a computationally expensive task. In fact, it is known that the problem of finding the optimal solution for a finite-horizon Dec-POMDP with even only two agents is NEXP-complete [56]. In practice, this means that solving a Dec-POMDP takes doubly exponential time in the worst case.

There are several methods proposed in the literature for finding approximate solutions of Dec-POMDP, from which we use *Joint Equilibrium based Search for Policies* (JESP) [66], which is guaranteed to find a locally optimal joint policy. It relies on a procedure called *alternating maximization*, that computes a maximizing policy for one agent at a time, while keeping the policies of the other agents fixed. This process is repeated until the joint policy converges to a Nash equilibrium: a tuple of policies such that for each agent's policy is a best response to the policies employed by the other agents. JESP uses a dynamic programming approach to compute the best-response policy for a selected agent, using a reformulation of the problem as an augmented POMDP by fixing the other agent's policies.

6.3.2 Learning: MARL

A *Reinforcement Learning* (RL) agent learns by trial-and-error interaction with its dynamic environment [18]. Well-understood algorithms with good convergence and consistency properties are available for solving the single-agent RL tasks, both when the agent knows the dynamics of the environment and the reward function, and when it does not.

There had been an extensive research effort during past decade to extend reinforcement learning theory and techniques to multi-agents situations often encountered in real-world

Algorithm 1 Multi-Agent Reinforcement Learning Algorithm for Adaptive Streaming using a Dec-POMDP Framework

Initialize Q-table for all possible observations and actions;
Choose a random action (sending rate): for $a_n^1 = x_n^1$;
$k \leftarrow 1$;
loop
  Get new observation of packet loss rate: $o_n^k = \hat{p}_n^k$;
  Get video source rate: r_n^k;
  Calculate the utility/QoE: $u_n^k = V_q(r_n^k, \hat{p}_n^k)$;
  Calculate the estimated reward: $\hat{R} = \sum_{\ell=0}^{h-1} \gamma^\ell u^{k-\ell}$;
  Pick a random number ζ uniformly from $[0, 1]$;
  if $\zeta > 1 - \epsilon$ **then**
    $x_n^k = a_n^k = \underset{a_n}{\operatorname{argmax}} \, \mathsf{Q}(\hat{p}_n, a_n)$;
  else
    Choose a random sending rate as action $x_n^k = a_n^k$;
  end if
  $\mathsf{Q}(\hat{p}_n^k, a_n^k) \leftarrow (1 - \eta_k) \ \mathsf{Q}(\hat{p}_n^k, a_n^k) + \eta_k \left[\hat{R} + \gamma \ \mathsf{Q}(\hat{p}_n^{k+1}, a_n^{k+1})\right]$;
  $k \leftarrow k + 1$;
end loop

applications. Many variations of Q-learning algorithm have been proposed for zero-sum and nonzero-sum non-cooperative games and also cooperative games with different levels of theoretical convergence guarantees and practical capabilities. For a comprehensive survey on this vast literature see [117].

However, in many practical cases, single-agent reinforcement learning methods are applied without much modification [118]. Such an approach treats other agents as a part of the environment. Following this track, we propose a multi-agent reinforcement learning scheme based on the single-agent temporal difference (TD) learning algorithm of SARSA [18]. The key idea is to see if the agents can learn the optimal policies for the Dec-POMDP model, only based on their locally available information. For this purpose, the MARL algorithm tries to learn the action-value function $\mathsf{Q}^\pi(s, a)$, which gives the expected reward of taking an action a at state s and then following the policy π.

As listed, the proposed MARL algorithm uses a simple SARSA learning rule based on the observations of packet loss rate and decides which sending rate to pick based on the learned action-value function. The proposed algorithm is computationally less complex than any dynamic programming based planning solutions, either exact or approximate. Based on the following proposition, we expect that this learning process would finally converge to the optimal solutions, obtained by planning the Dec-POMDP.

Proposition The proposed distributed reinforcement learning algorithm converges to the optimal value function of the DEC-POMDP problem if $\eta_k = \frac{1}{k}$.

Proof This result follows directly from the Theorem 1 of [119]. It is known that in sequential decision problems with finite action and state spaces, SARSA algorithm converges to the optimal policy and optimal value function if the reward function is bounded and the learning rate η^k meets the following two condition: $\sum_{k=1}^{\infty} \eta_k = \infty$, $\sum_{k=1}^{\infty} (\eta_k)^2 < \infty$. Since $\eta_k = \frac{1}{k}$ satisfies both of these conditions, we conclude that the Algorithm 1 converges to the optimal action-value function.

6.4 Implementation and Performance Evaluation

In order to find the optimal policy for our proposed Dec-POMDP model, we used *Multi-Agent Decision Process* (MADP) Toolbox [120], which provides software tools for modeling, specifying, planning and learning a variety of decision-theoretic problems in multi-agent systems. This toolbox includes a *parser* for reading text descriptions in Tony Cassandra's POMDP file format (`.pomdp`) [121] and its Dec-POMDP extension (`.dpomdp`).

The description of our Dec-POMDP model for two agents is shown in Listing 6.1. The state space (interval of $[0, 1]$ for congestion level) is discretized into five bands from very low to very high, namely `CgLL`, `CgL`, `CgM`, `CgH`, `CgHH`. Initial distribution of the state is assumed to be uniform over all states excluding the extreme cases: `CgLL`, `CgHH`. Both agents, assumed to be homogeneous in terms of their actions and observations, have three choices for their actions at each time step: sending video packets at 1, 2, or 3 Mbps rates, denoted by `R1M, R2M, R3M`, respectively.

The observation space (PLR estimates) is also discretized into five levels in logarithmic scale: `pLL` $\approx 10^{-5}$, `pL` $\approx 10^{-4}$, `pM` $\approx 10^{-3}$, `pH` $\approx 10^{-2}$, and `pHH` $\approx 10^{-1}$.

For each pair of joint action $\mathbf{a}$, a $|\mathcal{S}| \times |\mathcal{S}|$ matrix specifies the transition probabilities from start sate $\mathbf{s}$ to end state $\mathbf{s}'$. The probability values are calculated as described in Section 7.2. For the sake of brevity, not all matrices are shown in Listing 1; only the first and the last cases. As mentioned before, the probability of each observation only depends on the network state and is independent of the actions taken by the users and observations of other agents. Based on this assumption, the probability of occurrence of all possible joint observations are calculated and specified in the description of the model.

```
# Dec-POMDP Model for Adaptive Video Streaming
#--------------------------------------------
#Agents
#------
agents: 2
#Discount factor
#---------------
discount: 1.0
#Type of Values
#--------------
values: reward
#States (Congestion Level)
#-------------------------
states: CgLL CgL CgM CgH CgHH
#Initial state distribution
#---------------------------
start exclude: CgLL CgHH
#Actions (Sending Rate kbps)
#---------------------------
actions:
R1M R2M R3M
R1M R2M R3M
#Observations (Packet Loss Rate)
#-------------------------------
observations:
pLL pL pM pH pHH
pLL pL pM pH pHH
#Transition Probabilities
#------------------------
#  T: <a1 a2> : matrix of %f for all <s> & <s'>
#    CgLL      CgL       CgM       CgH       CgHH
T: R1M R1M :
    0.3845    0.5515    0.0635    0.0005    0.0000
    0.0671    0.8661    0.0665    0.0003    0.0000
    0.0642    0.5527    0.3820    0.0011    0.0000
    0.1363    0.5864    0.2703    0.0070    0.0000
    0.1584    0.6818    0.1571    0.0027    0.0000
...
T: R3M R3M :
    0.0000    0.0027    0.1571    0.6818    0.1584
    0.0000    0.0070    0.2702    0.5865    0.1363
    0.0000    0.0011    0.3820    0.5527    0.0642
    0.0000    0.0003    0.0665    0.8661    0.0671
    0.0000    0.0005    0.0635    0.5515    0.3845
#Observation Probabilities
#-------------------------
#  O: <a1 a2> : <s'> : <o1 o2> : %f
O: * * : CgLL : pLL pLL : 0.8191
O: * * : CgLL : pLL pL : 0.0814
O: * * : CgLL : pLL pM : 0.0045
O: * * : CgLL : pLL pH : 0.0000
O: * * : CgLL : pLL pHH : 0.0000
...
O: * * : CgHH : pHH pHH : 0.8191
#Rewards
#-------
#  R: <a1 a2> : <s> : <s'> : <o1 o2> : %f
R: R1M R1M : CgLL : * : * : 1.8855
R: R1M R1M : CgL : * : * : 1.8831
R: R1M R1M : CgM : * : * : 1.8591
R: R1M R1M : CgH : * : * : 1.6343
R: R1M R1M : CgHH : * : * : 0.7411
...
R: R3M R3M : CgHH : * : * : 0.6932
```

Listing 6.1: Description of Dec-POMDP Model in a Text Format

The reward model, as described in Section 6.2, only depends on the joint action and state: $R(\mathbf{a}, \mathbf{s})$. The calculated reward values for all combinations of actions and states constitute the last part of our model description. Note that only a few lines of the observation probabilities and rewards specification are included in Listing 1 due to limited space.

We used the JESP algorithm implemented in MADP Toolbox to solve our Dec-POMDP model. The result of JESP planning is a deterministic policy which maps every possible sequence of observations (PLR) with different lengths to an optimal action (sending rate). This policy is guaranteed to yield a local maximum of expected common reward, which provides both efficiency and fairness according to Equation (5.3). Using a simple look-up table, the resulting optimal policy could be hard-coded into the transport layer protocol of the media streaming servers to replace/augment TFRC.

In order to evaluate our proposed rate adaption scheme, we compare its performance with that of TFRC in terms of total utility of the users as well as the fairness index of the distribution of QoE. Since there are stochastic components in our problem formulation and proposed model, we conducted 50 rounds of simulations and calculated the mean values to cross out the random effects. Figure 6.4 illustrates one sample of these simulations for each of the methods. We can see that the sending rates chosen by TFRC tend to oscillate between minimum and maximum values due to the reactive nature of this mechanism, whereas the foresighted optimization of Dec-POMDP model provides less switching in sending rate, lower congestion levels and higher common reward at the same time.

Table 6.1 shows the average results of 50 runs of simulations for three main metrics: total utility of users, fairness of users' QoE, and the social welfare function (Equation (5.3)). Simulation results confirm that the optimal solution of our proposed Dec-POMDP model outperforms TFRC congestion control mechanism both in terms of efficiency and fairness.

Table 6.1: Comparison of the proposed solution with TFRC based on the average results of 50 runs of simulations

Metric	**Dec-POMDP**	**TFRC**	**Improvement**
Total QoE	7.452	5.975	24.7%
Fairness Index	0.881	0.780	12.9%
Social Welfare	1.946	1.663	17.0%

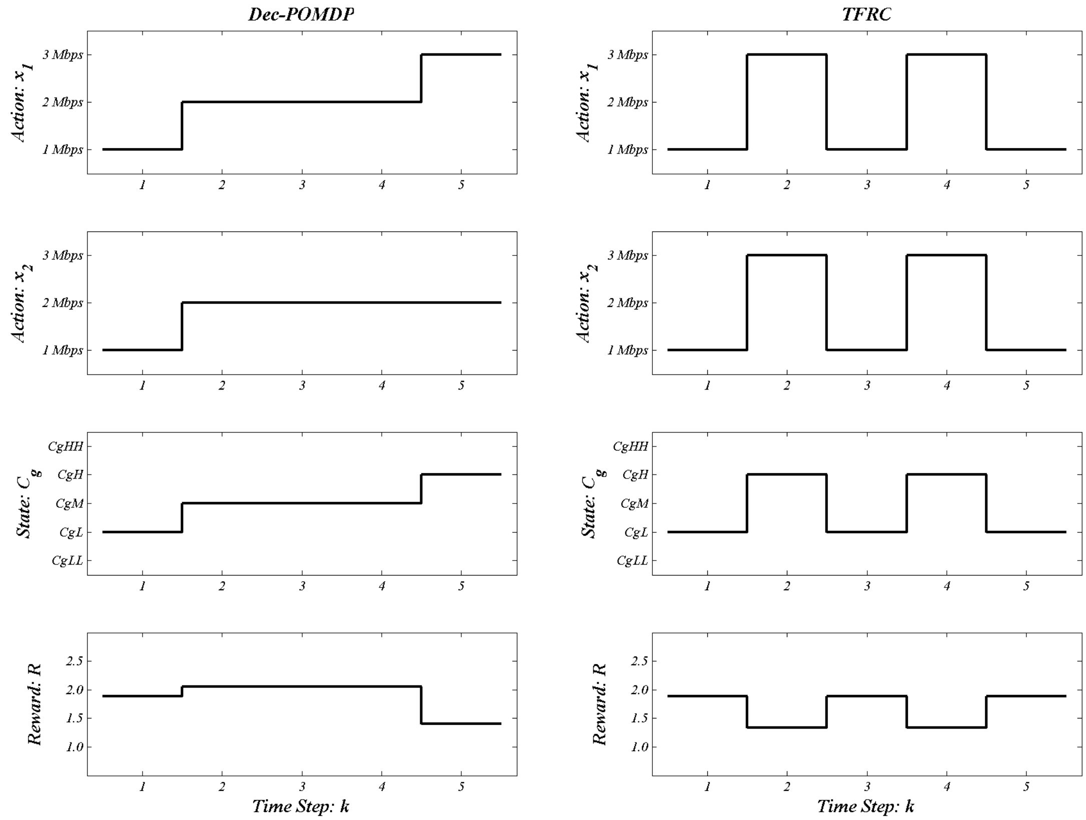

Figure 6.4: Samples of optimal Dec-POMDP solution and TFRC sending rates and the resulting network congestion and social welfare

6.5 Summary

The problem of network bandwidth sharing among multiple video streaming sessions was considered from a decision-theoretic and computational intelligence point of view. A Dec-POMDP model was proposed to capture the multi-agent aspects of the dynamic interaction between network users. A common objective function, called social welfare, which incorporates maximization of total utility while achieving a fair distribution of QoE, was designed to be collectively optimized by different users. The solution of this sequential decision-making process provides an optimal policy for each agent to adapt its sending rate based on the sequence of packet loss rate observations. The optimal solution of Dec-POMDP induces an implicit cooperation among network users, resulting in a much higher total QoE for users as well as improved fairness, in contrast to the popular TFRC.

We also showed that the proposed MARL algorithm would ultimately converge to the optimal policies. This is a promising result for the prospect of fully autonomous and online-learning network entities, capable of adapting their behavior to the dynamic and rapid changes of network conditions.

Chapter 7

SUM with Full Observability of Congestion: MMDP

The partial observability assumption made in previous chapter results in a multi-agent decision process that is computationally expensive to solve. In this chapter, we try to overcome this complexity issue by proposing an alternative model based on SDN, which provides full observability of the network congestion level. Specifically, we consider the problem of network bandwidth allocation, shared by several adaptive bitrate video streaming sessions over next-generation SDN-enabled wireless networks.

The *distributed* nature of conventional rate allocation methods lacks a substantial coordination in resource optimization and hence results in some inefficiencies and unfairness. Therefore, we utilize SDN's ability to provide a *centralized* global view of the network's current state, including its congestion level, which we share with all mobile users of the network by the SDN controllers and through the base stations. This transforms the client-side rate adaptation problem into a network-assisted one and gives us a fully-observable multi-agent decision process, which we take advantage of to design a multi-objective optimization model for network resources, seeking the distribution of QoE among all users in a fair and efficient manner.

Our proposed framework for mathematical modeling of the problem of fair and efficient allocation of network bandwidth among concurrent video streaming sessions is developed based on four key components from Chapter 5: 1) QoE as the utility function of users, 2) A social notion of optimality by integrating both efficiency and fairness, 3) Foresighted optimization under uncertainty of the environment, and 4) Common interest game between multiple agents of the system. We first present the QoE model used for HAS, which captures three important factors: the average video quality, stability of video quality, and stalls or freezes of video playback. Subsequently, a social welfare function is developed to capture both fairness and efficiency objectives at the same time, as explained in Chapter 5. We then propose a Multi-agent Markov Decision Process (MMDP) for finding the optimal network bandwidth allocation that leads to the maximization of the said social welfare function, resulting in a fair and efficient distribution of QoE from the video consumers' perspective. The foresighted optimization of MMDP framework attempts to avoid video stalling/re-buffering and also minimize video quality switches, in order to improve QoE.

We first provide a brief description of the system under study, and then the details of the proposed MMDP model would be presented.

7.1 System Description

Consider an SDN-enabled wireless access network such as LTE [73]. Following the recent trend of virtualizing the mobility core [122], we assume that the core network that connects the radio segment of the cellular network to the backbone IP network is equipped with SDN and NFV technologies. In such settings, channels of communication are available to SDN controllers for exchanging information, such as the global view of the network and the state of their own WAN, with mobile users through packet data network gateway (P-GW) and base station (eNodeB). This offers significant opportunities for enhancing the quality of mobile video delivery.

We also assume that n mobile users are using an HTTP-based adaptive video streaming service over this SDN-enabled wireless network. A client-side rate adaptation is taking place at the DASH players of these users, fetching video segments from some video delivery servers connected to packet data network. We further assume that all users are sharing a bottleneck link somewhere in the core network, the congestion level of which is observed by the SDN controllers and this piece of information is communicated with a centralized resource optimized, which resides in the "LTE SDN Application" box next to the PCRF in Figure 7.1, which is a modified version of the intelligent content delivery network proposed in [74], illustrating a schematic of our network settings.

The mobile DASH clients (agents) can share their states with the PCRF through existing 3GPP standard mechanisms. According to the Policy and Charging Control (PCC) architecture for Evolved Packet Core (LTE-Advanced Release 10), each Internet media service provider is given a separate Access Point Name (APN), and each APN is associated with an Application Function (AF). As defined in the 3GPP standardized PCC architecture [123], AF is in charge of interacting with applications running at UEs, e.g. HTTP-based Adaptive Streaming (HAS) applications in our case. HAS clients running at

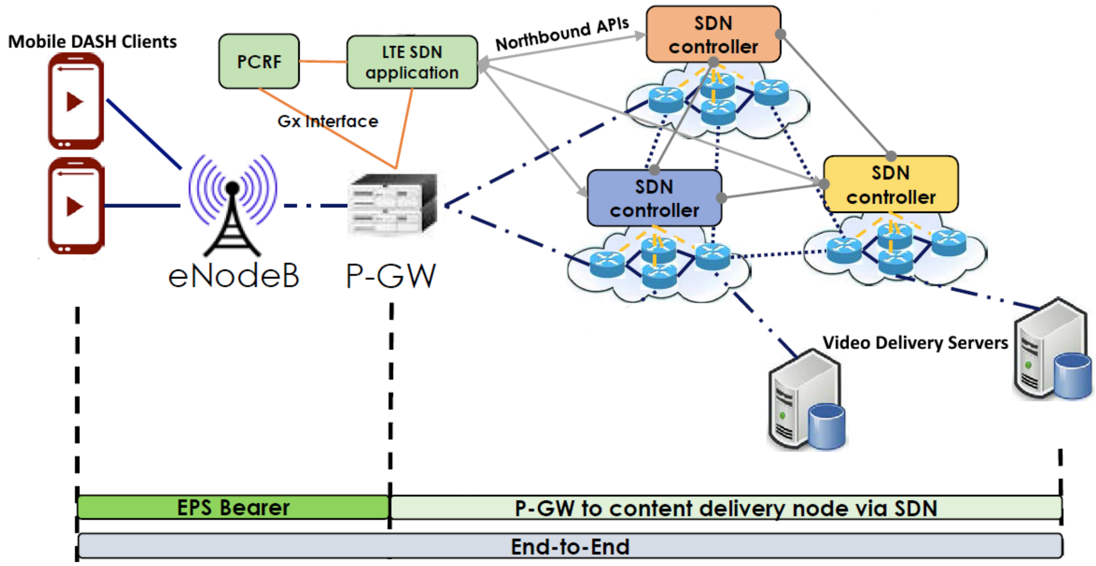

Figure 7.1: A schematic of SDN-enabled wireless network with a number of mobile DASH clients fetching video segments from video delivery servers (Nam2015)

the UEs are also informed to use the corresponding AF as a server to report their QoE feedbacks including average video bitrate, temporal quality changes and re-buffering time. Based on the above description, the received information is then passed to the PCRF which communicates with the resource optimizer residing at "LTE SDN Application".

7.1.1 SDN and Network Congestion Observability

The tremendous growth in cellular data traffic in recent years is mostly attributed to the surge of mobile video traffic which is in turn caused by huge popularity of mobile hand-held devices like smartphones and tablets and the convenience of generating and consuming of video using mobile devices. This ever-increasing demand has introduced several challenges in delivering video to mobile users with a good QoE. The quality of users' experience for mobile video is of critical concern today, since the increase in network resources has not kept pace with the surge in mobile video traffic demand [122].

Emerging technologies such as Network Functions Virtualization (NFV) and Software-Defined Networking (SDN) [71] are being considered to address the future needs and challenges of mobile networks [124], [125]. This trend of virtualizing the mobility core, in which the core network that connects the radio segment of the cellular network to the backbone IP network is getting equipped with SDN and NFV technologies, offers significant opportunities for enhancing the quality of mobile video delivery. A number of recent studies have shown the importance of cooperation between mobile devices and network infrastructure to improve the performance of video streaming in cellular networks [126], [127]. Since a global view of the entire network is available to the control plane of SDN, and also highly customized solutions for different scenarios could be easily deployed using NFV, the current trend of network softwarization can facilitate the development of collaborative solutions. For instance, [127] proposes a collaboration framework for mobile devices and a cellular network to exchange both device level information (e.g. delay tolerance at the given time instant) and cell-level information (e.g., the total traffic demand in the cell at a given time instant) in order to make proper decisions about delaying video streaming traffic.

During last few years, many researchers and practitioners in cellular network industry have argued that carrier networks can benefit from advances in SDN specially in terms of network management. For instance, a Software-Defined Mobile Networks (SDMN) architecture was introduced in [73] with programmability, flexibility, and openness features backed by experimental testing. In a more recent work, [74] addresses the challenges of implementing intelligent content delivery on SDN-enabled wireless access systems such as LTE and Wi-Fi. They leverage the SDN concept to dynamically control network traffic over WANs from edge nodes of wireless networks. A schematic of their proposed solution is depicted in 7.1. In such framework, channel of communication are available to SDN controllers for exchanging information (such as the global view and network state of their own WAN) with mobile users through P-GW and eNodeB.

Base on the above discussion, one argue that in next-generation SDN-enabled cellular networks, the state of the core network, including its frequently updated congestion level, could be shared with all mobile users of the network by the SDN controllers and through the base stations. This results in a fully-observable multi-agent decision process for distributed rate allocation. Observability of the network state could lead to a less complex decision problem from the viewpoint of each mobile user compared to the previously-studied case of partially-observable MDP.

7.1.2 QoE Model

Designing good QoE models for video streaming is an active area of research [48]. While a higher video quality is expected to improve QoE, existing literature on QoE of video streaming provides evidence that frequent switches or sudden changes in video quality or bitrate result in user's dissatisfaction and reduces QoE [49]. To this end, [50] introduced the notion of penalty for temporal variability of quality in modeling QoE. However, video freeze or stalling, which happens due to rebuffering of video player, is the most harmful factor in QoE degradation [51]: a single rebuffering event has three times the impact of a bitrate change. Another reported finding of [51] is that rebuffering incur abandonment rates six times higher than start-up latency.

One of the well-accepted QoE models for adaptive streaming provides a prediction or estimation of MOS through a linear combination of the average quality of video segments over a streaming session, the variation of video qualities and the effect of rebuffering:

$$QoE = MOS_{est} = \alpha.\mu - \beta.\sigma - \gamma.\phi + \delta \tag{7.1}$$

where μ is the average quality of video segments, σ is the standard deviation of video qualities, ϕ is a measure of rebuffering or freezes, and α, β, γ, and δ are tunable parameters [52]. Here, we formulate our QoE model, based on the control-theoretic model developed in [109].

Consider a video sequence, broken into K segments or chunks of equal duration and encoded at different bitrates. Let $\mathcal{R}$ be the set of all available bitrate levels. At each time step k, the video player chooses to download the k^{th} segment at bitrate $r_k \in \mathcal{R}$, which has a size of $d_k(r_k)$. Moreover, we assume that a nondecreasing function $q(\cdot) : \mathcal{R} \to \mathbb{R}^+$ maps the bitrate r_k to video quality perceived by user: $q(r_k)$. We also denote the buffer level of the player with $B_k \in [0, B_{max}]$, where B_{max} is the buffer size.

Following Equation (7.1), we define the utility u of each user as his QoE for the streaming session of the above video sequence by:

$$u = QoE = \alpha \sum_{k=1}^{K} \frac{q(r_k)}{K} - \beta \sum_{k=1}^{K-1} |q(r_{k+1}) - q(r_k)| - \gamma \sum_{k=1}^{K} \left(\frac{d(r_k)}{C_k} - B_k \right)^+ + \delta, \tag{7.2}$$

where the summations represent average video quality, video quality variation, and total rebuffering time, respectively, and α, β, γ, and δ are non-negative weighting parameters.

7.2 Proposed Model: MMDP

We consider a client-side rate adaption problem in an SDN-enabled wireless cellular network. As mentioned above, a global view of the entire network is available at the control plane of the SDN, and thus the network state including its congestion level could be communicated to all mobile users' devices by the base station. Hence, the distributed rate adaptation problem could be modeled by a fully observable cooperative multi-agent system. We will use MMDP framework to model the dynamic interactions among mobile users in this case.

Our proposed MMDP model $\langle \mathcal{N}, \mathcal{S}, \mathcal{A}, T, R, h \rangle$, is specified by the following components:

- **Agents**

  The dynamic interaction is taking place among a finite number of video streaming sessions with client-side rate adaptation (like DASH), which are referred to as agents or users and are indexed by $n \in \mathcal{N} = \{1, 2, \ldots, N\}$.

  Discrete time steps are indexed by $k = 0, 1, 2, ...$ Each time step is equal to duration of the video segments.

- **States**

  The state of the proposed MMDP is a vector comprised of two elements: the *buffer level* of the player B_k and the *congestion level* of the network G_k. Since the buffer level is locally maintained by each user, and the information about congestion level is shared by the SDN controllers with the users, this state is fully observable by all the agent, individually. Hence, the observable state of our model at time step k, is $\mathbf{s}_k = [B_k, G_k]^T$, where $B_k \in [0, B_{max}]$, and $G_k \in \mathcal{G}$, and $\mathcal{G}$ is a discretization of the interval of $[0, 1]$ into a finite number of possibilities, e.g., $\{LL, L, M, H, HH\}$.

- **Actions**

  In a client-side rate adaptation, agents choose the desired bitrate of the video chunks from a finite set of options put forward by the manifest file, and communicate it to the delivery server. We take this rate selection as the action of the agent. Formally speaking, the action taken by user n at a given time step k is it choice of rate $a_k^n = r_k^n \in \mathcal{A}_n$, where $\mathcal{A}_n$ is the set of discrete representations (bitrate levels) of the encoded video segment, available for user n. The joint action of all users would be denoted by $\mathbf{a}_k = \langle a_k^1, \ldots, a_k^N \rangle \in \mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_N$.

- **Transition Function**

  In order to specify the transition function: $T : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$, which provides a probability distribution of the new state given the current state and the joint action of the users, we use a probabilistic model for the available bandwidth at the bottleneck link. If the joint action of the users at time step k is $\mathbf{a} = \langle r_k^1, r_k^2, \ldots, r_k^N \rangle$, the probability that the new congestion level G^k falls into the interval $[\alpha, \beta]$, denoted by:

  $$P\left(a \le G_k \le b\right) = P\left(a \le \frac{\max(C, \sum_{n=1}^{N} r_k^n)}{C} \le b\right),$$

  is a straightforward function of the sum of the elements of joint action vector.

- **Reward Function**

  We assume that the common goal of all agents is to achieve and maintain an optimal allocation of the network bandwidth leading to an efficient and fair distribution of end-users' QoE. Using the social welfare function defined in Equation (5.3), we can specify the common reward function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ of our MMDP model as:

$$R = \mathbb{E}\left[\ \sum_{k=1}^{h} \gamma^k\ \Phi(\mathbf{u}^k)\ \right] \tag{7.3}$$

  where γ is the discount factor for future utilities and $\mathbf{u}_k = [u_k^1, u_k^2, \ldots, u_k^N]^T$ is the vector of utilities (i.e., QoE's) achieved for all N users at time step k as defined by Equation (7.2).

- **Horizon**

  Since MMDP has a feasible computational complexity compared to other multi-agent decision processes such as Dec-POMDP (see [112] for a partially observable model for multiuser video streaming), we are not confined by the computational tractability limit of the problem in choosing the horizon. We choose $h = 10$ for our MMDP model, to get a decent foresight capability over an extended horizon.

The MMDP model described above captures all four key components of our modeling approach and provides a QoE-driven, fairness-aware, foresighted optimization framework in a multi-agent setting for adaptive video streaming problem. To the best of our knowledge, no prior work has *explicitly* handled the desired goal of QoE-fairness. One novel aspect of our proposed method is that it directly incorporates QoE-fairness into the objective function of the multi-agent optimization problem. Using an MDP-based foresighted optimization, we also introduce an anticipatory and proactive behavior to the bitrate selection process which outperforms reactive and shortsighted adaptation schemes.

It is worth mentioning that finding a solution for MMDP is as computationally complex as solving a single agent MDP; they are both P-Complete [58]. This is much more tractable compared to Dec-POMDP, whose worst-case computational complexity is NEXP-complete, even with finite horizon. Taking the joint action space of an MMDP to be the set of basic actions, it could be viewed as a standard single-agent MDP [69]. Specifically, since there is a single reward function, the agents do not have competing interests; so any course of action is equally good for all. Therefore, the optimal joint policies over joint action space could be computed by solving the standard MDP using an algorithm like value iteration [18].

7.3 Performance Evaluation

7.3.1 Simulation Framework

We now evaluate and compare performance of MMDP rate adaptation with two state-of-the-art algorithms, PANDA [128] and FESTIVE [129], according to their default design parameters. FESTIVE has been the first adaptive video streaming algorithm specifically designed to address fairness among multiple DASH clients. PANDA also dynamically computes the segment inter-request time to address fairness issues, and video bitrate oscillations.

We use a MATLAB-based simulation framework which models the behavior of the above three rate adaptation methods, including their rate selection logic, video download process and buffer dynamics. At each time step t_k, the simulation calls the bitrate control and selection module of the algorithms to get the requested bitrate level r_k. It should be noted that all state updates from network and client players take place at the beginning of each time step.

In order to simulate the available bandwidth, we employ the 3G/HSDPA throughput traces [130]. This dataset contains logs from HTTP over TCP streaming sessions in a mobile wireless network in Norway. In each test, adaptive video streams were downloaded at maximum speed, with the video segment duration fixed to 2 seconds, which is the segment size for our simulation setup. Figure 7.2 shows the mean and variance of the available bandwidth used in our simulations. The average available bandwidth in this dataset is about $1.5Mbps$. In each simulation scenario, we multiply the values from a randomly picked ensemble of this available bandwidth dataset by the number of the users sharing the bottleneck to determine the bandwidth of the bottleneck link.

We assume that the length of video sequences are 10 minutes, consisting of 300 segments, each having $2sec$ duration and pre-encoded in the following bitrate levels: $\mathcal{R} = \{300Kbps, 700Kbps, 1000Kbps, 2000Kbps, 3000Kbps\}$. The playout buffer size is set to $B_{max} = 20s$ and the buffer level takes discrete integer values from 0 to B_{max}.

For solving our proposed MMDP model and finding the optimal policies for the agents, we used Multi-Agent Decision Process (MADP) Toolbox [131], which provides software tools for modeling, specifying, planning and learning a variety of decision-theoretic problems in multi-agent systems.

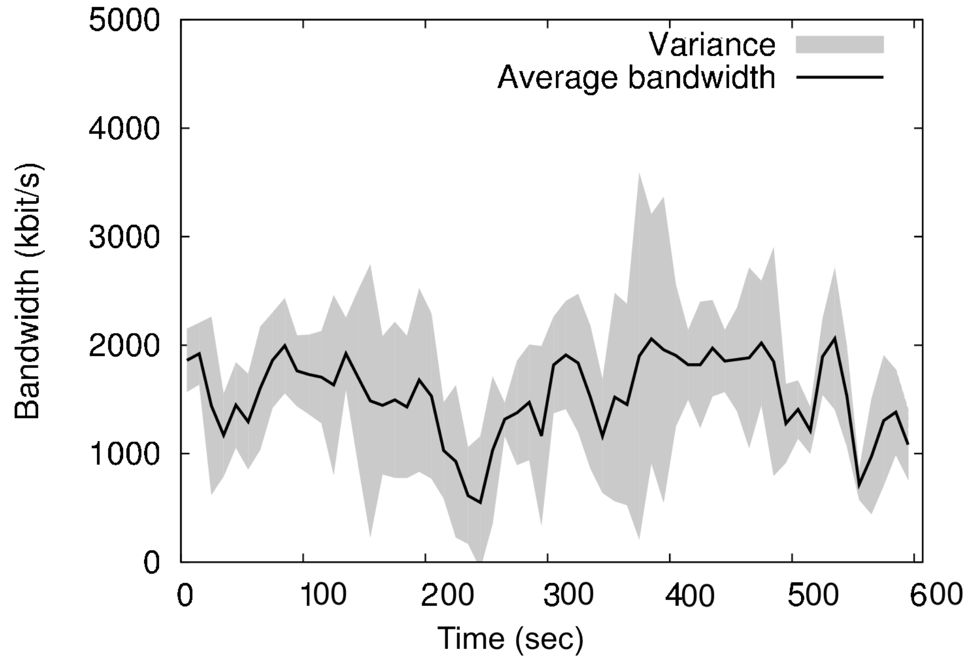

Figure 7.2: Measured traces of available bandwidth in a 3G/HSDPA mobile wireless network (Riiser2013)

7.3.2 Evaluation Metrics

We use two sets of evaluation metrics: *rate-based* and *QoE-based*. For the first set, we use the same metrics defined in [129] and inherited by [128] with a slight modification They are i) inefficiency, ii) unfairness, and iii) instability, defined based on the video bitrate fetched by competing players. If the bitrate requested by player n at time step k is denoted by r_k^n, then the rate-based metrics are defined as following:

- **Inefficiency Index**: Let C be the available bandwidth on the bottleneck link. [129] defined inefficiency index at time step k as:

  $$\frac{|\sum_n r_k^n - C|}{C}.$$

  Since the sum of requested bitrates $\sum_n r_k^n$ could sometimes be greater than C, [128] revised this index to:

  $$\frac{\max\left(0, C - \sum_n r_k^n\right)}{C}$$

  to avoid unnecessary penalty. We use the latter definition.

- **Unfairness Index**: Let $\mathbf{r}_k$ denote the vector of requested rates r_k^n of all players at time step k. If $J(\mathbf{r}_k)$ is the Jain's fairness index (see Equation (5.2)) over all rates, the unfairness index at time step k is defined as:

$$\sqrt{1 - J(\mathbf{r}_k)}.$$

- **Instability Index**: To measure the variation of requested bitrates, the instability index at time step k for each player n is defined as:

$$\frac{\sum_{d=0}^{K-1} \left| r_{k-d}^n - r_{k-d-1}^n \right| w(d)}{\sum_{d=0}^{K-1} r_{k-d}^n w(d)},$$

  where $w(d) = K - d$ is a weight function that puts more weight on more recent rate switchings, with a backward horizon of $K = 10$ time steps. Note that unlike the other two metrics, we have a separate instability index for each of the competing players. We will report the average over all agents.

In addition to the above rate-based metrics, we also evaluate the performance of the proposed rate adaptation scheme based on the QoE model presented in Section 7.1.2. We use the following three QoE-based metrics:

- **Average QoE**: Using Equation (7.2), we calculate the average QoE of all players, which is a measure of satisfaction level of users. Note that since our QoE model include terms related to average video quality as well as stability and smoothness of it, this metric is actually representing both efficiency and stability.

- **Fairness of QoE**: We are also interested in a fair distribution of QoE across all users. Let $\mathbf{u}$ denote the vector of utilities or QoE's achieved by all players. We use the Jain's index $J(\mathbf{u})$ as a measure of fairness of QoE.

- **Social Welfare**: Finally, we combine the efficiency, stability and fairness measures into one single social welfare metric, as defined in Equation (5.3) to evaluate the overall performance of the rate adaptation methods.

7.3.3 Simulation Results: Benchmarking

In the first scenario we investigate the behavior of our proposed MMDP rate adaptation scheme compared to two other methods, PANDA [128] and FESTIVE [129] when two DASH clients, using the same rate control algorithm, are sharing a bottleneck link with a varying available bandwidth as described before.

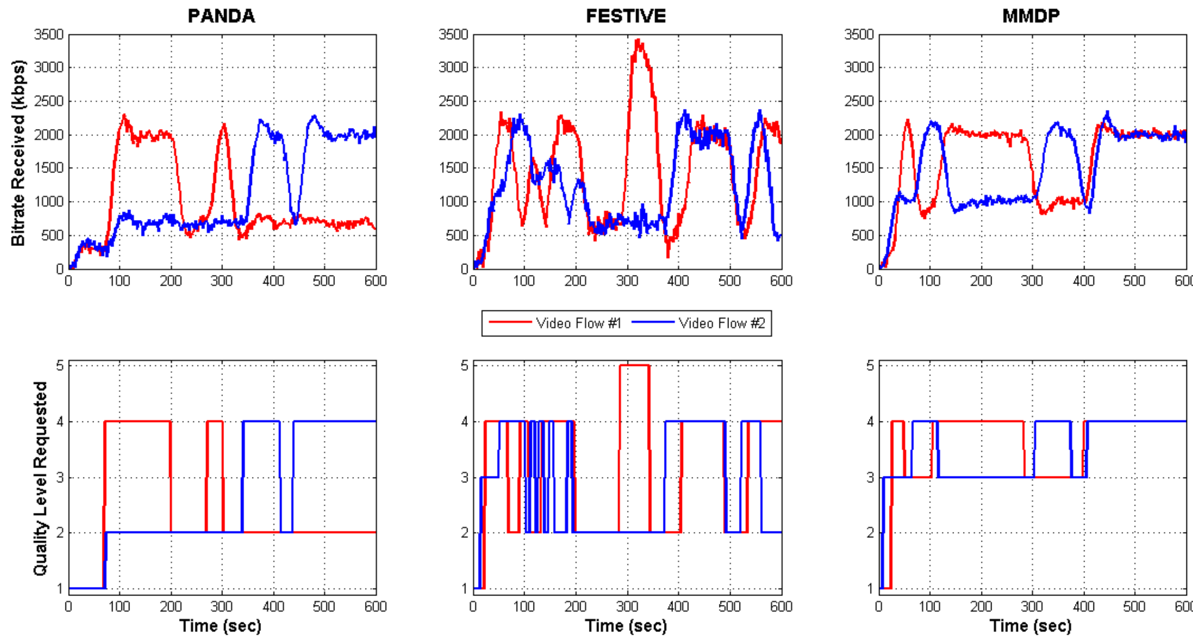

Figure 7.3: Sample behavior of different rate adaptation schemes when two DASH clients are sharing a bottleneck link

Figure 7.3 shows the dynamic behavior of all three rate adaptation algorithms in terms of the requested quality levels and received bitrates in one sample run. Our proposed MMDP method generally uses smaller jumps in choosing quality levels, compared to PANDA, and has fewer number of switches compared to FESTIVE. This is expected, as we have included penalties for quality switches in our QoE model, serving as the objective function of the optimization problem.

We carried out 20 runs of our simulation and calculated the average of bitrates chosen by each algorithm over the course of a video streaming session. Rate-based metrics are depicted in Figure 7.4, with lower values indicating better performance. We observe that all three methods perform more or less equally well in terms of rate-efficiency and rate-fairness, while MMDP outperform the other two in terms of rate-stability, as mentioned above.

Figure 7.5 shows the QoE-based metrics for the same scenario, with higher values indicating better performance. We observe significant improvements in all these metrics, compared to existing methods. Specifically, MMDP achieves an average QoE which is 12% and 18% higher than that of FESTIVE and PANDA, respectively. There is also an improvement in terms QoE-fairness, since Jain's index over the QoE of competing players goes beyond 0.8, with up to 24% increase compared to other methods. These improvements are also reflected in a higher value for social welfare metric with up to 25% gain.

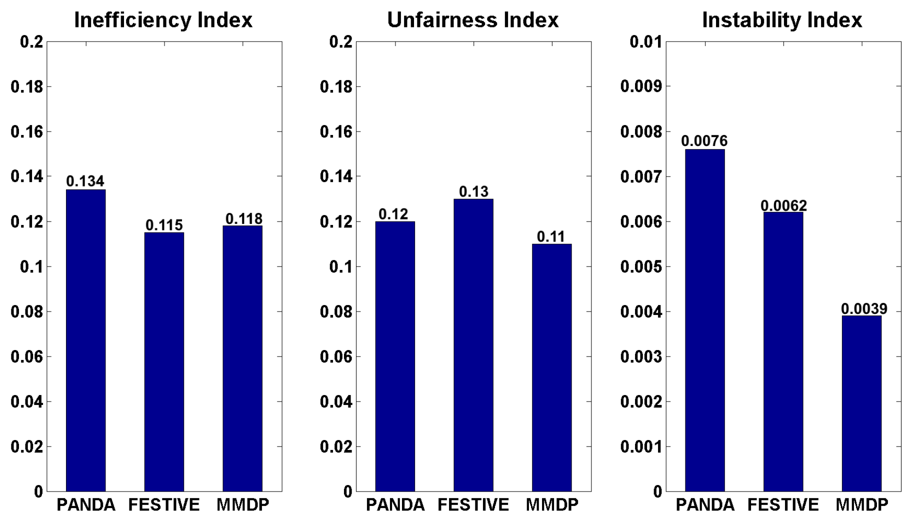

Figure 7.4: Comparison of rate-based metrics when two DASH clients are sharing a bottleneck link

7.3.4 Simulation Results: Robustness

In this scenario we investigate the robustness of our MMDP rate adaptation algorithm when i) the number of active agents sharing a bottleneck link varies, or ii) the standard deviation of the available bandwidth changes.

For the case of varying number of agents, to simulate the variable available bandwidth on the bottleneck link, we use the same real traces as before but multiply the values from a

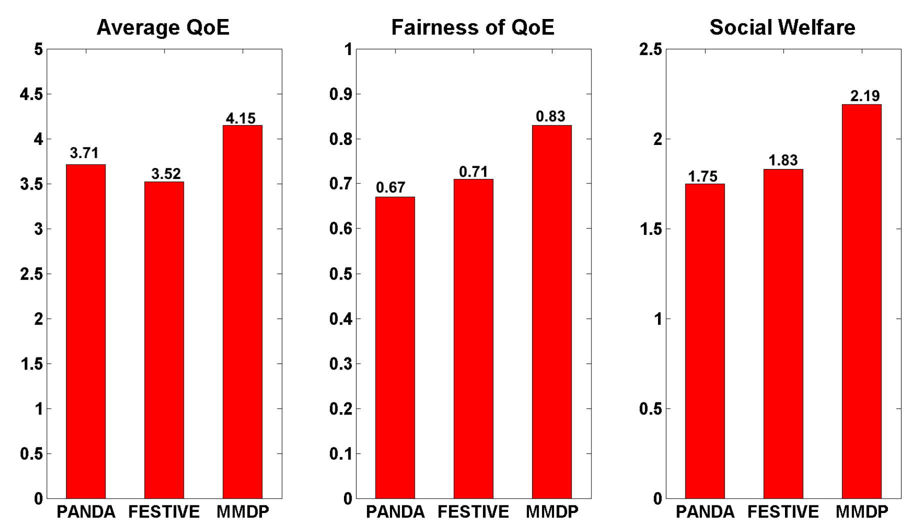

Figure 7.5: Comparison of QoE-based metrics when two DASH clients are sharing a bottleneck link

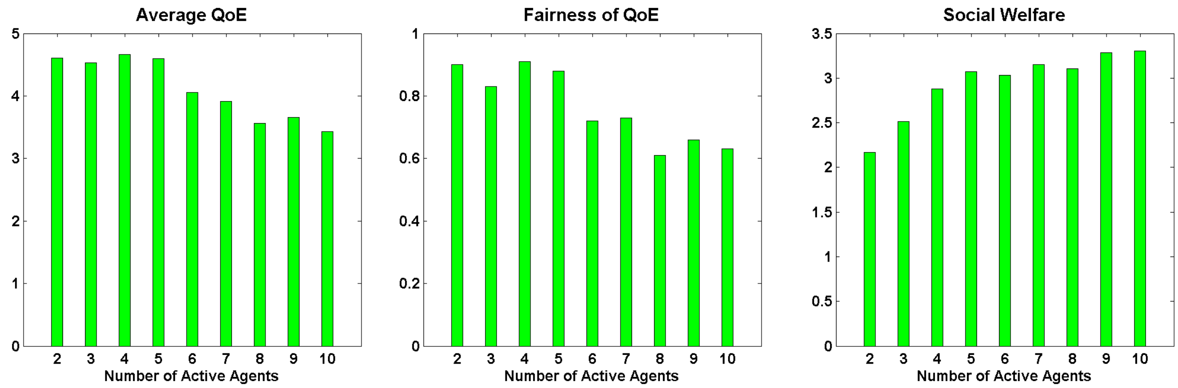

Figure 7.6: Robustness of MMDP's QoE-based metrics to varying number of active agents

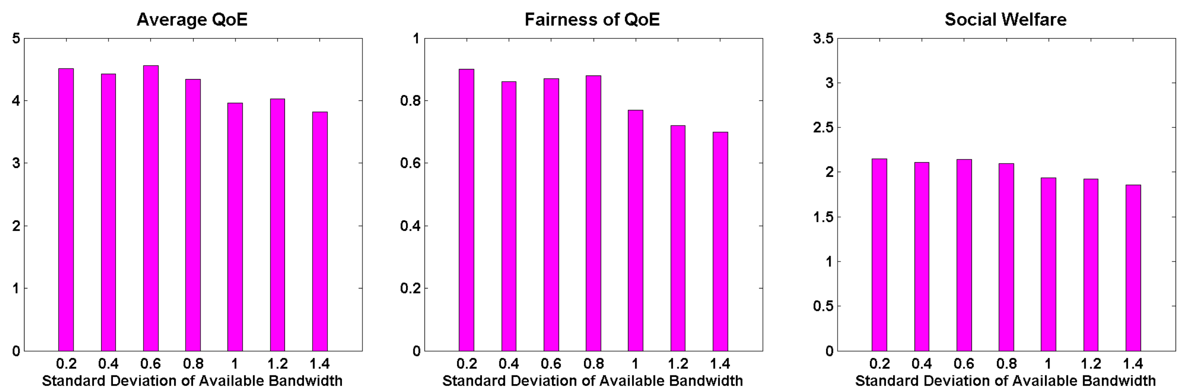

Figure 7.7: Robustness of MMDP's QoE-based metrics to different values of standard deviation of available bandwidth

randomly picked ensemble of the aforementioned dataset [130] by 10 and vary the number of concurrent users from 2 to 10. We run 20 runs of simulation, and in each run, all users arrive within the first 30 seconds of the simulation duration, which is 10 minutes.

Using the average of bitrates chosen by MMDP agents over all simulation runs, we calculate the QoE-based metrics, including the average and fairness of QoE. These metrics are plotted against the varying number of users in Figure 7.6. We observe that the proposed MMDP rate adaptation method has a robust performance against variation of number of users and is able to maintain satisfactory levels of efficiency and fairness at the same time. Note that social welfare metric depends on the *total* QoE of all users and hence, is generally increasing with the number of users, assuming the same average QoE for all cases. We also would like to mention that the fairly notable drop in fairness measure when the number of agents goes beyond 6, could be attributed to discreteness of video bitrate levels in HAS. In these cases, the bandwidth share available to each user falls somewhere between the bitrate levels, resulting in inevitable disparity between chosen bitrates of different users, and hence a higher unfairness.

The second robustness experiment involves changing the variability of the available bandwidth, which is a common characteristic of wireless networks. In this case, we use synthetic traces for available bandwidth by generating uniform distributions with a fixed mean of $2 \times 1.5Mbps = 3Mbps$ (for two agents sharing the bottleneck), and different values for standard deviation in the interval of $[0.2, 1.4]Mbps$. Again we run the simulation for 20 rounds for each distribution, and calculate the QoE-based metrics based on te average results over all runs.

Figure 7.7 shows the performance of the proposed MMDP method for different variability degrees of network available bandwidth. We observe that the rate allocation algorithm of MMDP is maintaining a satisfactory average QoE (≥ 3.7), fairness measure (≥ 0.7), and consistent values for social welfare across the entire range of standard deviations of available bandwidth. Given the extremely instable and volatile conditions of wireless networks, this form of robustness is a highly desired characteristic of any rate adaptation scheme.

7.4 Summary

Despite rapid growth of OTT services and video traffic, media delivery over the Internet is still facing many issues regarding fairness and stability of QoE in resource sharing situations. Some of these deficiencies regarding instability, unfairness, and even inefficiency stem from the distributed nature of existing resource allocation methods. As one of the most promising emerging network technologies, SDN allows for centralized and coordinated optimization to improve the QoE of video consumption.

In this chapter, we proposed a network-assisted SDN-based rate adaptation scheme by employing multi-agent decision process models to provide a coordinated solution to this inherently multi-agent problem. A multi-objective optimization problem with multiple decision-makers was developed, formulated and solved, seeking a fair and efficient distribution of QoE for all users. Performance evaluation of the proposed method shows its superiority compared to the top representatives of existing rate adaptation algorithms.

Chapter 8

Conclusion

We studies the problem of network bandwidth allocation among concurrent video streaming users and proposed new methods to provide some form of fairness across the users' QoE. In this concluding chapter, we summarize the contribution and achieved results and show possible directions for fututre research in this area.

8.1 Summary of Results

- Following the well-established practice of expressing the rate control problem in terms of NUM framework, we formulated the problem of bandwidth allocation for video flows in the context of NUM, using sigmoidal utility functions, rather than conventional but unrealistic concave functions. We used approximation algorithm for Sigmoidal Programming to solve the resulting nonconvex optimization problem, called NUM-SP. Simulation results for video streaming over a range of tree-shaped content delivery networks showed improvements of at least 60% in average utility/QoE and 45% in fairness, while using slightly less network resources, compared to two representative methods: Proportional Fair and Max-Min Fair.

- Based on a collaborative decision-theoretic approach to the problem of rate adaptation of multiple video streaming sessions, a social welfare function was developed to capture both fairness and efficiency objectives at the same time. Then, assuming a common altruistic goal for all network users, we proposed the novel framework of Social Utility Maximization (SUM), as opposed to the well-known Network Utility Maximization (NUM), by explicitly incorporating fairness into the objective function of the optimization problem.

- We proposed a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) model for the conventional IP networks and a Multi-agent Markov Decision Process (MMDP) model for the SDN-enabled wireless networks. By planning these cooperative decision process models, we determined the optimal network bandwidth allocation that leads to social welfare maximization.

- To reduce the computational complexity of planning decision processes and to alleviate the need for a complete stochastic model of the dynamic transitions of the environment, we developed a distributed multi-agent reinforcement learning algorithm as a low-complexity model-free solution to the posed optimization problems.

- Simulations of the proposed methods showed that the resulting optimal policies of the proposed decision processes outperform existing approaches in terms of both efficiency (total utility/QoE) and fairness. The Dec-POMDP model applied to a server-side rate adaptation resulted in 25% improvement in network bandwidth efficiency and 13% improvement in fairness index, compared to TFRC as a popular protocol of congestion control for multimedia streaming. Our performance evaluations also showed that the MMDP model applied to a client-side rate adaptation like DASH improves efficiency, fairness, and social welfare by as much as 18%, 24%, and 25%, respectively compared to two other representative existing methods, PANDA and FESTIVE.

8.2 Suggestions for Future Work

For future research in the area of NUM-SP, a number of extensions can be further studied. One is to enhance our method to dynamically handle churn: newly arriving and leaving flows. Another is to examine the feasibility of a parallel processing implementation of our algorithm for real-time performance. Finally, investigating the impact of having different sigmoidal utility functions of different users, rather than having the same function for all, will also make our algorithm more practical. This can be done in at least two ways. The simplest way is to increase or decrease the bitrate inflection point in the sigmoid function for different users. The other way is to use a function different than the logistic function to represent the sigmoid function. For example, a piecewise linear function approximating a sigmoid would more closely resemble the YouTube QoE utility function discovered in [96].

We would like to mention that although one could also think of using a staircase function consisting of multiple sigmoid functions connected one after the other, and each representing a different video quality (e.g., 720p, 1080p, 4K, etc.), this scheme is not appropriate because it does not match the real-world observations reported in the literature, which have found that even with multi-quality video, such as those used in YouTube, the video's utility function is still a single sigmoid [96].

The SUM framework offers lots of opportunities for further research. We used Jain's index as our measure of fairness to construct the social welfare function. Investigating other possibilities by incorporating different fairness measures into the objective function could be an interesting research avenue. Specifically, finding a good measure of fairness such that the resulting SUM becomes a convex optimization problem, would open many doors for inexpensively solving the SUM problem.

Even if the resulting SUM is nonconvex, there could exist ways to use decomposition methods and convert this macro-problem into a number of smaller subproblems to reduce its computational complexity.

The simulation setup for evaluation of our proposed decision processes could also be improved to a more realistic asynchronous setting, where actions (rate selection) of different agents are not forced to occur at synchronized time instances. Moreover, implementing a packet-level video trace-driven simulation of the proposed method is intended as the next step for performance evaluation.

APPENDICES

Appendix A

Description of Dec-POMDP Model in *dpomdp* Text Format

```
# Dec-POMDP Model of Multi-User Adaptive Rate Control for Video Streaming
# By: Mahdi Hemmati
#-------------------------------------------------------------------------
#
#Agents
#------
agents: 2
#Discount factor
#---------------
discount: 1.0
#Type of Values
#--------------
values: reward
#States (Congestion Level)
#-------------------------
states: CgLL CgL CgM CgH CgHH
#Initial state distribution
#---------------------------
start exclude: CgLL CgHH
#Actions (Sending Rate kbps)
#---------------------------
actions:
R1M R2M R3M
R1M R2M R3M
#Observations (Packet Loss Rate)
#-------------------------------
observations:
pLL pL pM pH pHH
pLL pL pM pH pHH
#Transition Probabilities
#------------------------
```

```
#  T: <a1 a2> :
#  matrix of values for <s> & <s'>
#    CgLL      CgL       CgM       CgH       CgHH
T: R1M R1M :
    0.3845    0.5515    0.0635    0.0005    0.0000
    0.0671    0.8661    0.0665    0.0003    0.0000
    0.0642    0.5527    0.3820    0.0011    0.0000
    0.1363    0.5864    0.2703    0.0070    0.0000
    0.1584    0.6818    0.1571    0.0027    0.0000
T: R1M R2M :
    0.0858    0.5999    0.3000    0.0143    0.0000
    0.0117    0.7369    0.2456    0.0058    0.0000
    0.0059    0.2456    0.7368    0.0117    0.0000
    0.0143    0.3000    0.6000    0.0857    0.0000
    0.0222    0.4666    0.4666    0.0444    0.0002
T: R1M R3M :
    0.0070    0.2702    0.5865    0.1351    0.0012
    0.0011    0.3820    0.5527    0.0637    0.0005
    0.0003    0.0665    0.8664    0.0665    0.0003
    0.0005    0.0637    0.5527    0.3820    0.0011
    0.0012    0.1351    0.5865    0.2702    0.0070
T: R2M R1M :
    0.0858    0.5999    0.3000    0.0143    0.0000
    0.0117    0.7369    0.2456    0.0058    0.0000
    0.0059    0.2456    0.7368    0.0117    0.0000
    0.0143    0.3000    0.6000    0.0857    0.0000
    0.0222    0.4666    0.4666    0.0444    0.0002
T: R2M R2M :
    0.0070    0.2702    0.5865    0.1351    0.0012
    0.0011    0.3820    0.5527    0.0637    0.0005
    0.0003    0.0665    0.8664    0.0665    0.0003
    0.0005    0.0637    0.5527    0.3820    0.0011
    0.0012    0.1351    0.5865    0.2702    0.0070
T: R2M R3M :
    0.0002    0.0444    0.4666    0.4666    0.0222
    0.0000    0.0857    0.6000    0.3000    0.0143
    0.0000    0.0117    0.7368    0.2456    0.0059
    0.0000    0.0059    0.2456    0.7368    0.0117
    0.0000    0.0143    0.3000    0.5999    0.0858
T: R3M R1M :
    0.0070    0.2702    0.5865    0.1351    0.0012
    0.0011    0.3820    0.5527    0.0637    0.0005
    0.0003    0.0665    0.8664    0.0665    0.0003
    0.0005    0.0637    0.5527    0.3820    0.0011
    0.0012    0.1351    0.5865    0.2702    0.0070
T: R3M R2M :
    0.0002    0.0444    0.4666    0.4666    0.0222
```

```
    0.0000   0.0857   0.6000   0.3000   0.0143
    0.0000   0.0117   0.7368   0.2456   0.0059
    0.0000   0.0059   0.2456   0.7368   0.0117
    0.0000   0.0143   0.3000   0.5999   0.0858
T: R3M R3M :
    0.0000   0.0027   0.1571   0.6818   0.1584
    0.0000   0.0070   0.2702   0.5865   0.1363
    0.0000   0.0011   0.3820   0.5527   0.0642
    0.0000   0.0003   0.0665   0.8661   0.0671
    0.0000   0.0005   0.0635   0.5515   0.3845
#Observation Probabilities
#-------------------------
#  O: <a1 a2> : <s'> : <o1 o2> : %f
O: * * : CgLL : pLL pLL : 0.8191
O: * * : CgLL : pLL pL : 0.0814
O: * * : CgLL : pLL pM : 0.0045
O: * * : CgLL : pLL pH : 0.0000
O: * * : CgLL : pLL pHH : 0.0000
O: * * : CgLL : pL pLL : 0.0814
O: * * : CgLL : pL pL : 0.0081
O: * * : CgLL : pL pM : 0.0005
O: * * : CgLL : pL pH : 0.0000
O: * * : CgLL : pL pHH : 0.0000
O: * * : CgLL : pM pLL : 0.0045
O: * * : CgLL : pM pL : 0.0005
O: * * : CgLL : pM pM : 0.0000
O: * * : CgLL : pM pH : 0.0000
O: * * : CgLL : pM pHH : 0.0000
O: * * : CgLL : pH pLL : 0.0000
O: * * : CgLL : pH pL : 0.0000
O: * * : CgLL : pH pM : 0.0000
O: * * : CgLL : pH pH : 0.0000
O: * * : CgLL : pH pHH : 0.0000
O: * * : CgLL : pHH pLL : 0.0000
O: * * : CgLL : pHH pL : 0.0000
O: * * : CgLL : pHH pM : 0.0000
O: * * : CgLL : pHH pH : 0.0000
O: * * : CgLL : pHH pHH : 0.0000
O: * * : CgL : pLL pLL : 0.0090
O: * * : CgL : pLL pL : 0.0770
O: * * : CgL : pLL pM : 0.0086
O: * * : CgL : pLL pH : 0.0004
O: * * : CgL : pLL pHH : 0.0000
O: * * : CgL : pL pLL : 0.0770
O: * * : CgL : pL pL  : 0.6561
O: * * : CgL : pL pM  : 0.0729
O: * * : CgL : pL pH  : 0.0041
```

```
O: * * : CgL : pL pHH : 0.0000
O: * * : CgL : pM pLL : 0.0086
O: * * : CgL : pM pL  : 0.0729
O: * * : CgL : pM pM  : 0.0081
O: * * : CgL : pM pH  : 0.0004
O: * * : CgL : pM pHH : 0.0000
O: * * : CgL : pH pLL : 0.0004
O: * * : CgL : pH pL  : 0.0041
O: * * : CgL : pH pM  : 0.0004
O: * * : CgL : pH pH  : 0.0000
O: * * : CgL : pH pHH : 0.0000
O: * * : CgL : pHH pLL : 0.0000
O: * * : CgL : pHH pL : 0.0000
O: * * : CgL : pHH pM : 0.0000
O: * * : CgL : pHH pH : 0.0000
O: * * : CgL : pHH pHH : 0.0000
O: * * : CgM : pLL pLL : 0.0000
O: * * : CgM : pLL pL : 0.0004
O: * * : CgM : pLL pM : 0.0042
O: * * : CgM : pLL pH : 0.0004
O: * * : CgM : pLL pHH : 0.0000
O: * * : CgM : pL pLL : 0.0004
O: * * : CgM : pL pL  : 0.0081
O: * * : CgM : pL pM  : 0.0729
O: * * : CgM : pL pH  : 0.0081
O: * * : CgM : pL pHH : 0.0004
O: * * : CgM : pM pLL : 0.0042
O: * * : CgM : pM pL  : 0.0729
O: * * : CgM : pM pM  : 0.6560
O: * * : CgM : pM pH  : 0.0729
O: * * : CgM : pM pHH : 0.0042
O: * * : CgM : pH pLL : 0.0004
O: * * : CgM : pH pL  : 0.0081
O: * * : CgM : pH pM  : 0.0729
O: * * : CgM : pH pH  : 0.0081
O: * * : CgM : pH pHH : 0.0004
O: * * : CgM : pHH pLL : 0.0000
O: * * : CgM : pHH pL : 0.0004
O: * * : CgM : pHH pM : 0.0042
O: * * : CgM : pHH pH : 0.0004
O: * * : CgM : pHH pHH : 0.0000
O: * * : CgH : pLL pLL : 0.0000
O: * * : CgH : pLL pL : 0.0000
O: * * : CgH : pLL pM : 0.0000
O: * * : CgH : pLL pH : 0.0000
O: * * : CgH : pLL pHH : 0.0000
O: * * : CgH : pL pLL : 0.0000
```

```
O: * * : CgH : pL pL  : 0.0000
O: * * : CgH : pL pM  : 0.0004
O: * * : CgH : pL pH  : 0.0041
O: * * : CgH : pL pHH : 0.0004
O: * * : CgH : pM pLL : 0.0000
O: * * : CgH : pM pL  : 0.0004
O: * * : CgH : pM pM  : 0.0081
O: * * : CgH : pM pH  : 0.0729
O: * * : CgH : pM pHH : 0.0086
O: * * : CgH : pH pLL : 0.0000
O: * * : CgH : pH pL  : 0.0041
O: * * : CgH : pH pM  : 0.0729
O: * * : CgH : pH pH  : 0.6561
O: * * : CgH : pH pHH : 0.0770
O: * * : CgH : pHH pLL : 0.0000
O: * * : CgH : pHH pL : 0.0004
O: * * : CgH : pHH pM : 0.0086
O: * * : CgH : pHH pH : 0.0770
O: * * : CgH : pHH pHH : 0.0090
O: * * : CgHH : pLL pLL : 0.0000
O: * * : CgHH : pLL pL : 0.0000
O: * * : CgHH : pLL pM : 0.0000
O: * * : CgHH : pLL pH : 0.0000
O: * * : CgHH : pLL pHH : 0.0000
O: * * : CgHH : pL pLL : 0.0000
O: * * : CgHH : pL pL : 0.0000
O: * * : CgHH : pL pM : 0.0000
O: * * : CgHH : pL pH : 0.0000
O: * * : CgHH : pL pHH : 0.0000
O: * * : CgHH : pM pLL : 0.0000
O: * * : CgHH : pM pL : 0.0000
O: * * : CgHH : pM pM : 0.0000
O: * * : CgHH : pM pH : 0.0005
O: * * : CgHH : pM pHH : 0.0045
O: * * : CgHH : pH pLL : 0.0000
O: * * : CgHH : pH pL : 0.0000
O: * * : CgHH : pH pM : 0.0005
O: * * : CgHH : pH pH : 0.0081
O: * * : CgHH : pH pHH : 0.0814
O: * * : CgHH : pHH pLL : 0.0000
O: * * : CgHH : pHH pL : 0.0000
O: * * : CgHH : pHH pM : 0.0045
O: * * : CgHH : pHH pH : 0.0814
O: * * : CgHH : pHH pHH : 0.8191
#Rewards
#-------
#  R: <a1 a2> : <s> : <s'> : <o1 o2> : %f
```

```
R: R1M R1M : CgLL : * : * : 1.8855
R: R1M R1M : CgL : * : * : 1.8831
R: R1M R1M : CgM : * : * : 1.8591
R: R1M R1M : CgH : * : * : 1.6343
R: R1M R1M : CgHH : * : * : 0.7411
R: R1M R2M : CgLL : * : * : 2.0058
R: R1M R2M : CgL : * : * : 2.0012
R: R1M R2M : CgM : * : * : 1.9550
R: R1M R2M : CgH : * : * : 1.5495
R: R1M R2M : CgHH : * : * : 0.7172
R: R1M R3M : CgLL : * : * : 2.0158
R: R1M R3M : CgL : * : * : 2.0101
R: R1M R3M : CgM : * : * : 1.9541
R: R1M R3M : CgH : * : * : 1.4838
R: R1M R3M : CgHH : * : * : 0.7171
R: R2M R1M : CgLL : * : * : 2.0058
R: R2M R1M : CgL : * : * : 2.0012
R: R2M R1M : CgM : * : * : 1.9550
R: R2M R1M : CgH : * : * : 1.5495
R: R2M R1M : CgHH : * : * : 0.7172
R: R2M R2M : CgLL : * : * : 2.1260
R: R2M R2M : CgL : * : * : 2.1191
R: R2M R2M : CgM : * : * : 2.0508
R: R2M R2M : CgH : * : * : 1.4648
R: R2M R2M : CgHH : * : * : 0.6933
R: R2M R3M : CgLL : * : * : 2.1359
R: R2M R3M : CgL : * : * : 2.1280
R: R2M R3M : CgM : * : * : 2.0499
R: R2M R3M : CgH : * : * : 1.3988
R: R2M R3M : CgHH : * : * : 0.6932
R: R3M R1M : CgLL : * : * : 2.0158
R: R3M R1M : CgL : * : * : 2.0101
R: R3M R1M : CgM : * : * : 1.9541
R: R3M R1M : CgH : * : * : 1.4838
R: R3M R1M : CgHH : * : * : 0.7171
R: R3M R2M : CgLL : * : * : 2.1359
R: R3M R2M : CgL : * : * : 2.1280
R: R3M R2M : CgM : * : * : 2.0499
R: R3M R2M : CgH : * : * : 1.3988
R: R3M R2M : CgHH : * : * : 0.6932
R: R3M R3M : CgLL : * : * : 2.1459
R: R3M R3M : CgL : * : * : 2.1369
R: R3M R3M : CgM : * : * : 2.0490
R: R3M R3M : CgH : * : * : 1.3328
R: R3M R3M : CgHH : * : * : 0.6932
#
```

Listing A.1: Text Description of Dec-POMDP Model

References

[1] "Cisco Visual Networking Index: Forecast and Methodology, 2015-2020," 2016.

[2] "Global Internet Phenomena Report," 2016. [Online]. Available: https://www.sandvine.com/trends/global-internet-phenomena/

[3] A. C. Begen, T. Akgul, and M. Baugher, "Watching video over the web: Part 1: Streaming protocols," *IEEE Internet Computing*, vol. 15, no. 2, pp. 54–63, 2011.

[4] T. Stockhammer, "Dynamic Adaptive Streaming over HTTP – Design Principles and Standards," in *Proc. ACM Conf. on Multimedia Systems*, 2011, pp. 133–134.

[5] C. Zhou, C.-w. Lin, X. Zhang, and Z. Guo, "A Control-Theoretic Approach to Rate Adaption for DASH over Multiple Content Distribution Servers," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 4, pp. 681–694, 2014.

[6] F. Chiariotti, S. D'Aronco, L. Toni, P. Frossard, S. D'Aronco, L. Toni, and P. Frossard, "Online learning adaptation strategy for DASH clients," in *Proc. 7th ACM Int. Conf. on Multimedia Systems (MMSys)*, 2016, p. 8.

[7] S. Petrangeli, J. Famaey, M. Claeys, S. Latre, and F. De Turck, "QoE-Driven Rate Adaptation Heuristic for Fair Adaptive Video Streaming," *ACM Trans. on Multimedia Computing, Communications, and Applications*, vol. 12, no. 2, pp. 28:1–28:24, 2015.

[8] A. Bokani, M. Hassan, S. Kanhere, and X. Zhu, "Optimizing HTTP-Based Adaptive Streaming in Vehicular Environment Using Markov Decision Process," *IEEE Trans. on Multimedia*, vol. 17, no. 12, pp. 2297–2309, 2015.

[9] C. Zhou, C.-W. Lin, and Z. Guo, "mDASH: A Markov Decision based Rate Adaptation Approach for Dynamic HTTP Streaming," *IEEE Transactions on Multimedia*, vol. 18, no. 4, pp. 1–1, 2016.

[10] T.-Y. Huang, R. Johari, N. Mckeown, M. Trunnell, and M. Watson, "A Buffer-Based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service," in *Proc. ACM SIGCOMM*, 2014.

[11] DARPA Internet Program, "Transmission Control Protocol (TCP): Protocol Specification (Internt Standard)," 1981. [Online]. Available: https://tools.ietf.org/html/rfc793

[12] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification (Proposed Standard)," 2008. [Online]. Available: https://tools.ietf.org/html/rfc5348

[13] R. Rejaie, M. Handley, and D. Estrin, "Quality Adaptation for Congestion Controlled Playback Video over the Internet," 1999.

[14] L. Cai, X. Shen, J. Pan, and J. Mark, "Performance analysis of TCP-friendly AIMD algorithms for multimedia applications," *IEEE Transactions on Multimedia*, vol. 7, no. 2, pp. 339–355, 2005.

[15] N. Wang, K. Ho, G. Pavlou, and M. Howarth, "An overview of routing optimization for internet traffic engineering," *IEEE Communications Surveys & Tutorials*, vol. 10, pp. 36–56, 2008.

[16] H.-P. Shiang and M. van der Schaar, "A Quality-Centric TCP-Friendly Congestion Control for Multimedia Transmission," *IEEE Transactions on Multimedia*, vol. 14, no. 3, pp. 896–909, jun 2012.

[17] A. G. Barto, R. S. Sutton, and C. Watkins, "Learning and sequential decision making," University of Massachusetts, Tech. Rep., 1989.

[18] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. The MIT Press, 1998.

[19] D. P. Bertsekas, *Dynamic programming and optimal control*. Athena Scientific Belmont, MA, 1995, vol. I-II.

[20] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, no. 1-2, pp. 99–134, may 1998.

[21] D. Bansal and H. Balakrishnan, "TCP-friendly Congestion Control for Real-time Streaming Applications," Technical Report MIT-LCS-TR-806, MIT Laboratory for Computer Science, Tech. Rep. May, 2000.

[22] ——, "Binomial congestion control algorithms," in *Proc. IEEE INFOCOM*, vol. 2, 2001, pp. 631–640.

[23] D. Bansal, H. Balakrishnan, S. Floyd, and S. Shenker, "Dynamic Behavior of Slowly-Responsive Congestion," *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 4, pp. 263–274, 2001.

[24] R. Rejaie, M. Handley, and D. Estrin, "RAP: An end-to-end rate-based congestion control mechanism for realtime streams in the Internet," in *Proc. IEEE INFOCOM*, vol. 3, 1999, pp. 1337–1345.

[25] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-Based Congestion Control for Unicast Applications," *ACM SIGCOMM Computer Communication Review*, vol. 30, no. 4, pp. 43–56, 2000.

[26] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," *ACM SIGCOMM Computer Communication Review*, vol. 28, no. 4, pp. 303–314, 1998.

[27] E. Kohler, M. Handley, and S. Floyd, "Designing DCCP: Congestion control without reliability," in *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4. ACM, 2006, pp. 27–38.

[28] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," 2003. [Online]. Available: https://www.ietf.org/rfc/rfc3550.txt

[29] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, no. 3, pp. 237–252, 1998.

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

[31] H. Shi, R. V. Prasad, E. Onur, and I. Niemegeers, "Fairness in Wireless Networks:Issues, Measures and Challenges," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 5–24, 2014.

[32] Radunović, Bozidar and J.-Y. L. Boudec, "A unified framework for max-min and min-max fairness with applications," *IEEE/ACM Trans. on Networking*, vol. 15, no. 5, pp. 1073–1083, 2007.

[33] J. M. Jaffe, "Bottleneck flow control," *IEEE Trans. on Communications*, vol. 29, no. 7, pp. 954–962, 1981.

[34] E. L. Hahne, "Round-robin scheduling for max-min fairness in data networks," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 7, pp. 1024–1039, 1991.

[35] H. Jiang and W. Zhuang, "Effective packet scheduling with fairness adaptation in ultra-wideband wireless networks," *IEEE Trans. on Wireless Communications*, vol. 6, no. 2, pp. 680–690, 2007.

[36] R. K. Guha, C. Gunter, S. Sarkar, and Others, "Fair coalitions for power-aware routing in wireless networks," *IEEE Trans. on Mobile Computing*, vol. 6, no. 2, pp. 206–220, 2007.

[37] P. Wang, H. Jiang, W. Zhuang, and H. V. Poor, "Redefinition of max-min fairness in multi-hop wireless networks," *IEEE Trans. on Wireless Communications*, vol. 7, no. 12, pp. 4786–4791, 2008.

[38] W. Saad, Z. Han, M. Debbah, and A. Hjørungnes, "A distributed coalition formation framework for fair user cooperation in wireless networks," *IEEE Trans. on Wireless Communications*, vol. 8, no. 9, pp. 4580–4593, 2009.

[39] H. T. Cheng and W. Zhuang, "An optimization framework for balancing throughput and fairness in wireless networks with QoS support," *IEEE Trans. on Wireless Communications*, vol. 7, no. 2, pp. 584–593, 2008.

[40] J. W. Wong, J. P. Sauvé, J. Field, and Others, "A study of fairness in packet-switching networks," *IEEE Trans. on Communications*, vol. 30, no. 2, pp. 346–353, 1982.

[41] C. E. Koksal, H. Kassab, and H. Balakrishnan, "An analysis of short-term fairness in wireless media access protocols (poster session)," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 28, no. 1. ACM, 2000, pp. 118–119.

[42] M. Dianati, X. Shen, and S. Naik, "A new fairness index for radio resource allocation in wireless networks," in *Proc. IEEE Wireless Communications and Networking Conference*, vol. 2, 2005, pp. 712–717.

[43] M. Bredel and M. Fidler, "Understanding fairness and its impact on quality of service in IEEE 802.11," in *Proc. IEEE INFOCOM*, 2009, pp. 1098–1106.

[44] R. K. Jain, D.-M. W. Chiu, and W. R. Hawe, "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer System," Eastern Research Lab, Tech. Rep., 1984.

[45] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 556–567, 2000.

[46] T. Lan, D. Kao, M. Chiang, A. Sabharwal, and M. Hiang, "An Axiomatic Theory of Fairness in Network Resource Allocation," in *Proc. IEEE INFOCOM*, 2010, pp. 1–9.

[47] Y. Chen, K. Wu, and Q. Zhang, "From QoS to QoE: A tutorial on video quality assessment," *IEEE Communications Surveys and Tutorials*, vol. 17, no. 2, pp. 1126–1165, 2015.

[48] A. Balachandran, V. Sekar, A. Akella, S. Seshan, I. Stoica, and H. Zhang, "Developing a Predictive Model of Quality of Experience for Internet Video," in *Proc. ACM SIGCOMM*, 2013, p. 339.

[49] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hossfeld, and P. Tran-gia, "A Survey on Quality of Experience of HTTP Adaptive Streaming," *IEEE Communication Surveys & Tutorials*, vol. 17, no. 1, pp. 469–492, 2015.

[50] V. Joseph, G. de Veciana, and A. Arapostathis, "Resource Allocation: Realizing Mean-Variability-Fairness Tradeoffs," in *Proc. 50th Annual Allerton Conference on Communication, Control, and Computing*, 2012, pp. 831–838.

[51] H. Nam, K.-H. Kim, and H. Schulzrinne, "QoE matters more than QoS: Why people stop watching cat videos," in *Proc. IEEE INFOCOM*. IEEE, apr 2016, pp. 1–9.

[52] J. De Vriendt, D. De Vleeschauwer, and D. Robinson, "Model for estimating QoE of video delivered using HTTP adaptive streaming," in *Proc. IFIP/IEEE Int. Symposium on Integrated Network Management*, 2013, pp. 1288–1293.

[53] M. Wiering and M. V. Otterlo, Eds., *Reinforcement Learning: State-of-the-Art*. Springer, 2012.

[54] Y. Shoham and K. Leyton-brown, *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*, 1st ed., 2009.

[55] F. a. Oliehoek, "Decentralized POMDPs," in *Reinforcement Learning: State-of-the-Art*, M. Wiering and M. V. Otterlo, Eds. Springer, 2012, pp. 471–503.

[56] D. S. Bernstein, S. Zilberstein, and N. Immerman, "The Complexity of Decentralized Control of Markov Decision Processes," in *Proc. Uncertainty in Artifical Intelligence*, 2000, pp. 32–37.

[57] Z. Rabinovich, C. V. Goldman, and J. S. Rosenschein, "The Complexity of Multiagent Systems: The Price of Silence," in *Proc. AAMAS*, 2003, pp. 1102–1103.

[58] C. Amato, G. Chowdhary, A. Geramifard, N. K. Ure, and M. J. Kochenderfer, "Decentralized Control of Partially Observable Markov Decision Processes," in *Proc. IEEE CDC*, 2013, pp. 2398–2405.

[59] D. S. Bernstein, E. A. Hansen, S. Zilberstein, and C. Amato, "Dynamic programming for partially observable stochastic games," in *Proc. AAAI*, 2004, pp. 547–552.

[60] D. Szer and F. Charpillet, "Point-based dynamic programming for DEC-POMDPs," in *Proc. AAAI*, 2006, pp. 1233–1238.

[61] S. Seuken and S. Zilberstein, "Memory-Bounded Dynamic Programming for DEC-POMDPs." in *Proc. Int. Joint Conf. on Artificial Intelligence*, 2007, pp. 2009–2015.

[62] A. Boularias and B. Chaib-draa, "Exact Dynamic Programming for Decentralized POMDPs with Lossless Policy Compression," in *Int. Conf. on Automated Planning and Scheduling (ICAPS)*, 2008, pp. 20–27.

[63] D. Szer and I. Lorraine, "MAA*: A Heuristic Search Algorithm for Solving Decentralized POMDPs," in *Proc. Conf. on Uncertainty in Artificial Intelligence*, 2005, pp. 576–583.

[64] F. a. Oliehoek, M. T. J. Spaan, and N. Vlassis, "Optimal and approximate Q-value functions for decentralized POMDPs," *Journal of Artificial Intelligence Research*, vol. 32, pp. 289–353, 2008.

[65] F. a. Oliehoek, M. T. J. Spaan, C. Amato, and S. Whiteson, "Incremental clustering and expansion for faster optimal planning in decentralized POMDPs," *Journal of Artificial Intelligence Research*, vol. 46, pp. 449–509, 2013.

[66] R. Nair, M. Tambe, M. Yokoo, D. Pynadath, and S. Marsella, "Taming decentralized POMDPs: Towards efficient policy computation for multiagent settings," in *Proc. IJCAI*, 2003, pp. 705–711.

[67] M. J. Osborne and A. Rubinstein, *A Course in Game Theory.* MIT Press, 1994.

[68] C. Boutilier, "Planning, learning and coordination in multiagent decision processes," *Proceedings of the 6th conference on Theoretical aspects of rationality and knowledge*, pp. 195–210, 1996.

[69] ——, "Sequential optimality and coordination in multiagent systems," in *Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI)*, vol. 1, 1999, pp. 478–485.

[70] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1617–1634, 2014.

[71] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," *Proceedings of The IEEE*, vol. 103, no. 1, pp. 14–76, 2015.

[72] A. Yassine, H. Rahimi, and S. Shirmohammadi, "Software defined network traffic measurement: Current trends and challenges," *IEEE Instrumentation & Measurement Magazine*, vol. 18, no. 2, pp. 42–50, apr 2015.

[73] K. Pentikousis, Y. Wang, and W. Hu, "Mobileflow: Toward software-defined mobile networks," *IEEE Communications Magazine*, vol. 51, no. 7, pp. 44–53, 2013.

[74] H. Nam, D. Calin, and H. Schulzrinne, "Intelligent Content Delivery over Wireless via SDN," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC).* IEEE, 2015, pp. 2185 – 2190.

[75] O. Habachi, Y. Hu, M. van der Schaar, Y. Hayel, and F. Wu, "MOS-Based Congestion Control for Conversational Services in Wireless Environments," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 7, pp. 1225–1236, aug 2012.

[76] N. Changuel, B. Sayadi, and M. Kieffer, "Online learning for QoE-based video streaming to mobile receivers," in *Proc. IEEE Globecom Workshops.* IEEE, dec 2012, pp. 1319–1324.

[77] M. Claeys, S. Latré, J. Famaey, T. Wu, W. Van Leekwijck, and F. De Turck, "Design and optimisation of a (FA) Q-learning-based HTTP adaptive streaming client," *Connection Science*, vol. 26, no. 1, pp. 25–43, 2014.

[78] B. Briscoe, "Flow rate fairness: Dismantling a religion," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 2, pp. 63–74, 2007.

[79] J. Jiang, V. Sekar, and H. Zhang, "Improving Fairness, Efficiency, and Stability in HTTP-Based Adaptive Video Streaming With FESTIVE," *IEEE/ACM Transactions on Networking*, vol. 22, no. 1, pp. 326–340, 2014.

[80] V. Joseph, G. de Veciana, and A. Arapostathis, "Resource Allocation: Realizing Mean-Variability-Fairness Tradeoffs," *IEEE Transactions on Automatic Control*, vol. 60, no. 1, pp. 19–33, 2015.

[81] S. Akhshabi, L. Anantakrishnan, A. C. Begen, and C. Dovrolis, "What Happens when HTTP Adaptive Streaming Players Compete for Bandwidth?" in *Proceedings of the 22Nd International Workshop on Network and Operating System Support for Digital Audio and Video*, ser. NOSSDAV '12. New York, NY, USA: ACM, 2012, pp. 9–14.

[82] S. D'Aronco, L. Toni, and P. Frossard, "Price-Based Controller for Quality-Fair HTTP Adaptive Streaming," in *IEEE Int. Symposium on Multimedia (ISM)*. IEEE, dec 2016, pp. 113–118.

[83] A. Bentaleb, A. C. Begen, and R. Zimmermann, "SDNDASH: Improving QoE of HTTP Adaptive Streaming Using Software Defined Networking," in *Proc. ACM Multimedia*, 2016, pp. 1296–1305.

[84] S. Cicalo, N. Changuel, V. Tralli, B. Sayadi, F. Faucheux, and S. Kerboeuf, "Improving QoE and Fairness in HTTP Adaptive Streaming over LTE Network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8215, no. c, pp. 1–1, 2015.

[85] X. Zhu, R. Pan, M. S. Prabhu, N. Dukkipati, V. Subramanian, and F. Bonomi, "Layered Internet video adaptation (LIVA): Network-assisted bandwidth sharing and transient loss protection for video streaming," *IEEE Transactions on Multimedia*, vol. 13, no. 4, pp. 720–732, 2011.

[86] X. Liu, F. Dobrian, H. Milner, J. Jiang, V. Sekar, I. Stoica, and H. Zhang, "A case for a coordinated internet video control plane," *ACM SIGCOMM Computer Communication Review*, vol. 42, no. 4, p. 359, 2012.

[87] N. Bouten, S. Latré, and J. Famaey, "In-Network Quality Optimization for Adaptive Video Streaming Services," *IEEE Transactions on Multimedia*, vol. 16, no. 8, pp. 2281–2293, 2014.

[88] S. Ramakrishnan, X. Zhu, F. Chan, and K. Kambhatla, "SDN Based QoE Optimization for HTTP-Based Adaptive Video Streaming," in *IEEE International Symposium on Multimedia*, 2015, pp. 3–6.

[89] E. Thomas, M. O. V. Deventer, T. Stockhammer, A. C. Begen, M.-l. Champel, and O. Oyman, "Applications and Deployments of Server and Network Assisted DASH (SAND)," in *IBC*, 2016.

[90] P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race, "Towards network-wide QoE fairness using openflow-assisted adaptive video streaming," in *Proceedings of the 2013 ACM SIGCOMM workshop on Future human-centric Multimedia Networking*. New York, New York, USA: ACM Press, 2013, p. 15.

[91] H. Nam, K. Kim, J. Kim, and H. Schulzrinne, "Towards QoE-aware Video Streaming using SDN," in *Proc. IEEE GLOBECOM*, 2014, pp. 1317–1322.

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

[93] S. Shenker, "Fundamental design issues for the future Internet," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 7, 1995.

[94] J. W. Lee, R. R. Mazumdar, and N. B. Shroff, "Non-convex optimization and rate control for multi-class services in the internet," *IEEE/ACM Transactions on Networking*, vol. 13, no. 4, pp. 827–840, 2005.

[95] P. Hande, S. Zhang, and M. Chiang, "Distributed Rate Allocation for Inelastic Flows," *IEEE/ACM Transactions on Networking*, vol. 15, no. 6, pp. 1240–1253, 2007.

[96] P. Casas, A. D'Alconzo, P. Fiadino, A. Bar, and A. Finamore, "On the analysis of QoE-based performance degradation in YouTube traffic," *Proc. 10th Int. Conf. on Network and Service Management (CNSM)*, pp. 1–9, 2014.

[97] M. Fazel and M. Chiang, "Network utility maximization with nonconcave utilities using sum-of-squares method," in *Proc. IEEE CDC and the European Control Conference*, no. 1, 2005, pp. 1867–1874.

[98] M. Udell and S. Boyd, "Maximizing a Sum of Sigmoids," pp. 1–25, 2014.

[99] ——, "Bounding Duality Gap for Problems with Separable Objective," *arXiv:1410.4158v1*, pp. 1–24, 2014.

[100] M. Udell, K. Mohan, D. Zeng, J. Hong, S. Diamond, and S. Boyd, “Convex Optimization in Julia,” in *Proc. First Workshop for High Performance Technical Computing in Dynamic Languages*, 2014, pp. 18–28.

[101] F. Wilson, I. Wakeman, and W. Smith, “Quality of service parameters for commercial application of video telephony,” in *Proc. Human Factors in Telecommunication Symposium*, Darmstadt, Germany, 1993.

[102] Mung Chiang, “Nonconvex Optimization for Communication Networks,” in *Advances in Mechanics and Mathematics*, D. Y. Gao and H. D. Sherali, Eds., 2010, vol. III, ch. 5, pp. 137–196.

[103] G. Tychogiorgos and K. K. Leung, “Optimization-based resource allocation in communication networks,” *Computer Networks*, vol. 66, pp. 32–45, 2014.

[104] E. Lawler and D. Wood, “Branch-and-bound methods: A survey,” *Operations Research*, vol. 14, no. 4, pp. 699–719, 1966.

[105] M. Udell, “Sigmoidal Programming,” 2014. [Online]. Available: https://github.com/madeleineudell/SigmoidalProgramming.jl

[106] M. Hemmati, B. McCormick, and S. Shirmohammadi, “Fair and Efficient Bandwidth Allocation for Video Flows Using Sigmoidal Programming,” in *Proc. IEEE Int. Symposium on Multimedia (ISM)*, San Jose, California, USA, 2016.

[107] R. Johari and J. Tsitsiklis, “Network resource allocation and a congestion game,” in *Proc. Annual Allerton Conference on Communication, Control, and Computing*, 2003.

[108] L. Peshkin, K.-E. Kim, N. Meuleau, and L. P. Kaelbling, “Learning to Cooperate via Policy Search,” in *16th Conference on Uncertainty in Artificial Intelligence*, 2000, pp. 489–496.

[109] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, “A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP,” in *Proc. ACM SIGCOMM*, London, UK, 2015, pp. 325–338.

[110] W. Saad, Z. Han, M. Debbah, A. Hjorungnes, and T. Basar, “Coalitional game theory for communication networks,” *IEEE Signal Processing Magazine*, vol. 26, no. 5, pp. 77–97, 2009.

[111] C. Joe-Wong, S. Sen, T. Lan, and M. Chiang, "Multi-Resource Allocation: Fairness-Efficiency Tradeoffs in a Unifying Framework," in *Proc. IEEE INFOCOM*, mar 2012, pp. 1206–1214.

[112] M. Hemmati, A. Yassine, and S. Shirmohammadi, "A Dec-POMDP Model for Congestion Avoidance and Fair Allocation of Network Bandwidth in Rate-Adaptive Video Streaming," in *Proc. IEEE Symposium on Computational Intelligence for Communication Systems and Networks (CI Comms)*. Cape Town, South Africa: IEEE, 2015.

[113] ITU-T, "Recommendation ITU-T G.1070: Opinion model for video-telephony applications," Tech. Rep., 2012.

[114] ——, "Recommendation ITU-T G.1071: Opinion model for network planning of video and audio streaming applications," Tech. Rep., 2016.

[115] Huawei Technologies Co., "Server Management in Adaptive Streaming on the Internet," in *Proc. 4th W3C Web TV Workshop*, 2014, pp. 1–2.

[116] C. Barakat, P. Thiran, G. Iannaccone, C. Diot, and P. Owezarski, "Modeling Internet backbone traffic at the flow level," *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2111–2124, 2003.

[117] L. Busoniu, R. Babuška, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, vol. 38, no. 2, pp. 156–172, 2008.

[118] J. Lundén, S. Kulkarni, V. Koivunen, and H. V. Poor, "Multiagent Reinforcement Learning Based Spectrum Sensing Policies for Cognitive Radio Networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 5, pp. 858–868, 2013.

[119] S. Singh, T. Jaakkola, M. Littman, and C. Szepesvári, "Convergence results for single-step on-policy reinforcement-learning algorithms," *Machine Learning*, vol. 38, no. 3, pp. 287–308, 2000.

[120] F. A. Oliehoek, M. T. J. Spaan, and P. Robbel, "MultiAgent Decision Process (MADP) Toolbox 0.3," 2014.

[121] A. R. Cassandra, "Exact and Approximate Algorithms for Partially Observable Markov Decision Processes," Ph.D. dissertation, 1998.

[122] R. K. Panta, "Mobile Video Delivery : Challenges and Opportunities," *IEEE Internet Computing*, 2015.

[123] 3GPP, "Policy and charging control architecture: TS 23.203 V13.10.0," 2016.

[124] J. Costa-requena, J. L. Santos, V. F. Guasch, K. Ahokas, I. Ahmad, M. Liyanage, M. Ylianttila, and S. A. Nextel, "SDN and NFV Integration in Generalized Mobile Network Architecture," in *Proc. European Conference on Networks and Communications (EuCNC)*, 2015, pp. 154–158.

[125] A. Hakiri and P. Berthou, "Leveraging SDN for The 5G Networks: Trends, Prospects and Challenges," *arXiv preprint arXiv:1506.02876*, pp. 1–23, 2015.

[126] N. Theera-Ampornpunt, S. Bagchi, K. R. Joshi, and R. K. Panta, "Using big data for more dependability: a cellular network tale," in *Proc. ACM Workshop on Hot Topics in Dependable Systems (HotDep)*, 2013, pp. 1–5.

[127] C. Shi, K. Joshi, R. K. Panta, M. H. Ammar, and E. W. Zegura, "CoAST : Collaborative Application-Aware Scheduling of Last-Mile Cellular Traffic," in *MobiSys '14*, 2014, pp. 245–258.

[128] Z. Li, X. Zhu, J. Gahm, and R. Pan, "Probe and Adapt: Rate Adaptation for HTTP Video Streaming At Scale," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 4, pp. 719 – 733, 2014.

[129] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with FESTIVE," *IEEE/ACM Transactions on Networking*, vol. 22, no. 1, pp. 326–340, 2014.

[130] H. Riiser, P. Vigmostad, C. Griwodz, and P. Halvorsen, "Commute path bandwidth traces from 3G networks," *Proceedings of the 4th ACM Multimedia Systems Conference on - MMSys '13*, pp. 114–118, 2013.

[131] F. Oliehoek, M. Spaan, J. Messias, and P. Robbel, "MultiAgent Decision Process (MADP) Toolbox," 2015. [Online]. Available: https://github.com/MADPToolbox/MADP