# Optimizations and Economics of Multimedia Services

**TANG, Ming**

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Doctor of Philosophy
in
Information Engineering

The Chinese University of Hong Kong
September 2018

Abstract of thesis entitled:

Optimizations and Economics of Multimedia Services

Submitted by TANG, Ming

for the degree of Doctor of Philosophy

at The Chinese University of Hong Kong in September 2018

With the fast development of communication and information technologies, humans have been increasingly enjoying multimedia services over the Internet. In order to better understand and provide such services, this thesis studies two important aspects of the multimedia over the Internet — multimedia service provision and multimedia platform operation.

In the first part on the multimedia service provision, we study how to provide high quality-of-experience multimedia services. The study focuses on mobile network scenarios, which is quite challenging due to the heterogeneous and limited mobile device resources. To enhance the services, we propose crowdsourced resource sharing models that enable mobile users to form cooperative groups through device-to-device connections and share their resources for multimedia service provision. We start with communication resource sharing in a video streaming application scenario, where we propose a crowdsourced video streaming model that enables mobile users to share communication resources for video streaming downloading. Analyzing this model is challenging due to the asynchronous downloading behaviors of the video users and the private user information (e.g., their video buffer sizes). Overcoming the challenges, we design an online algorithm that approaches to the system theoretical best performance, and further design auction-based incentive mechanisms (to motivate user cooperation) that achieve truthful user information revelation and efficient resource allocation. We further study a joint communication, computation, and caching resources sharing in a general multimedia scenario, where we propose a joint sharing framework of three kinds of resources. The framework generalizes many existing mobile user resource sharing models, and it can offer more flexibilities in terms of device cooperation and resource scheduling. Under the general framework, we focus on a non-convex energy consumption minimization problem, and propose a linear programming heuristic resource allocation algorithm, which can produce an output that is empirically close to the optimal solution.

In the second part on the multimedia platform operation, we study how the platform and users should behave on these platforms to maximize their payoffs. We focus on the emerging live streaming platforms, where streamers broadcast live streams for viewers. These platforms implement distinctive donation-based markets: streamers live stream free of charge, and viewers can voluntarily donate money to the streamers. The donations are split between the streamers and the platform with a fixed pre-agreed fraction. Under the donation-based markets, we study the platform's decision on the fixed split fraction design and streamers' decisions on their participations and service attribute selections (considering the preferences of streamers and viewers). To gain real-world insights, we further perform a case study based on the dataset collected from Twitch platform, and demonstrate how to compute the platform's optimal fraction without knowing the service attribute preferences of the streamers and viewers.

論文題目:

多媒體服務的優化與經濟學

作者: 唐茗

學校: 香港中文大學

學系: 信息工程學系

修讀學位: 哲學博士

隨著通信與信息技術的發展，基於網絡的多媒體服務成為社會生活中極其重要的一部分。為更好地了解并提供該類服務，本論文從兩個方面研究了網絡多媒體，包括多媒體服務的提供以及多媒體平台的運營。

本論文的第一部分研究了多媒體服務的提供，旨在研究如何為用戶提供高質量的多媒體服務。這部分的研究專注于移動網絡多媒體。由於移動設備具有異質性並且其資源往往有限，針對移動網絡的多媒體研究更加困難。為了提升服務質量，我們提出了衆包資源共享模型—使移動用戶通過設備對設備的連接，建立資源共享合作組，共同為組內用戶的多媒體服務調度資源。我們首先在視頻流的場景下，研究了通信資源的共享。 我們提出了視頻流資源衆包模型，允許移動用戶之間共享通信資源用以共同完成視頻調度。分析該模型十分困難，其主要原因是用戶的異步視頻下載行為以及用戶私人信息的隱藏行為。 克服這些困難，我們設計了在線資源調度算法，其調度結果可以趨近於系統的理論最優結果。同時，我們設計了基於拍賣理論的激勵機制，鼓勵用戶之間的資源共享。該機制實現了用戶私人信息的真實揭示，同時實現了高效的資源分配。我們進一步在通用的多媒體場景下，研究了關於通信，計算，及存儲資源的聯合共享模型。該模

型概括了許多當前已有的移動資源共享模型，並且為資源的共享與調度提供了更大的靈活性。基於該模型，我們分析了一個非凸的能量損耗最小化問題，并提出了與系統最優解相近線性資源調度算法。

本論文的第二部分研究了多媒體平台的運營，旨在研究在多媒體平台中，平台本身及用戶的收益優化行為。這部分的研究主要基於當前新興的直播平台。該類平台應用了獨特的基於捐款的市場模型：主播免費提供直播服務，觀看者自願為主播支付費用。這些自願的捐款會按照既定的比例分成給平台及主播。在該基於捐款的市場中，我們研究了平台對分成比例的最優決策以及主播對參與及直播視頻屬性的最優決策。我們進一步對從Twitch平台收集到的數據進行案例分析，并討論了如何在未知主播及觀看者的視頻屬性喜好的情況下，從平台的角度最優其分成比例的決策。

# Acknowledgements

I would like to express my heartfelt thanks to my supervisor Professor Jianwei Huang for his patient guidance and strong support. I always feel grateful and fortunate to have him as my supervisor. His influence on me does not limit to the skills and serious attitudes on doing research, but also his great attitudes toward life. I'm so grateful that Professor Huang always stands with me and support me, especially at the times I face difficulties.

I would like to thank my collaborators for their kindness and helps. Thanks to Professor Lin Gao for his support for my study and for his patient guidance on my research and technical writing. He provided significant helps on the two research projects we collaborated. Thanks Professor Lifeng Sun, Haitian Pang, and Shou Wang at Tsinghua University for our enjoyable collaborations. They gave me good advices and helps throughout our collaboration on my first research project. Thanks for Professor Ramesh Johari for his kind host at Stanford University. During my visit, he offered innovative ideas and encouraged me to explore deeper.

I would like to thank my present and past NCEL group members. Thanks to Dr. Lin Gao and Dr. Man Hon Cheung for their significant helps on my research. Thanks to Dr. Yuan Luo, Dr. Haoran Yu, Dr. Qian Ma for their suggestions and significant helps throughout my four-year study. Thanks to Dr. Hao Wang, Dr. Changkun Jiang, Dr. Yunlin Yu, and Meng Zhang for their strong supports at all of my difficult moments. Thanks to

Dr. Yanru Zhang, Dongwei Zhao, Guocheng Liao, Zhiyuan Wang, Qi Shao, Chao Huang, Yunshu Liu, and Ningning Ding for the company at NCEL. I also want to thank Haoyuan Pan, Zhaorui Wang, and Qing Yang for our time spent together at CUHK. Thanks Yidi Sun and Ruoyi Qiu for accompanying and supporting me at Stanford University.

I devote the most special gratitude to my parents, sister, and husband for their unconditional love and support over the years. I feel so grateful that they are always there with me throughout all the difficult times.

*To my parents, sister, and husband.*

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

With the fast development of communication and information technologies, humans have been increasingly enjoying multimedia services over the Internet. As mentioned in Internet Trends Report 2018 [67], an US adult spent 5.9 hours daily on average on multimedia services over the Internet in 2017, which includes 3.3 hours services over mobile networks. These values were 3.2 hours and 0.4 hours in 2010, respectively.

Such a significant demand makes it important to study the multimedia services thoroughly. There are two key aspects of the multimedia services.

The first aspect is *multimedia service provision*, i.e., how to provide high quality-of-experience (QoE) multimedia services to users. This is extremely important and challenging for mobile networks. This is because the mobile network connections are always unstable and heterogeneous among users, and the mobile devices are always limited in their resources. Hence, special designs are needed for providing high QoE mobile multimedia services.

The second aspect is *multimedia platform operation*, i.e., how the platform and users should behave on a multimedia platform to maximize their payoffs, considering the specific features of the platform. For example, on a live streaming platform, individual users (instead of companies) act as streamers, broadcasting live streams for viewers. On this platform, the traditional pricing market may not be suitable due to the large number of heterogeneous and unofficial individual streamers. This makes the live streaming platform implement a new type of market — donation-based market, where the streamers stream free of charge, and the viewers voluntarily donate to the streamers. Hence, when analyzing such a live streaming platform, the platform operation analysis should be attentive to the corresponding donation-based feature.

In this thesis, we focus on these two aspects and study the optimizations and economics of multimedia services. Regarding the *multimedia service provision*, we focus on the service provisions over mobile networks. We propose frameworks that enable mobile users to share their resources to enhance their mobile multimedia services. Based on the frameworks, we study optimization problems to properly schedule resource allocation, and design incentive mechanisms to motivate user sharing. Regarding the *multimedia platform operation*, we focus on the emerging live streaming platform, which implement the donation-based market, and analyze the platform and user behaviors. Our study on the service provision aspect significantly enhances the mobile multimedia service performance from both the technical and economic perspectives, and our study on the platform operation aspect provides helpful insights on the marketing of the emerging live streaming platform.

In Sections 1.1 and 1.2, we introduce the background of the *multimedia service provision* and the *multimedia platform operation*, respectively. In Section 1.3, we outline the main results and contributions of this thesis.

## 1.1 Multimedia Service Provision

For *multimedia service provision*, we aim to understand how to provide high QoE multimedia services to users. We focus on the multimedia service over mobile networks, whose demand increases dramatically (from 2010 to 2017, the average daily time that an US adult spent on mobile multimedia increases from 0.4 hours to 3.3 hours [67]). Providing high QoE multimedia services over mobile networks is challenging due to two main reasons.

The first reason is due to the mismatch between the large resource requirements of multimedia services and the limited resource supplies of mobile devices and networks. Specifically, the multimedia services, such as video streaming and live streaming, always request a large amount of *communication resources* (e.g., stream downloading and uploading), *computation resources* (e.g., stream decoding and encoding), and *caching resources* (e.g., stream storage), named "3C resources". On the other hand, comparing with wired devices, mobile devices always have limited 3C resources.

The second reason is due to the heterogeneities of the mobile users' multimedia service requirements and their 3C resources. Specifically, different mobile users can have very different service requirements (e.g., high quality or low quality videos depending on the device capabilities) and 3C resources (e.g., network resources of 3G or 4G cellular links), leading to challenges for effective QoE provision. This also leads to the potential mismatch of service requirement and resource supply at a single user level.

Addressing these two challenges, we propose *3C resource crowdsourcing frameworks* that enable mobile users (with or without multimedia service requirements) to form cooperation groups through device-to-device (D2D) connections (e.g., Bluetooth and WiFi Direct), in order to share their 3C resources for mobile multimedia services. Such frameworks can effectively pool the mobile users' heterogeneous resources, and improve the overall performance through efficient 3C resource allocation.

In this thesis, we first consider a video streaming application scenario, studying the communication resource sharing. The study includes resource allocation optimization and incentive mechanism design. Then, we consider a general multimedia scenario, studying the joint 3C resource sharing.

### 1.1.1 Communication Sharing: Optimization

To enhance the users' QoE in mobile video streaming services, researchers and industries has implemented *Adaptive BitRate* (ABR) streaming [14], which enables video players to adapt the video bitrate to real-time network conditions.

Most of the existing works focused on single-user streaming models in ABR [51, 65], under which the quality of the user's received service is always bounded by his own network resource. To better utilize the network resources, some recent works studied multi-user streaming models. For example, papers [60, 64, 28] studied D2D models, where users share their downloaded video segments with other users through D2D links. Papers [92, 62, 13] studied peer-to-peer (P2P) models, where users download video segments from other users who have already downloaded it through the Internet. Papers [99, 81, 98] studied bandwidth aggregation (BA) models, where multiple users aggregate their bandwidth to serve one user's video streaming need.

To further exploiting the heterogeneities among users, we propose a crowdsourced mobile video streaming (CMS) model based on the ABR streaming. This model enables nearby mobile users to form cooperative groups and share their network resources for more efficient video streaming. Different from the video content sharing in the D2D and P2P streaming models, users in the CMS model share their network resources so that different users can watch different videos. Different from aggregating multiple users' bandwidth for one user's streaming in the BA models, the CMS model aggregates multiple users' bandwidth for all users' video streaming needs, enhancing the users' QoE through proper network resource allocations.

Under this CMS model, a most important question is how the video streaming operation is scheduled among multiple users, including bitrate adaptation and network resource allocation. However, answering this question is challenging due to the asynchronous downloading operation and heterogeneous service requirements among mobile users.

In this thesis, we analyze the theoretical performance bound of the CMS model, and design an online scheduling algorithm that converges to the theoretical performance bound asymptotically.

### 1.1.2 Communication Sharing: Incentive Mechanism

Despite the advantages of implementing the CMS model, in practice, selfish users may not be willing to help others unless they receive proper incentives (e.g., increased online reputation or virtual currency).

Designing an effective incentive mechanism for the CMS model is very challenging. First, the video scheduling in ABR is segment based instead of time-slotted based, so it is challenging to schedule the downloading cooperation among the users who request and download videos at different times. Second, a user's valuation for a segment at a particular bitrate is his private information and can vary over time. The diverse and varying private valuation introduces difficulties in evaluating users' contributions for the CMS model and determining the proper incentive levels.

In this thesis, we propose auction-based incentive mechanisms for the CMS model. Overcoming the challenges, the mechanisms achieve the truthful users' private information revelation and the efficient network resource allocation.

### 1.1.3 Communication, Computation, and Caching Sharing

Mobile multimedia service provision involves the deployment of all the 3C resources: communication resources for downloading and uploading; computation resources for decoding, encoding, and analysis; caching resources for storage. Similar as the study on the video streaming scenario, to promote the QoE of the mobile multimedia services, a promising way is to enable mobile users to share their 3C resources through D2D connections, in order to efficiently utilize the mobile users' resources to satisfy their intensive and heterogeneous service requirements.

There have been several existing works that focus on the sharing of one type of the 3C resources, named 1C sharing models. For example, the user-provided networking in [54] and [85] focus on the sharing of communication resource, the ad hoc computation offloading in [34] and [32] focus on the sharing of computation resource, and the ad hoc content sharing in [57] and [33] focus on the sharing of caching resource. Some other recent works further considered the sharing of two types of the 3C resources, named 2C sharing model. Typical examples of 2C models include the distributed data analysis in [84], [38], which focus on the sharing of computation and caching resources.

Despite the success of the earlier 1C/2C resource sharing models, there are still significant potential benefits of exploiting the joint 3C resource sharing framework. Such a 3C sharing framework can further improve the resource utilization efficiency, by offering more flexibilities in terms of device cooperation and resource scheduling.

In this thesis, we present the first study regarding the general 3C resource sharing framework, which can generalize many of existing mobile resource sharing (1C/2C) models. A key feature of this new 3C sharing framework is "resource-centric" — any of the services (e.g., downloading, analysis, or caching) is modeled by the resources that it requests (instead of the functionalities that it achieves), so that various types of services requesting any combination of the 3C resources can coexist in the same framework. Because of the "resource-centric" feature, the 3C sharing framework further provides additional network design and optimization flexibilities.

## 1.2 Multimedia Platform Operation

For *multimedia platform operation*, we aim to understand how the platform and users should behave on a multimedia platform to maximize their payoffs. We focus on the live streaming platform, which is quite different from traditional multimedia platforms, because of its donation-based feature.

### 1.2.1 Live Streaming Platform

Live streaming platform is an emerging type of multimedia platform, where individual users (instead of companies) act as streamers, broadcasting streaming to viewers. Twitch (https://www.twitch.tv/) is an important example, who attracts more than 15 million unique daily visitors and 2 million unique monthly streamers in 2017 [12].

One important feature of the live streaming platform is its donation-based market. Specifically, the streamers provide streaming services without mandatory charges, and the viewers enjoy the services and voluntarily donate to the streamers (due to their desires of being acknowledged on the platform and supporting the streamers for future service provisions [77]). The donations are split between the streamers and the live streaming platform with a fixed *donation-split-fraction* (DSF), which corresponds to the fraction of donations kept by the streamers. In real-world, the total donation volume on live streaming platforms is huge. In 2017, a total of $101 million dollars of donations were received by top live streaming platforms including Twitch, YouTube Live, Mixer, Facebook Live, and Periscope [7].

The donation-based feature of the platforms bring two unique questions as follows. First, from the streamers' point of view, *how should they decide their* *service attributes (e.g., in a live streaming platform, what game to broadcast at what time) given a fixed DSF?* The streamers and viewers may have different preferences over the service attributes, and streamers' choices (which can be different from their own preferences) will affect the competition levels among streamers and the satisfactions of the viewers. Second, *from the platform's point of view, how should it set the DSF to maximize its payoff?* A higher DSF leads to a smaller per-donation revenue to the platform. On the other hand, it can increase the incentive for the streamers to participate in the platform and better match the viewers' preferences to induce more donations.

Despite the fact that the live streaming platforms have been embraced by top companies (e.g., YouTube and Facebook) and attracts millions of streamers and viewers, there does not exist a good understanding regarding the answers of the above two key questions.

In this thesis, motivated by the sequential decision process in real platforms, we formulate the problem using a two-stage game to understand the above two key questions: in Stage I, the platform first announces the DSF, the fraction of donations kept by the streamers; in Stage II, each streamer decides whether to participate and what service attribute to choose (for example, at what time to stream). The results suggest that from the streamers' point of view, a larger DSF leads to more streamer participations and a better match to the viewers' preferences; from the platform's point of view, as the streamers' costs increase, the platform may not always choose to share less donations with the streamers (hence provide less incentives to streamers).

## 1.3 Outline and Contributions

This thesis is organized into two main parts: in Chapters 2, 3 and 4, we study the multimedia service provision; in Chapter 5, we study the multimedia platform operation. In Chapter 6, we conclude the thesis, and discuss future research directions. We outline our contributions in each chapter as follows.

In Chapter 2, we focus on the communication resource sharing in a mobile video streaming scenario, where we propose a crowdsourced mobile video streaming (CMS) model and study its resource allocation optimization problem. First, to our best knowledge, this is the first work that proposes a general multi-user cooperative streaming framework for mobile video streaming. The framework enables mobile video users to crowdsource their mobile network connections for cooperative video streaming, and can effectively improve the QoE of video users. Second, we analyze the theoretical performance bound of the proposed cooperative streaming system, overcoming the challenging issue of asynchronous operations by using a virtual synchronous system. Such a performance bound analysis is fundamental for the design, evaluation, and implementation of practical algorithms in such a cooperative streaming system. Third, we implement the cooperative streaming system in the practical scenario without future and global network information, and design a Lyapunov-based online streaming algorithm. The proposed algorithm converges to the theoretical performance bound asymptotically.

In Chapter 3, we design incentive mechanisms for the CMS model. First, we propose multi-dimensional auction based incentive mechanisms for the CMS model, supporting the asynchronous downloading and bitrate adapting of video users. Specifically, we propose mechanisms for single and multiple video segments allocations, both of which achieve truthful private information revelation and efficient network resource allocation. To the best of our knowledge, the mechanism for multiple segments is the first mechanism achieving both truthfulness and efficiency in a multi-object multidimensional auction, overcoming the known challenge of the preferential dependent bidding dimensions (including video segment quality, quantity, and bidders' willingness-to-pay) [23]. Second, to enhance the long-term social welfare of video streaming services, we further improve the proposed mechanisms by allowing bidders (in the auction mechanisms) to refrain from bidding according to their current situations. The simulation results show that such a modification can successfully decrease rebuffer and bitrate degradation frequency along the entire video streaming. Third, we construct a demo system that enables the cooperation among multiple users watching multiple videos. Using the demo, we further analyze the real-world performances of the CMS model.

In Chapter 4, we study the joint 3C resource sharing for general mobile multimedia scenarios. First, we propose a general 3C sharing framework and the corresponding "resource-centric" mathematical formulation. This framework generalizes many existing 1C/2C resource sharing models, and improves the resource utilization efficiency by encouraging more devices participating and more flexible resource scheduling. Second, we focus on the energy consumption of mobile devices under the 3C framework, and formulate and solve an energy consumption minimization problem. The problem is difficult as it is an integer non-convex optimization problem. We first transform it to an integer linear programming problem, and then proposed a linear programming heuristic algorithm, which can produce an output that is empirically close to the optimal solution. Third, we analyze the energy consumption reduction due to the 3C resource sharing analytically. We show that if the 3C framework can double the number of cooperative devices (comparing with 1C models), it can reduce the energy by a maximum of about 20% of the energy consumed in noncooperation case (where devices do not cooperate).

In Chapter 5, we study the live streaming platform, with particular emphasis on its donation-based market. First, to the best of our knowledge, this is the first work that presents a two-stage model of the donation-based market, and study how the platform should set a donation-split fraction (DSF), which is the fraction of donations kept by the streamers, in Stage I, and how the streamers should decide their participations and service attributes in Stage II. Second, for the Stage II streamer decision problem, we prove that it is a potential game and derive the asymmetric equilibria, which is quite challenging [61]. We show that a larger DSF leads to more streamer participations and a better match to the viewers' preferences at the equilibrium. Third, for the Stage I platform decision non-convex optimization problem, we derive the lower-bound of the optimal solution, which reflects how the optimal DSF changes with system parameters. We further analyze a special case with two attribute values, and show that as the streamers' costs increase, the platform may not always choose to share less donations with the streamers (hence provide less incentives to streamers) to maximize the platform's revenue. Forth, we collect two weeks' data about streamers' and viewers' behaviors from the Twitch platform. Based on the data, we demonstrate how to compute the platform's optimal DSF without knowing the preferences of the streamers and viewers, and provide suggestions on how Twitch should set its DSF.

# Part I

# Multimedia Service Provision

# Chapter 2

# Communication Sharing: Optimization

## 2.1 Introduction

### 2.1.1 Background

Mobile video streaming is growing at an unprecedented rate today. In 2015, mobile video traffic accounted for around 55% of global mobile traffic, and is expected to grow at an annual rate of 62% between 2015 and 2020 [10]. *Adaptive BitRate (ABR)* streaming [51] is a widely-used technology for video streaming over large distributed HTTP networks such as Internet. The key idea is to enable video players to *adapt* the video bitrate (which corresponds to the video quality such as resolution) to the real-time network conditions to achieve the desirable quality of experience (QoE). For example, a video user can switch to a high bitrate/quality to enjoy a high resolution video when the channel condition is good, while to a low bitrate/quality to avoid a too long waiting time when the channel condition is poor. Nowadays, the ABR streaming technology has been adopted by many popular online video streaming systems, such as HTTP Dynamic Streaming of Adobe Systems [2], HTTP Live Streaming of Apple Inc. [72], and Smooth Streaming of Microsoft [6]. One important standard of ABR streaming is Dynamic Adaptive Video Streaming over HTTP (DASH) [83], also known as MPEG-DASH, which has been supported by several technical specifications (e.g., 3GPP [89] and OIPF [9]) and applications (e.g., Netflix and YouTube) for streaming over both wired and wireless networks.

To achieve the flexible bitrate adaptation in ABR streaming, a source video is first partitioned into a sequence of short multi-second *segments*, and each segment is encoded at multiple pre-defined bitrates. Then, the bitrate adaptation of each video user can be achieved by choosing different bitrates for different segments. Clearly, with proper bitrate adaptations, video users can achieve the desirable tradeoff between the QoE and the streaming cost (e.g., resource consumption). While most of the existing work focused on the bitrate adaptation methods of a *single user* [94, 51, 65, 46], in this chapter we consider a more general *multi-user* video streaming over wireless cellular networks.

In a multi-user wireless scenario, the QoE of each user is affected not only by the stochastically changing of his own network condition (such as wireless channel fading), but also by the potential resource competition and interference of other users [58]. Without proper coordination or cooperation among users, such competition and interference may greatly degrade the network condition (e.g., leading to network congestion), hence increase the streaming cost (due to, for example, the increased transmission power or repeated data retransmissions) and harm the QoE of users. Therefore, single-user based bitrate adaptation methods in the existing literature often fail to provide the desirable QoE for video users in the multi-user scenario, due to the lack of considerations of the potential cooperation among video users.

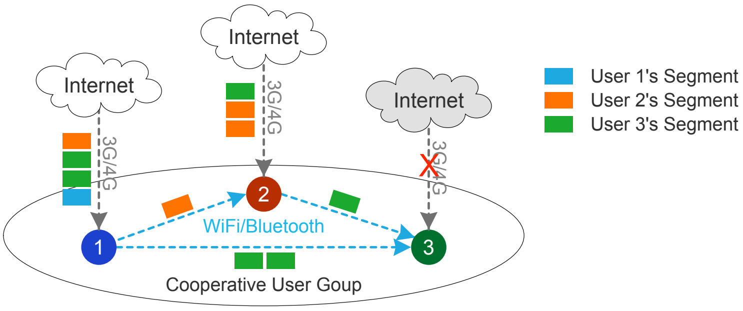


Figure 2.1: Crowdsourced video streaming model.

### 2.1.2 Motivations

In this chapter, we propose a novel user cooperation framework, called *crowdsourced (video) streaming*, for multi-user video streaming over wireless cellular networks, based on the crowdsourced *user-provided networking* (UPN) technology [53, 52]. The key idea is to enable nearby mobile users to form a cooperative group (via WiFi or Bluetooth) and crowdsource their cellular radio connections and resources for cooperative video streaming. Namely, in a cooperative group, each user can download video data for other users using his cellular link and resources and download his own video data through other users' links and resources. For example, those users with lower cost cellular links (e.g., those consuming less resources in data downloading) can help other users with higher cost cellular links or without available cellular links to download their video data. Figure 2.1 illustrates the crowdsourced streaming model with a cooperative user group $\{1, 2, 3\}$, where user 1 downloads one segment for user 2 and two segments for user 3 (who has no cellular link), and user 2 downloads one segment for user 3.

Motivations of using such a crowdsourced UPN framework for multi-user cooperative streaming are as follows. First, mobile users can be highly heterogeneous in terms of their cellular link capabilities, as illustrated by the measured video downloading throughputs (of mobile users) reported from

Netflix and Youtube. Hence, crowdsourcing (aggregating) the capacity of nearby mobile users can effectively reduce the impact of wireless network variation. Second, by exploiting the user diversity of resource availability and service requirement, this crowdsourced UPN can reduce the negative externality (e.g., competition and interference among users), while amplifying the positive network effect (e.g., cooperation among users).[1] Moreover, such a crowdsourced UPN can be easily implemented in practice by installing some customized mobile apps on smartphones (e.g., OpenGarden), and the related optimization and incentive issues have been studied in the recent literature (e.g., [53, 52]). However, the existing optimization techniques in [53, 52] cannot be directly applied to the video streaming model, mainly because of the asynchronous operation of video streaming and the unique QoE requirements of video applications.

### 2.1.3 Solutions and Contributions

Specifically, we propose a crowdsourced streaming framework for multi-user cooperative video streaming over wireless cellular networks, which enables nearby users to crowdsource their cellular links and resources for cooperative video streaming. We focus on studying the users' cooperative streaming operations (including *download scheduling* and *bitrate adaptation*) in the proposed crowdsourced streaming system. Namely, for each user, when and for whom he is going to download the video segments at which bitrates?

First, *we formally define the users' operations in the crowdsourced streaming system, and formulate the corresponding social welfare optimization problem (Section 2.4).* The optimal solution of this problem provides the theoretical social welfare performance bound of the proposed crowdsourced streaming

[1] As a result, UPN has some successful commercial applications, e.g., FON (www.fon.com) and Karma (yourkarma.com).

system. Here, social welfare is defined as he difference between the user utility (from video service) and the energy cost (for downloading video). In particular, the user utility characterizes the QoE of users, and depends on factors such as video quality, quality fluctuation, and rebuffering. The energy cost mainly consists of the energy consumptions for downloading data via cellular links (and Internet) and exchanging data via local WiFi links.

Second, *we analyze the social welfare performance bound of the proposed crowdsourced streaming system (Section 2.5).* Directly solving such a performance is challenging, due to the asynchronous operations of users as well as the mixed-integer nature of the problem. To this end, we introduce a *virtual* time-slotted system with the synchronized operations, and formulate the new social welfare optimization problem as a linear programming (which can be solved efficiently with many standard methods). We show that with proper choices of time parameters, the optimal social welfare performance of the virtual time-slotted system provides effective upper-bound and lower-bound for the optimal performance (bound) of the original asynchronous system.

Finally, *we design a Lyapunov-based online streaming algorithm for the practical implementation of the proposed crowdsourced streaming system (Section 2.6).* The proposed algorithm converges to the theoretical performance bound asymptotically, with a controllable approximation error bound. Moreover, it relies only on the current state and historical streaming information (while not on any future network information), hence can be implemented in the *online* manner; and it requires only the local information exchange within each cooperative group (while not the global network information exchange), hence can be implemented in the *distributed* manner. We perform extensive experimental simulations with real data traces to evaluate its performance gap with the theoretical bound and to compare its performance with state-of-art online algorithms in the existing literature.

For clarity, we summarize the logical relationship among the above three parts as follows: *(i) the social welfare optimization problem in Section 2.4 defines the theoretical performance bound of the proposed crowdsourced streaming system (but it is challenging to solve);(ii) the virtual time-slotted system in Section 2.5 helps characterize the region (i.e., upperbound and lowerbound) of the above theoretical performance bound; (iii) the online algorithm in Section 2.6 converges to the above theoretical performance bound asymptotically in the scenario without complete future network information.* More precisely, the key contributions of this chapter are summarized as follows.

- *Novel Model:* To our best knowledge, this is the first work that proposes a crowdsourced streaming framework for multi-user cooperative video streaming. The framework enables mobile video users to crowdsource their radio connections and resources for cooperative video streaming, and can effectively improve the QoE of video users. Moreover, we provides both theoretical performance analysis and practical algorithm design for such a crowdsourced streaming system.

- *Performance Bound Analysis:* We analyze the theoretical performance bound of the proposed crowdsourced system, overcoming the challenging issue of asynchronous operations by introducing a virtual time-slotted system. Such a performance bound analysis is fundamental for the design, evaluation, and implementation of practical algorithms in such a crowdsourced streaming system.

- *Online Algorithm Design:* We design a Lyapunov-based online streaming algorithm for the implementation of the proposed crowdsourced streaming system in the practical scenario without future and global network information. The proposed algorithm converges to the theoretical performance bound asymptotically.

- *Experiments and Performances:* Experiments with real data traces show that our proposed online algorithm outperforms the existing algorithms in terms of both achieved bitrate (with an average gain of $20\% \sim 30\%$) and social welfare (with an average gain of $10\% \sim 50\%$).

We would like to clarify that the focus of this study is system performance optimization instead of incentive mechanism design. Namely, we suppose that some well-designed incentive mechanisms have been adopted, such that all users are willing to participate in the crowdsourced streaming system. In the complete information scenario, this can be achieved through a Nash bargaining [69] between the downloader and the receiver (for each segment downloading), with which each of them can achieve a payoff no worse than that in the non-cooperative system (hence is willing to join the crowdsourced system). In the incomplete information scenario, it is necessary to adopt an incentive compatible mechanism (e.g., auction) to elicit the private information of users first, and then distribute the generated social welfare properly among the receiver and the downloader. We leave the detailed analysis of incentive mechanism design in Chapter 3.

The rest of the chapter is organized as follows. In Section 2.2, we review the related works. In Section 2.3, we present the system model. In Section 2.4, we provide the problem formulation. In Section 2.5, we propose the virtual time-slotted system and the performance bound analysis. In Section 2.6, we propose the Lyapunov-based online algorithm. We provide simulation results in Section 2.7 and conclude in Section 2.8.

## 2.2 Literature Review

Prior works on ABR streaming mainly focused on the bitrate adaptation of a *single user* (see [94] for a comprehensive discussion), including the buffer-based method [51], the channel prediction-based method [65], and the hybrid buffer- and prediction-based method [46]. Specifically, in [51], Huang *et al.* adapted the bitrate by a mapping from the current buffer level to reference bitrate. In [65], Li *et al.* choosed the segment bitrate based on the predicted channel capacity. Among the hybrid scheduling methods, Hao *et al.* in [46] proposed a geo-predictive streaming system for predicting the available network bandwidth in different locations and constructing the bandwidth map, which is used for the bitrate adaptation with the buffer size.

Some recent works extended the basic single-user model to more advanced ones, such as multi-link model [91], multi-server model [88], and P2P model [62, 75]. Specifically, in [91], Xing *et al.* considered the scenario where a user connects to server via multiple links (e.g. 3G and WiFi) to reduce the user energy cost and increase the user utility. However, the above work did not consider the cooperation of multiple users. In [88], Tian *et al.* considered the multi-server model, where users download video from multiple servers to reduce the server load. In [62, 75], researchers constructed adaptive streaming models on P2P systems to reduce the server load, and studied related technical and economic issues for promoting the user cooperation in the P2P streaming. There are several key differences between our proposed crowdsourced streaming model and the above multi-server and P2P streaming models. In the multi-server or P2P streaming model, each video segment has multiple copies resided on multiple servers or multiple user devices (peers), and video users can download a video segment either from a server or from a user peer, via his own wireless cellular link. Hence, the key design purpose of such multi-server or P2P model is *to reduce the load of the video server.* In our crowdsourced streaming model, however, each video segment has a unique copy residing on the video server, and users can download a video segment (from the video server) either via his own wireless cellular link or a neighbor's cellular link. Hence, the key design purpose of such a crowdsource streaming model is *to reduce the uncertainty or improve the efficiency of user's wireless cellular link*.

Moreover, there is also a growing interest in exploiting the multi-user cooperative video streaming scenarios [81, 98, 74, 44, 41]. Specifically, in [81] and [98], researchers considered the user cooperation in mobile video streaming, and proposed models to aggregate the bandwidth of a mobile video user and its neighbors. However, these models focused on the simple one-to-many cooperation between a single video user and multiple helpers. We consider a more general many-to-many cooperation framework with multiple video users, where each user acts as both the video player and the helper. In [74], Pu *et al.* proposed a rate adaptation algorithm for optimizing the adaptive streaming across multiple mobile users, but they did not consider the individual characteristics of different users. In [44], Georgopoulos *et al.* studied multiuser streaming in a home networking scenario, but they did not consider the important network characteristics such as the transmission rates. In [41], El Essaili *et al.* studied the mobile network potential for enhancing the user QoE in multiuser DASH over LTE cellular networks, and proposed the joint optimization of transmission and bitrate of the mobile DASH users considering their buffer levels. However, they neither considered the performance bound, nor the online algorithm. In this chapter, we study multi-user cooperative video streaming using the crowdsourced UPN technology, and provide both the theoretical performance bound analysis and practical online algorithm design.

## 2.3 System Model

### 2.3.1 Network Model

We consider a set $\mathcal{N} \triangleq \{1, \ldots, N\}$ of mobile video users in wireless cellular networks, who want to watch videos (on their smartphones) via 3G/4G cellular links. Mobile users are heterogeneous in terms of their cellular link capacities and video quality requirements. For example, a user requesting a high quality video may suffer from a low cellular link capacity, due to factors such as a severe channel fading and a high cellular network congestion. This may reduce the quality of the video and increase the video quality variation, both harming the user's quality of experience (QoE). On the other hand, a user requesting a low quality video (or not playing a video at all) may experience a high cellular link capacity, and have extra capacity to help other users. Thus, it is desirable to enable users to connect with each other to download the streaming video contents cooperatively.

*1)* ***Crowdsourced User-Provided Network:*** We consider a user cooperation scheme based on the *crowdsourced* user-provided network (UPN) technology, where nearby users form a cooperative group (via WiFi) and crowdsource their radio connections and resources for cooperative video streaming. Namely, in a cooperative group, each user can download video data for other users using his cellular link and resources and download his own video data through other users' links and resources. We refer to such a multi-user cooperative video streaming scheme as *crowdsourced (video) streaming.* Figure 2.1 illustrates such a crowdsourced streaming model with a cooperative user group {1, 2, 3}, where user 1 downloads one segment for user 2 and two segments for user 3, user 2 downloads one segment for user 3, and user 3 does not download any video content due to the temporary interruption of his cellular link.

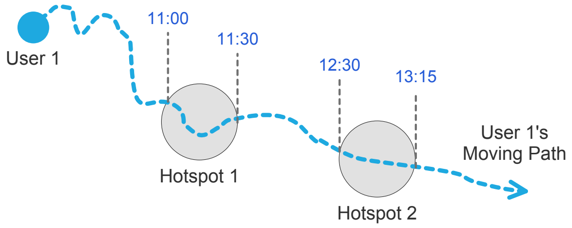


Figure 2.2: Hotspot-based mobility model.

We assume that some well-designed incentive mechanisms (e.g., Nash bargaining or auction) have been adopted, such that video users are willing to participate in the crowdsourced system to help others. Note that two users in the crowdsourced system can only help each other when they are close enough, such that they can connect with each other through WiFi.

*2)* ***Mobility Model:*** The cooperation gain of crowdsourced streaming highly depends on the number of cooperative users and the duration of the cooperative user group, both closely related to the users' mobility patterns. In this chapter, we adopt a *hotspot-based* mobility model [27], where the whole area is divided into a set of small hotspots and the non-hotspot area,[2] and each user moves across a sequence of hotspots during his travel in the following pattern: *staying for a certain period of time in each hotspot that he passes, and taking some time for each transition (from one hotspot to another).* Figure 2.2 illustrates such a mobility model, where user 1 stays at hotspot 1 for 30 minutes (11:00∼11:30), and then takes 1 hour to move to hotspot 2 and stays at hotspot 2 for 45 minutes (12:30∼13:15).

In such a hotspot-based mobility model, users in the same hotspot at the same time can connect with each other (hence form a cooperative group), while users in different hotspots or in the non-hotspot area cannot. Such a

[2] A hotspot is a small area where users are likely to stay for a substantial amount of time (e.g., a bus stop or a coffee shop), hence can maintain their WiFi connections with each other for a reasonable amount of time. The non-hotspot area is the area where users are likely to move fast (e.g., an expressway), hence can hardly maintain long-time connections among users.

mobility model has been widely-used in the scenarios where users need to take certain time to interact with each other (e.g., mobile data forwarding in [95]).

***Notations:*** We consider the operation in a period of continuous time $\mathcal{T} \triangleq [0,\ T]$, where $t = 0$ is the initial time and $T$ is the ending time. Let $\mathcal{A} \triangleq \{1, \ldots, A\}$ denote the set of all hotspots, and $\{0\}$ denote the non-hotspot area. The key notations in this part are listed below.

- $a_n(t) \in \mathcal{A} \bigcup \{0\}$: the location of user $n$ at time $t$;
- $h_n(t) > 0$: the cellular link capacity of user $n$ at time $t$;
- $e_{n,m}(t) \in \{0, 1\}$: the indicator denoting whether users $n$ and $m$ are encountered (i.e., in the same hotspot) at time $t$, i.e., $e_{n,m}(t) = 1$ if $a_n(t) = a_m(t) \in \mathcal{A}$ and $e_{n,m}(t) = 0$ otherwise.

For convenience, we refer to the user location and cellular link capacity $\{(a_n(t), h_n(t)), \forall n \in \mathcal{N}, t \in \mathcal{T}\}$ as the *network information*, which varies randomly over time. Note that the encounter indicator $e_{n,m}(t)$ are derived from the location information of users $n$ and $m$.

### 2.3.2 Video Streaming Model

We consider a typical ABR streaming model [51], where a single source video file is partitioned into multiple segments and delivered to a video user using HTTP. The key features of ABR model are summarized below.

(i) *Video Segmenting:* To facilitate the video delivery over the Internet, a source video file is divided into a sequence of small HTTP-based file segments, each containing a short interval of playback time (e.g., 2–10 seconds) of the source video, which is possibly several hours in term of the total duration (e.g., a movie). A user downloads the video segment by segment.

(ii) *Multi-Bitrate Encoding:* Each segment is encoded at multiple bitrates, each corresponding to a specific video quality (such as resolution).[3] A user can select different bitrates for different segments according to real-time network conditions.

(iii) *Data Buffering:* For smoothly playing, each downloaded segment is first stored in a buffer at the user's device, and then fetched to the video player sequentially for playback. The maximum buffer size on user device is usually limited (e.g., 20–40 Seconds).

***Notations:*** Key notations in this part are listed below.

- $\beta_n > 0$: segment length (in seconds) of user $n$'s video;
- $\mathcal{R}_n \triangleq \{R_n^1, R_n^2, ..., R_n^Z\}$ (with $0 < R_n^1 < R_n^2 < ... < R_n^Z$): the set of bitrates (in Mbps) available for user $n$, which depends on both the sever-side protocols and the user-side parameters such as device type.
- $B_n > 0$: maximum buffer size (in seconds) of user $n$.

## 2.4 Problem Formulation

In this section, we characterize the users' cooperative streaming operations in the crowdsourced model, and formulate the associated optimization problem.

Specifically, with the ABR streaming, each source video is downloaded *segment by segment*. Namely, each user starts to download a new segment (with a specific bitrate) only when completing the existing segment downloading. Hence, users operate in an *asynchronous* manner, as they may complete segment downloading at different times. We refer to such an operation scheme as the *segmented download operation*.

[3] For example, a low quality video (e.g., with a resolution lower than $480 \times 360$) usually requires a bitrate lower than 400 Kbps, while a high quality video (e.g., with a resolution higher than $1280 \times 720$) usually requires a bitrate higher than 1 Mbps.

### 2.4.1 Downloading Sequence

With the segmented operation, each user $n$'s downloading operation can be characterized by a sequence:

$$\boldsymbol{S}_n \triangleq \left\{\boldsymbol{s}_{n[1]},\ \boldsymbol{s}_{n[2]},\ ...,\ \boldsymbol{s}_{n[k]},\ ...\right\}, \tag{2.1}$$

with each element $\boldsymbol{s}_{n[k]}$ denoting the information of the $k$-th downloaded segment, including the segment owner $u$, bitrate level $z$, bitrate $r = R_u^z$,[4] download start time $t^{\mathrm{s}}$, and end time $t^{\mathrm{e}}$. Namely, we can write $\boldsymbol{s}_{n[k]}$ as

$$\boldsymbol{s}_{n[k]} = (u,\ z,\ r,\ [t^{\mathrm{s}}, t^{\mathrm{e}}]). \tag{2.2}$$

To distinguish the information of different segments, we will also write the information of segment $\boldsymbol{s}_{n[k]}$ as $(u_{n[k]}, z_{n[k]}, r_{n[k]}, t^{\mathrm{s}}_{n[k]}, t^{\mathrm{e}}_{n[k]})$ whenever needed.

It is easy to see that our crowdsourced streaming model generalizes the model without crowdsourcing, in which case we can simply restrict each user $n$ downloading only his own segment, i.e., $u_{n[k]} = n, \forall n, k$.

Next we provide the constraints for a *feasible* downloading sequence $\boldsymbol{S}_n$ of user $n$.

(i) *Timing Constraint:* As users download segment by segment, we have the following timing constraint:

$$\text{C.1}: \quad t^{\mathrm{e}}_{n[k]} \leq t^{\mathrm{s}}_{n[k+1]}, \quad \forall k = 1, ..., |\boldsymbol{S}_n|; \tag{2.3}$$

A strict inequality implies that user $n$ waits for some time before starting to download the next segment $\boldsymbol{s}_{n[k+1]}$, for example, when all users' buffers are full.

(ii) *Capacity Constraint:* Each segment $\boldsymbol{s}_{n[k]} = (u, z, r,\ [t^{\mathrm{s}}, t^{\mathrm{e}}])$ consists of $r \cdot \beta_u$ Mbits of video data, and is downloaded by user $n$ within time interval

[4] Here the bitrate $r = R_u^z$ is redundant information, and mainly introduced for facilitating the later description.

$\left[t^{\mathrm{s}}_{n[k]}, t^{\mathrm{e}}_{n[k]}\right]$. Hence, we have the following cellular link capacity constraint:

$$\text{C.2}: \quad r \cdot \beta_u \leq \int_{t^{\mathrm{s}}_{n[k]}}^{t^{\mathrm{e}}_{n[k]}} h_n(t)\mathrm{d}t, \quad \forall k = 1, ..., |\boldsymbol{S}_n|, \tag{2.4}$$

where $h_n(t)$ is the real time cellular link capacity (in Mbps) of user $n$ at time $t$.

(iii) *Encounter Constraint:* Each user can only download data for a nearby encountered user. Hence, a segment with $\boldsymbol{s}_{n[k]} = (u, z, r, [t^{\mathrm{s}}, t^{\mathrm{e}}])$, $n \neq u$ is feasible only if users $n$ and $u$ are encountered during $\left[t^{\mathrm{s}}_{n[k]},\ t^{\mathrm{e}}_{n[k]}\right]$, i.e.,

$$\text{C.3}: \quad e_{n,u}(t) = 1,\ t \in \left[t^{\mathrm{s}}_{n[k]},\ t^{\mathrm{e}}_{n[k]}\right], \quad \forall k = 1, ..., |\boldsymbol{S}_n|. \tag{2.5}$$

### 2.4.2 Receiving Sequence

Given the feasible downloading sequences of all users, i.e., $\boldsymbol{S}_n, \forall n \in \mathcal{N}$, we can derive the segment receiving sequence of each user $m$ as follows:[5]

$$\widehat{\boldsymbol{S}}_m = \bigcup_{n \in \mathcal{N}, k \in \{1, ..., |\boldsymbol{S}_n|\}: u_{n[k]} = m} \left\{\boldsymbol{s}_{n[k]}\right\} \tag{2.6}$$

We assume that a proper download scheduling has been adopted, such that there is no repeated segments within $\widehat{\boldsymbol{S}}_m$, and all segments in $\widehat{\boldsymbol{S}}_m$ are sorted according to the playback order. We denote the $k$-th segment in the reordered $\widehat{\boldsymbol{S}}_m$ by $\hat{\boldsymbol{s}}_{m[k]}$. Then, we can write the receiving sequence of user $m$ as:

$$\widehat{\boldsymbol{S}}_m \triangleq \left\{\hat{\boldsymbol{s}}_{m[1]},\ \hat{\boldsymbol{s}}_{m[2]},\ ...,\ \hat{\boldsymbol{s}}_{m[k]},\ ...\right\}, \tag{2.7}$$

with each element $\hat{\boldsymbol{s}}_{m[k]} = \left(\hat{u},\ \hat{z},\ \hat{r},\ [\hat{t}^{\mathrm{s}}, \hat{t}^{\mathrm{e}}]\right)$ denoting the information of the $k$-th segment played by user $m$. Similarly, we will write the information of $\hat{\boldsymbol{s}}_{m[k]}$ as $(\hat{u}_{m[k]}, \hat{z}_{m[k]}, \hat{r}_{m[k]}, \hat{t}^{\mathrm{s}}_{m[k]}, \hat{t}^{\mathrm{e}}_{m[k]})$ whenever needed.

It is easy to see that $\hat{u}_{m[k]} = m$ for all $\hat{\boldsymbol{s}}_{m[k]} \in \widehat{\boldsymbol{S}}_m$. To facilitate the later analysis, we further assume that $\hat{t}^{\mathrm{e}}_{m[k]} \leq \hat{t}^{\mathrm{e}}_{m[k+1]}$, $\forall k = 1, ..., |\widehat{\boldsymbol{S}}_m|$, that is,

[5] We do not consider the WiFi transmission time here, as the WiFi link capacity (typically tens to hundreds Mbps) is usually much larger than a video bitrate (typically low than than two Mbps).

user $m$ receives the segments in $\widehat{\boldsymbol{S}}_m$ sequentially. Note that this can always be achieved by a proper schedule of downloading sequences with the full network information.[6]

As mentioned previously, each received segment is stored in a buffer at the user's device, and then is fetched to the video player sequentially for playback. Let $b_{m[k]}$ denote the buffer level (in seconds) of user $m$ *when receiving the k-th segment*, i.e., at the time $\hat{t}^{\mathrm{e}}_{m[k]}$. Then, we have the following ***buffer update rule*** for user $m$:

$$b_{m[k]} = \left[b_{m[k-1]} - \left(\hat{t}^{\mathrm{e}}_{m[k]} - \hat{t}^{\mathrm{e}}_{m[k-1]}\right)\right]^{+} + \beta_m, \tag{2.8}$$

where $[x]^+ = \max\{0, x\}$. Here $\hat{t}^{\mathrm{e}}_{m[k]} - \hat{t}^{\mathrm{e}}_{m[k-1]}$ is the time interval between receiving of $\hat{\boldsymbol{s}}_m[k-1]$ and $\hat{\boldsymbol{s}}_{m[k]}$, during which a period $\hat{t}^{\mathrm{e}}_{m[k]} - \hat{t}^{\mathrm{e}}_{m[k-1]}$ of video is played back and removed from the buffer; $\beta_m$ is the segment length (playback time) of the newly received segment $\hat{\boldsymbol{s}}_{m[k]}$.

Since each user $m$'s buffer size is limited with $B_m$ (seconds), we have the following *buffer constraint*:

$$\text{C.4}: \quad 0 \le b_{m[k]} \le B_m, \quad \forall k = 1, ..., |\widehat{\boldsymbol{S}}_m|.$$

### 2.4.3 User Payoff

The payoff of a user mainly consists of two parts: a *utility* function capturing the user's QoE for video service, and a *cost* function capturing the user's energy consumption for both video downloading and playing.

*1)* ***Quality-of-Experience (QoE):*** Users often desire for a higher video quality without frequent quality changes and freezes during playback. Hence, a user's QoE mainly depends on the video quality, quality fluctuation, and rebuffering. Note that bitrate is a good measurement of video quality, and

[6] For example, if $\hat{t}^{\mathrm{e}}_{m[k]} > \hat{t}^{\mathrm{e}}_{m[k+1]}$, i.e., the $k+1$-th segment is received before the $k$-th segment, we can simply change their downloading orders.

in general there is a distinct and monotonic relationship between bitrate and quality. Hence, we will define the QoE on bitrate for notational convenience.

(i) *Video Quality:* A higher video quality (bitrate) brings a higher value for users. Let $v_n(r)$ denote the value that user $n$ achieves from bitrate $r$ during one unit of playback time.[7] Then, the total value that user $n$ achieves from all received segments $\widehat{\boldsymbol{S}}_n$ (each with a playback time of $\beta_{\hat{u}_{n[k]}} = \beta_n$) is:

$$V_n(\widehat{\boldsymbol{S}}_n) \triangleq \sum_{k=1}^{|\widehat{\boldsymbol{S}}_n|} v_n\left(\hat{r}_{n[k]}\right) \cdot \beta_n. \tag{2.9}$$

Obviously, $v_n(\cdot)$ is an increasing function (as video quality monotonically increases with bitrate). In our simulations, we adopt the following value function [58]: $g_n(r) = \log(1+\theta_n \cdot r)$, where $\theta_n > 0$ is a user-specific evaluation factor capturing user $n$'s desire for a high quality video service.

(ii) *Quality Fluctuation:* The change of quality (bitrate) during playback decreases the user QoE, especially when the quality is degraded. In this chapter, we assume that there is a value loss proportional to the bitrate decrease once the quality is degraded, while there is no value loss when the quality is upgraded [58]. Let $\phi_n^{\mathrm{QD}} > 0$ denote the value loss of user $n$ for one unit (in Mbps) of bitrate decrease. Then, the total value loss of user $n$ induced by quality degradation is[8]

$$L_n^{\mathrm{QD}}(\widehat{\boldsymbol{S}}_n) \triangleq \sum_{k=2}^{|\widehat{\boldsymbol{S}}_n|} \phi_n^{\mathrm{QD}} \cdot \left[\hat{r}_{n[k-1]} - \hat{r}_{n[k]}\right]^+, \tag{2.10}$$

where $[x]^+ = \max\{0, x\}$. Here $\hat{r}_{n[k-1]} > \hat{r}_{n[k]}$ indicates that a quality degradation occurs between $\hat{\boldsymbol{s}}_n[k-1]$ and $\hat{\boldsymbol{s}}_{n[k]}$, with a bitrate decrease of $\hat{r}_{n[k-1]} - \hat{r}_{n[k]}$.

(iii) *Rebuffering:* If a video buffer is exhausted before receiving a new

[7] As mentioned previously, precisely speaking, the value should be a function of video quality. Nevertheless, under the assumption that there is a distinct and monotonic relationship between bitrate and quality, we can write it as a function of bitrate for convenience.

[8] Our model can be directly extended to the case with upgrade loss, by simply change $[x]^+$ into the absolute operation $|x|$.

segment, the video player has to freeze the playback and rebuffer the video for a certain time. Such a freeze during playback is called *rebuffering.* The rebuffering during playback greatly affects the user QoE. By the buffer update rule (2.8), a rebuffering occurs when

$$b_{n[k-1]} < \hat{t}^{\mathrm{e}}_{n[k]} - \hat{t}^{\mathrm{e}}_{n[k-1]}, \tag{2.11}$$

with a detailed rebuffering time $\hat{t}^{\mathrm{e}}_{n[k]} - \hat{t}^{\mathrm{e}}_{n[k-1]} - b_{n[k-1]}$. Let $\phi_n^{\mathrm{REB}} > 0$ denote the value loss of user $n$ for one unit (second) of rebuffering time. Then, the total value loss of user $n$ induced by video rebuffering is

$$L_n^{\mathrm{REB}}(\widehat{\boldsymbol{S}}_n) \triangleq \sum_{k=2}^{|\widehat{\boldsymbol{S}}_n|} \phi_n^{\mathrm{REB}} \cdot \left[\hat{t}^{\mathrm{e}}_{n[k]} - \hat{t}^{\mathrm{e}}_{n[k-1]} - b_{n[k-1]}\right]^+ . \tag{2.12}$$

Based on the above, we can define the *utility* of each user $n$ under a receiving sequence $\widehat{\boldsymbol{S}}_n$ as follows:

$$U_n(\widehat{\boldsymbol{S}}_n) \triangleq V_n(\widehat{\boldsymbol{S}}_n) - L_n^{\mathrm{QD}}(\widehat{\boldsymbol{S}}_n) - L_n^{\mathrm{REB}}(\widehat{\boldsymbol{S}}_n). \tag{2.13}$$

*2)* ***Energy Cost:*** Users incur some energy cost in video streaming. Such energy cost mainly includes the energy consumptions for downloading data via cellular links (and Internet) and exchanging data via WiFi links.

(i) *Energy Consumption for Video Downloading (via Celluar and Internet):* When downloading data via the cellular link (and Internet), users' energy consumption depends on both the downloading time and the downloaded data volume [18]. Let $c_n^{\mathrm{TIME}} \geq 0$ denote the time-related energy consumption factor of user $n$ (i.e., for each unit of downloading time), and $c_n^{\mathrm{DATA}} \geq 0$ denote the volume-related energy consumption factor of user $n$ (i.e., for each unit of downloaded data). Then, the energy consumption of user $n$ for downloading video contents via cellular links and Internet is [18]:[9]

$$E_n^{\mathrm{CELL}}(\boldsymbol{S}_n) \triangleq \sum_{k=1}^{|\boldsymbol{S}_n|} \left( c_n^{\mathrm{TIME}} \cdot (t^{\mathrm{e}}_{n[k]} - t^{\mathrm{s}}_{n[k]}) + c_n^{\mathrm{DATA}} \cdot r_{n[k]} \cdot \beta_{u_{n[k]}} \right) . \tag{2.14}$$

[9] In this chapter, we will choose the detailed values of these energy consumption factors based on the existing literature such as [18]. The detailed measurement for these factors is beyond our scope.

(ii) *Energy Consumption for Video Exchanging (via WiFi):* When downloading a segment for others, the user needs to transmit the data to the segment owner via local WiFi link, the energy consumption of which also depends on the transmitting time and the transmitted data volume [18]. Let $w_n^{\mathrm{TIME}} \geq 0$ and $w_n^{\mathrm{DATA}} \geq 0$ denote the time-related and volume-related energy consumption factors of user $n$ on the WiFi link, respectively. The energy consumption of user $n$ for video exchanging on WiFi link is [18]:

$$\begin{aligned} E_n^{\mathrm{WIFI}}(\boldsymbol{S}_n) \triangleq \sum_{k=1}^{|\boldsymbol{S}_n|} \Big( w_n^{\mathrm{TIME}} \cdot 0 + w_n^{\mathrm{DATA}} \cdot r_{n[k]} \cdot \beta_{u_{n[k]}} \Big) \\ \cdot \mathbf{1}(u_{n[k]} \neq n), \end{aligned} \tag{2.15}$$

where $\mathbf{1}(u_{n[k]} \neq n) = 1$ if $u_{n[k]} \neq n$ (i.e., the segment $\boldsymbol{s}_{n[k]}$ is downloaded for others), and $\mathbf{1}(u_{n[k]} \neq n) = 0$ otherwise. Here we have assumed that the WiFi transmission time of a single segment is small and hence negligible.

Based on the above, we can derive the total *energy consumption* of each user $n$ under a downloading sequence $\boldsymbol{S}_n$ and receiving sequence $\widehat{\boldsymbol{S}}_n$ as follows:

$$C_n(\boldsymbol{S}_n, \widehat{\boldsymbol{S}}_n) \triangleq E_n^{\mathrm{CELL}}(\boldsymbol{S}_n) + E_n^{\mathrm{WIFI}}(\boldsymbol{S}_n). \tag{2.16}$$

*3)* ***Payoff:*** The payoff of each user $n$, denoted by $P_n$, is defined as the difference between the utility (capturing the QoE of users) and the cost (capturing the energy consumption), i.e.,

$$P_n(\boldsymbol{S}_n, \widehat{\boldsymbol{S}}_n) \triangleq U_n(\widehat{\boldsymbol{S}}_n) - C_n(\boldsymbol{S}_n, \widehat{\boldsymbol{S}}_n) \tag{2.17}$$

The *social welfare* is the aggregate payoff of all users:

$$W(\boldsymbol{S}_1, ..., \boldsymbol{S}_N) \triangleq \sum_{n=1}^{N} P_n(\boldsymbol{S}_n, \widehat{\boldsymbol{S}}_n), \tag{2.18}$$

where the receiving sequence $\widehat{\boldsymbol{S}}_n$ of each user $n$ can be derived from the downloading sequences $\boldsymbol{S}_n, n \in \mathcal{N}$.

### 2.4.4 Problem Formulation

Our purpose is to find the proper download scheduling to maximize the social welfare achieved in the proposed crowdsourced streaming model.

First, in an ideal scenario with the complete network information, we can formulate the following *offline* social welfare maximization problem:

$$\begin{aligned} \max_{\{\boldsymbol{S}_n, n \in \mathcal{N}\}} \quad & W(\boldsymbol{S}_1, ..., \boldsymbol{S}_N), \\ \text{s.t.} \quad & \text{C.1} \sim \text{C.4}. \end{aligned} \tag{2.19}$$

To solve this *offline* optimization problem, we need to know the complete network information. The solution of (2.19), denoted by $W^*$, provides the theoretical performance bound (in term of social welfare) of the proposed crowdsourced system. We will analyze such a performance bound in Section 2.5.

Second, in a more general scenario without complete (future) network information, we need to design *online* scheduling algorithms, where the downloading operation of each user is performed in an online and distributed manner. We will study such an online scheduling algorithm design and the associated performance evaluation in Section 2.6.

## 2.5 Performance Bound Analysis

In this section, we study the theoretical social welfare performance bound of the proposed crowdsourced system (i.e., the solution of the offline social welfare maximization problem (2.19)), which serves as a benchmark for the online scheduling solutions in Section 2.6.

However, directly solving (2.19) is challenging due to the following reasons. First, users operate in an asynchronous manner. Namely, different users may start to download new segments at different times. Second, (2.19) involves both discrete variables (e.g., $u$ and $z$) and continuous variables (e.g., $t^{\mathrm{s}}$ and $t^{\mathrm{e}}$), hence is a complicated mixed-integral optimization problem. Third, (2.19) involves the integral operation (C.2), which is even more challenging. Hence, in what follows, we will focus on finding upper-bound and lower-bound for the desired performance bound of the crowdsourced system.

To achieve this, we propose a virtual *time-slotted download operation* scheme, under which the problem can be formulated as an linear programming, hence can be solved by many classic methods. We will show that the solution of (2.19) under the segmented operation scheme (i.e., the theoretical performance bound of the proposed crowdsourced system) is bounded by the solutions under this virtual time-slotted system. **It is important to note that this time-slotted operation scheme is only used for characterizing the theoretical performance bound, but not for the practical implementation.**

### 2.5.1 Time-Slotted Download Operation

To model the time-slotted operation scheme, we divide the whole time period $[0, T]$ into multiple time slots, each with the same length. For convenience, we normalize the length of each slot to be one. Hence, there is a set of $T$ time slots, denoted by $\mathcal{T} = \{1, 2, ..., T\}$, with the $\tau$-th slot corresponding to time interval $[\tau - 1,\ \tau]$.

Under the time-slotted operation scheme, each video is downloaded *slot by slot* in a synchronized manner, rather than segment by segment under the segmented operation. Thus, in this case, we can focus on the segments that each user downloads in each time slot, instead of the segment downloading sequence. Moreover, to guarantee the synchronous operation, we require that each segment must be completely downloaded within one time slot. Namely, users cannot download a segment across multiple time slots.

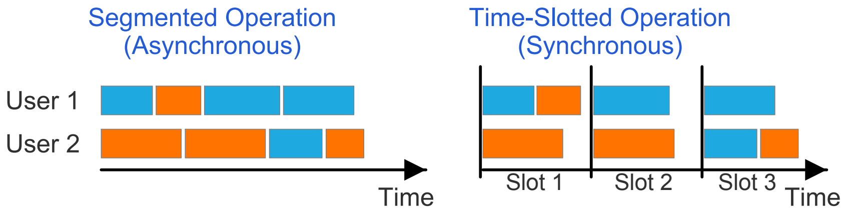


Figure 2.3: Segmented vs time-slotted operation.

For clarity, we illustrate the difference (in download scheduling) between the segmented operation and the time-slotted operation in Figure 2.3, where blue blocks denote user 1's data and orange blocks denote user 2's data. Under the segmented operation (left), users start to download data at different times, while under the time-slotted operation (right), users are synchronized, and download data at the beginning of each time slot.

*1)* ***Downloading Vector:*** With the time-slotted operation, the downloading operation of each user $n$ can be characterized by a downloading vector:

$$\boldsymbol{K}_n \triangleq \left\{\kappa_{n,m}^z(\tau), \quad \forall \tau \in \mathcal{T}, m \in \mathcal{N}, z \in \{1, ..., Z\}\right\}, \tag{2.20}$$

where each element $\kappa_{n,m}^z(\tau)$ is a non-negative integer, denoting the total number of segments with bitrate level $z$ that user $n$ downloads for user $m$ in time slot $\tau$.

Given the downloading vector $\boldsymbol{K}_n$, we can derive the total data that user $n$ downloads in each time slot $\tau$:

$$x_n^{\mathrm{DL}}(\tau) = \sum_{m=1}^{N} x_{n,m}(\tau) = \sum_{m=1}^{N}\sum_{z=1}^{Z} \kappa_{n,m}^z(\tau) \cdot \beta_m \cdot R_m^z, \tag{2.21}$$

where $x_{n,m}(\tau) \triangleq \sum_{z=1}^{Z} \kappa_{n,m}^z(\tau) \cdot \beta_m \cdot R_m^z$ is the amount of data for user $m$ in slot $t$. Then, we can define the link capacity constraint and encounter constraint for a feasible downloading vector $\boldsymbol{K}_n$:

$$\begin{aligned} \widetilde{\mathrm{C}}.2: &\quad x_n^{\mathrm{DL}}(\tau) \leq H_n(\tau), \\ \widetilde{\mathrm{C}}.3: &\quad e_{n,m}(t) = 1, t \in [\tau - 1, \tau], \text{ if } x_{n,m}(\tau) > 0, \end{aligned} \tag{2.22}$$

where $H_n(\tau) = \int_{\tau-1}^{\tau} h_n(t)\mathrm{d}t$ is the total cellular link capacity of user $n$ in time slot $\tau$. Note that here we do not need to consider the timing constraint (C.1) as the operation is already slot by slot.

*2)* ***Receiving Vector:*** Given feasible downloading vectors of all users, i.e., $\boldsymbol{K}_n, \forall n \in N$, we can derive the total playback time that user $m$ receives in each time slot $\tau$:

$$y_m^{\mathrm{RE}}(\tau) = \sum_{n=1}^{N} y_{n,m}(\tau) = \sum_{n=1}^{N} \sum_{z=1}^{Z} \kappa_{n,m}^{z}(\tau) \cdot \beta_m, \tag{2.23}$$

where $y_{n,m}(\tau) \triangleq \sum_{z=1}^{Z} \kappa_{n,m}^{z}(\tau) \cdot \beta_m$ is the total playback time that user $m$ receives from user $n$ in slot $\tau$.

Let $b_m(\tau)$ denote the buffer level (in seconds) of user $m$ *at the end of time slot* $\tau$. Then, we have the following ***buffer update rule*** for user $m$:

$$b_m(\tau) = [b_m(\tau - 1) - 1]^{+} + y_m^{\mathrm{RE}}(\tau), \tag{2.24}$$

where $[x]^{+} = \max\{0, x\}$. Here one time unit of video is played back during time slot $\tau$, and $y_m^{\mathrm{RE}}(\tau)$ is the playback time of the newly received segments in slot $\tau$.

Similarly, we have the following buffer constraint:

$$\widetilde{\mathrm{C}}.4: \quad 0 \le b_m(\tau) \le B_m, \quad \forall \tau = 1, ..., T. \tag{2.25}$$

*3)* ***User Payoff:*** Now we define the user payoff and social welfare under the time-slotted operation.

(i) *Video Quality:* Similar as (2.9), the value that user $n$ achieves from all received segments is:

$$\widetilde{V}_n \triangleq \sum_{\tau=1}^{T} \sum_{m=1}^{N} \sum_{z=1}^{Z} \kappa_{m,n}^{z}(\tau) \cdot \beta_n \cdot v_n(R_n^z). \tag{2.26}$$

(ii) *Quality Fluctuation:* Without loss of generality, we assume that all the received segments of each user $n$ in each time slot $\tau$ are sorted in ascending order of bitrate.[10] Hence, quality degradation only occurs between two suc-

[10] If not, we can simply exchange the violated segments in the corresponding downloading vector.

cessive time slots, while never occurs within a time slot. Let $r_n^{\mathrm{H}}(\tau)$ and $r_n^{\mathrm{L}}(\tau)$ denote the highest bitrate and lowest bitrate that user $n$ receives in slot $\tau$. Then, similar as (2.10), the value loss of user $n$ due to quality degradation is:

$$\widetilde{L}_n^{\mathrm{QDEG}} \triangleq \sum_{\tau=2}^{T} \phi_n^{\mathrm{QD}} \cdot \left[r_n^{\mathrm{H}}(\tau-1) - r_n^{\mathrm{L}}(\tau)\right]^+ , \tag{2.27}$$

(iii) *Rebuffering:* By the buffer update rule in (2.24), a rebuffering occurs in time slot $\tau$ when

$$b_m(\tau-1) < 1, \tag{2.28}$$

with a rebuffering time $1 - b_m(\tau-1)$. Then, similar as (2.12), the value loss of user $n$ induced by rebuffering is

$$\widetilde{L}_n^{\mathrm{REBUF}} \triangleq \sum_{\tau=2}^{T} \phi_n^{\mathrm{REB}} \cdot \left[1 - b_m(\tau-1)\right]^+ . \tag{2.29}$$

(iv) *Energy Consumption for Video Downloading (via Cellular and Interent):* Similar as (2.14), the energy consumption of user $n$ for downloading video is

$$\widetilde{E}_n^{\mathrm{CELL}} \triangleq \sum_{\tau=1}^{T} \left( c_n^{\mathrm{TIME}} \cdot \frac{x_n^{\mathrm{DL}}(\tau)}{H_n(\tau)} + c_n^{\mathrm{DATA}} \cdot x_n^{\mathrm{DL}}(\tau) \right) , \tag{2.30}$$

where $\frac{x_n^{\mathrm{DL}}(\tau)}{H_n(\tau)}$ is the actual downloading time in slot $\tau$.

(v) *Energy Consumption for Video Exchanging (via WiFi):* Similar as (2.15), the energy consumption of user $n$ for exchanging video on local WiFi links is

$$\widetilde{E}_n^{\mathrm{WIFI}} \triangleq \sum_{\tau=1}^{T} \sum_{m=1, m\neq n}^{N} \left( w_n^{\mathrm{TIME}} \cdot 0 + w_n^{\mathrm{DATA}} \cdot x_{n,m}(\tau) \right) . \tag{2.31}$$

Based on the above, the payoff of each user $n$ is

$$\widetilde{P}_n(\boldsymbol{K}_1, ..., \boldsymbol{K}_N) \triangleq \widetilde{V}_n - \widetilde{L}_n^{\mathrm{QDEG}} - \widetilde{L}_n^{\mathrm{REBUF}} - \widetilde{E}_n^{\mathrm{CELL}} - \widetilde{E}_n^{\mathrm{WIFI}}. \tag{2.32}$$

***4) Problem Formulation under Time-Slotted Operation:*** Now we can define the social welfare maximization problem under the time-slotted download operation:

$$\begin{aligned} \max_{\{\boldsymbol{K}_n, n\in\mathcal{N}\}} \quad & \widetilde{W} \triangleq \sum_{n=1}^{N} \widetilde{P}_n(\boldsymbol{K}_1, ..., \boldsymbol{K}_N), \\ \text{s.t.} \quad & \widetilde{\text{C}}.2 \sim \widetilde{\text{C}}.4. \end{aligned} \tag{2.33}$$

Similar to (2.19), this is an *offline* optimization problem and requires the complete network information. Moreover, (2.33) is an integer programming, and can be solved by many classic methods. Hence, we skip the detailed derivations. For notation convenience, we denote the solution of (2.33) by $\widetilde{W}^*$.

### 2.5.2 Performance Bound

Now we characterize the theoretical performance bound $W^*$ of the proposed crowdsourced system (under the segmented operation) by using the solution $\widetilde{W}^*$ of (2.33) under the virtual time-slotted operation.

For convenience, we denote $\boldsymbol{\beta} \triangleq (\beta_1, ..., \beta_N)$ as the vector consisting of all users' segment lengths, and denote $W^*_{(\boldsymbol{\beta})}$ and $\widetilde{W}^*_{(\boldsymbol{\beta})}$ as the solutions of (2.19) and (2.33) under $\boldsymbol{\beta}$, respectively. We refer to a vector $\boldsymbol{\beta}$ as an *integer multiple* of another vector $\boldsymbol{\beta}'$, if each element $\beta_n$ in $\boldsymbol{\beta}$ is an integer multiple of the corresponding element $\beta'_n$ in $\boldsymbol{\beta}'$. For example, $\boldsymbol{\beta} = (1, ..., N)$ is an integer multiple of $\boldsymbol{\beta}' = (0.5, ..., N/2)$.

**Proposition 2.1.** *If $\boldsymbol{\beta}$ is an integer multiple of $\boldsymbol{\beta}'$, then*

$$W^*_{(\boldsymbol{\beta})} \leq W^*_{(\boldsymbol{\beta}')}, \quad \textit{and} \quad \widetilde{W}^*_{(\boldsymbol{\beta})} \leq \widetilde{W}^*_{(\boldsymbol{\beta}')}. \tag{2.34}$$

This proposition can be proved by showing that in both schemes, any downloading operation under $\boldsymbol{\beta}$ can be equivalently achieved under $\boldsymbol{\beta}'$.

**Proposition 2.2.** *If $\boldsymbol{\beta} \to \mathbf{0}$ (i.e., $\beta_n \to 0, \forall n \in \mathcal{N}$), then*

$$W^*_{(\boldsymbol{\beta})} = \widetilde{W}^*_{(\boldsymbol{\beta})}. \tag{2.35}$$

This proposition can be proved by showing that with infinitely small segment lengths $\boldsymbol{\beta} \to \mathbf{0}$, any downloading operation under the time-slotted operation scheme can be equivalently achieved under the segmented operation scheme, and vise versa.

**Proposition 2.3.** *If $\boldsymbol{\beta} \succeq \mathbf{0}$ is a finite vector (i.e., each element $\beta_n \geq 0$ is a finite number), then*

$$W^*_{(\boldsymbol{\beta})} \geq \widetilde{W}^*_{(\boldsymbol{\beta})}. \tag{2.36}$$

This proposition can be proved by showing that with finite segment lengths $\boldsymbol{\beta} \succeq \mathbf{0}$, any downloading operation under the time-slotted operation scheme can be equivalently achieved under the segmented operation scheme, but *not* vise versa.

Based on the above, we have the following theorem.

**Theorem 2.1.** *Given a segment length $\boldsymbol{\beta}$, the theoretical performance upper-bound $W^*_{(\boldsymbol{\beta})}$ is bounded by:*

$$\widetilde{W}^*_{(\boldsymbol{\beta})} \leq W^*_{(\boldsymbol{\beta})} \leq \widetilde{W}^*_{(\boldsymbol{\beta}' \to \mathbf{0})}. \tag{2.37}$$

Intuitively, this theorem states that with any $\boldsymbol{\beta}$, the theoretical performance bound $W^*_{(\boldsymbol{\beta})}$ of our proposed crowdsourced system is (a) lower-bounded by $\widetilde{W}^*_{(\boldsymbol{\beta})}$ (i.e., the optimal performance of the virtual time-slotted system with the same segment length vector $\boldsymbol{\beta}$), and (b) upper-bounded by $\widetilde{W}^*_{(\boldsymbol{\beta}' \to \mathbf{0})}$ (i.e., the optimal performance of the virtual time-slotted system with infinitely small segment lengths $\boldsymbol{\beta}' \to \mathbf{0}$). Therefore, the performance of the virtual time-slotted system under different $\boldsymbol{\beta}$ characterizes the theoretical performance region of our proposed crowdsourced system.

## 2.6 Online Scheduling Algorithms

In the previous section, we have analyzed the theoretical performance bound of the proposed crowdsourced system, which is achievable in an ideal scenario with complete network information. In practice, however, network changes randomly over time, and hence it is difficult to obtain the future and global network information.

In this section, we study the practical scenario where the future and global network information is not available. We propose an online scheduling algorithm based on the Lyapunov optimization framework [30], which relies only on the current local network information and the scheduling history, while not on any future or global network information.

### 2.6.1 Online vs Offline

We first discuss the key difference between online scheduling and offline scheduling. In the offline scheduling, the segment downloading sequences of all users at all time are determined in advance, through, for example, the offline social welfare maximization problem (2.19), which requires the complete network information. In the online scheduling, however, each user makes the download scheduling decision (regarding the next segment to be downloaded) in real time, e.g., at the time when he completes a previous segment downloading.

In our proposed crowdsourced system, such a real time downloading decision mainly includes two problems: *whose segment to be downloaded, and at which bitrate level?* The decision may depend on different criteria such as the real time user buffer levels (e.g., in [51]), the channel bandwidth or throughput predictions (e.g., in [65]), and other specific objective functions (e.g., Lyapunov drift-plus-penalty described below).

### 2.6.2 Lyapunov-Based Online Scheduling

Lyapunov optimization [70] is a widely used technique for solving stochastic optimization problems with time average constraints. In our model, an implicit time average constraint is that the average segment arriving rate should be same as the video playback rate in term of segment.[11] If the video playback rate is smaller, then the downloaded segments will be frequently dropped due to the limited buffer size; if the video playback rate is larger, then the rebuffering will frequently happen. Both cases are not desirable in this system. To this end, we introduce the Lyapunov optimization technique to optimize the downloading scheduling in an online manner.

Suppose that a user $n$ completes a segment downloading at time $t$, and needs to make the downloading decision regarding the next segment to be downloaded. We denote such a decision by $(u, z)$, where $u \in \mathcal{N}$ is the owner of the segment to be downloaded, and $z \in \{1, ..., Z\}$ is the bitrate level of the segment to be downloaded. Obviously, a feasible decision $(u, z)$ of user $n$ at time $t$ satisfies the following user encounter constraint: $e_{n,u}(t) = 1$.

For analytical convenience, we further denote $b_m(t)$ as the buffer level of each user $m$ at time $t$, and denote $r_m$ as the bitrate of user $m$'s *last received* segment. This information captures the current network state and historical scheduling information that can be observed.

*1)* ***Objective Function:*** Given a feasible decision $(u, z)$ of user $n$, the data volume to be downloaded is $R_u^z \cdot \beta_u$ (Mbit), and the *estimated* downloading time is $\gamma_{u,z} \triangleq \frac{R_u^z \cdot \beta_u}{h_n(t)}$.[12] The total energy consumption of user $n$ (*for this particular downloading operation*) and user $u$ (for playing the downloaded

[11] For example, for a video with 2-second segment, the playback rate in term of segment is 0.5 (segments per second).

[12] Here we use the current channel capacity $h_n(t)$ to approximate the capacity in a period of future time. Note that the actual downloading time may be different from $\gamma_{u,z}$ due to the channel stochastic.

segment) is:

$$C_n(u,z) = E_n^{\text{CELL}} + E_n^{\text{WIFI}}; \tag{2.38}$$

The utility of receiver $u$ *on this particular segment* is

$$U_u(u,z) = V_u - L_u^{\text{QD}} - L_u^{\text{REB}} \tag{2.39}$$

The utility of other user $m \neq u$ due to this operation is

$$U_m(u,z) = L_m^{\text{REB}} = -\phi_m^{\text{REB}} \cdot [\gamma_{u,z} - b_m(t)]^+, \tag{2.40}$$

which only includes the potential rebuffering loss.

Therefore, the total payoff generated under $(u,z)$ is

$$P(u,z) \triangleq \sum_{m=1}^{N} U_m(u,z) - C_n(u,z). \tag{2.41}$$

*2)* ***Lyapunov Drift:*** Following the Lyapunov framework, we define a modified Lyapunov function:

$$J(t) \triangleq \frac{1}{2} \sum_{m=1}^{N} [B_m - b_m(t)]^2. \tag{2.42}$$

The *Lyapunov drift* is the change of Lyapunov function (from one decision-making time to the next), i.e.,

$$\begin{aligned} \Delta(t) &\triangleq J(t+\gamma_{u,z}) - J(t), \\ &= \frac{1}{2} \sum_{m=1}^{N} \left( [B_m - b_m(t+\gamma_{u,z})]^2 - [B_m - b_m(t)]^2 \right), \end{aligned} \tag{2.43}$$

where $b_m(t+\gamma_{u,z})$ is the estimated buffer level of user $m$ at time $t+\gamma_{u,z}$ (i.e., the next decision-making time of user $n$). For the receiver $u$, the estimated buffer level is:

$$b_u(t+\gamma_{u,z}) = \min\{B_u,\ [b_u(t) - \gamma_{u,z}]^+ + \beta_u\}; \tag{2.44}$$

For other user $m \neq u$, the estimated buffer level is:

$$b_m(t+\gamma_{u,z}) = [b_m(t) - \gamma_{u,z}]^+; \tag{2.45}$$

**Algorithm 1** Lyapunov-based Online Scheduling

```
while at each decision-making time $t$ do
    if $b_n(t) + \beta_n > B_n, \forall n \in \mathcal{N}$ then /* no buffer can afford one more segment */
Wait for $T_w = \min_{n \in \mathcal{N}}(b_n(t) + \beta_n - B_n)$ seconds;
    elseDownload a segment of bitrate level $z^*$ for user $u^*$:
$(u^*, z^*) = \arg\min_{u,z} \ \Phi(t) \triangleq \Delta(t) - \lambda \cdot P(u, z)$
    end if
end while
```

*3)* ***Online Scheduling Algorithm:*** By the Lyapunov optimization theorem, to stabilize the system while optimizing the objective, we can use such a scheduling policy that greedily minimizes *drift-plus-penalty*:

$$\Phi(t) \triangleq \Delta(t) - \lambda \cdot P(u, z), \tag{2.46}$$

where the negative payoff ($-P(u, z)$) is viewed as the penalty incurred at time $t$, and $\lambda \geq 0$ is a control parameter. It is important to note that the buffer levels (appearing in $\Delta(t)$) serve as regulation factors, such that the user with a larger idle buffer can be more likely to be scheduled (hence reducing the possibility of rebuffering). This term is different from the rebuffering loss in (2.12), which is the actually realized loss when a rebuffering event actually happens.

Based on the above analysis, we now design an on-line algorithm that aims at minimizing the drift-plus-penalty (2.46) in each decision-making time. We present the detailed algorithm in Algorithm 2.6.2. Note that a user may decide *not* to download any segment at a decision-making time, when, for example, all buffers are full and cannot afford one more segment. In this case, the user will wait for a certain time and then trigger decision-making event again. Hence, *a decision-making time can be either the time that a user completes a segment downloading, or the time that a user is triggered by the waiting timer.*

To implement Algorithm 2.6.2 in practice, users need to exchange certain information to coordinate the downloading decision and to avoid the redundant downloading of the same segment. The following two information exchanging processes are necessary. First, users periodically broadcast their buffer levels to nearby users through WiFi, such that each user can compute the accurate buffer levels of all neighbors when he needs to make a downloading decision. Second, when a user decides to download a segment for another user, he needs to query the latter for the detailed information of the segment to be downloaded. This can avoid the redundant downloading by another user.

*4)* ***Performance Analysis:*** Now we analyze the performance of Algorithm 2.6.2. Let $t_{[k]}$ denote the $k$-th decision-making time (counting all users), and let $P_{[k]}$ denote the associated payoff achieved in the $k$-th download operation. Then, the social welfare generated by Algorithm 2.6.2 during the whole time $[0, T]$ can be computed by:

$$W'_{(\boldsymbol{\beta})} = \sum_{t_{[k]} \leq T} P_{[k]}. \tag{2.47}$$

By the Lyapunov optimization theorem (Theorem 4.2 in [70]), we obtain the following gap for $W'_{(\boldsymbol{\beta})}$ and $W^*_{(\boldsymbol{\beta})}$, i.e., the theoretical performance upperbound.

**Theorem 2.2.**

$$\lim_{T \to \infty} \mathrm{E}\left[W'_{(\boldsymbol{\beta})}\right] \geq \mathrm{E}\left[W^*_{(\boldsymbol{\beta})}\right] - \frac{Q}{\lambda}, \tag{2.48}$$

*where* E[.] *is expectation, and $Q$ is a positive constant.*

Theorem 2.2 shows that Algorithm 2.6.2 converges to the theoretical performance bound $W^*_{(\boldsymbol{\beta})}$ asymptotically, with a controllable approximation error bound $O(\frac{1}{\lambda})$.

However, this theorem does not directly help us calculate the *actual gap* between $W'_{(\boldsymbol{\beta})}$ and $W^*_{(\boldsymbol{\beta})}$ in a particular experimental scenario, as $T$ is finite in practice. To this end, we propose another approach based on Theorem 1 for the practical calculation of the actual gap, i.e.,

$$\left|W^*_{(\boldsymbol{\beta})} - W'_{(\boldsymbol{\beta})}\right| \leq \left|\widetilde{W}^*_{(\boldsymbol{\beta}' \to \mathbf{0})} - W'_{(\boldsymbol{\beta})}\right|. \tag{2.49}$$

Note that $\widetilde{W}^*_{(\boldsymbol{\beta}' \to \mathbf{0})}$ is the solution of the integer programming problem (2.33), and can be easily computed in a practical experiment after collecting the complete network information. In our experiments, the average gap between $W'_{(\boldsymbol{\beta})}$ and $\widetilde{W}^*_{(\boldsymbol{\beta}' \to \mathbf{0})}$ is smaller than 3%.

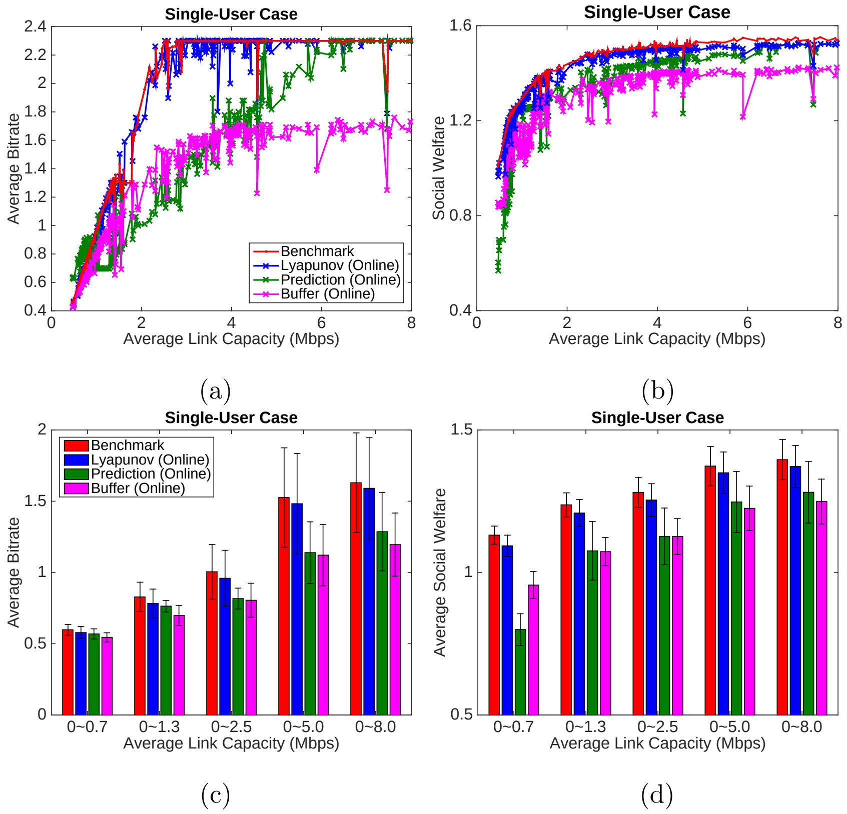
## 2.7 Experiments and Performance

### 2.7.1 Experiment Setting

*1)* ***Datasets:*** To evaluate the realistic performance of our proposed crowd-sourced system, we conduct experiments based on real data traces from two datasets: ISF [3] and NWF [4].[13] Both datasets trace the user access sessions at a set of WiFi hotspots in different countries during a long period of time (3 years for ISF and 5 months for NWF). Each session records the information of one access (of a particular user at a particular hotspot), including user id, hotspot id, login and logout time, incoming and outgoing traffic volumes, etc. These two datasets represent two different (hotspot-based) mobility scenarios: Users encounter more frequently in NWF, while the duration of each encounter is larger in ISF. Moreover, the energy consumption factors are chosen according to the real measurement given in [18, 73].

To simulate the video watching behaviours of mobile users and the real cellular link throughputs for video streaming, we use the video viewing session logs obtained from BestTV [1], one of the largest OTT (Over The Top) video service providers in China. Every session records the information of

[13] ISF is provided by a non-profit organization "Ile Sans Fil" in Canada, and is open source (available at CRAWDAD [3]). NWF is obtained from a wireless service provider "NextWiFi" in China [4].



(a)

(b)

(c)

(d)

Figure 2.4: Single-user case: (a) average bitrate in each experiment, (b) social welfare in each experiment, (c) average bitrate in 1000 experiments, (d) average social welfare in 1000 experiments.

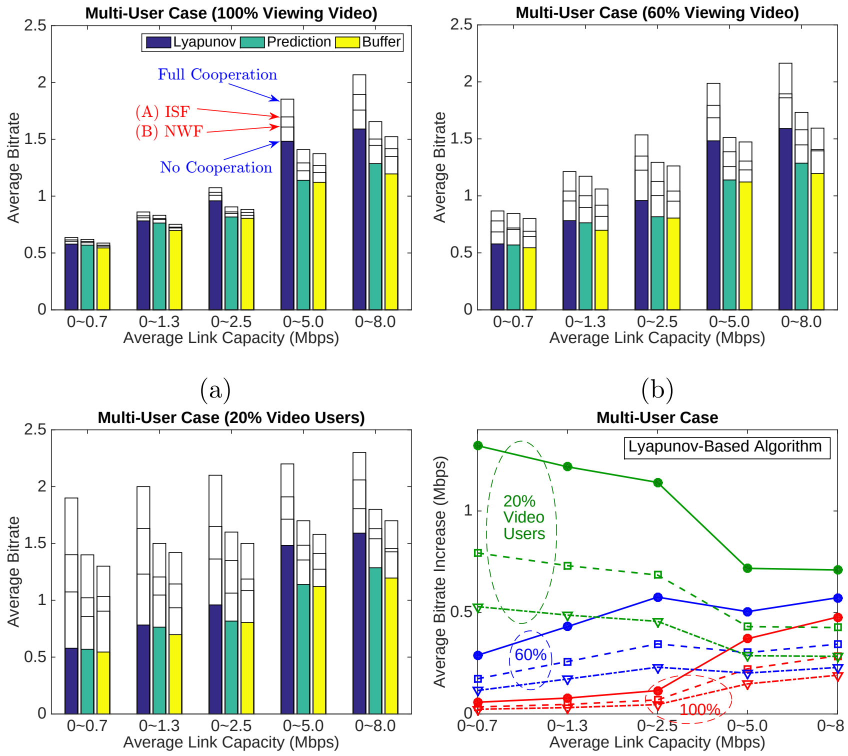
one video playing (of a particular mobile user towards a particular video), containing the detailed records of all segments, each including the user id, video id, segment id (playback index), segment length, resolution, bitrate, downloading time, etc. There are 5 different bitrate levels (for mobile users) in this dataset: $\{0.2, 0.4, 0.7, 1.3, 2.3\}$Mbps, corresponding to the lowest to the highest video resolutions, respectively. Based on the segment length, bitrate, and downloading time, we can calculate the *measured* end-to-end link throughput for each segment downloading. We use this measured throughput to approximate the cellular link capacity in our experiments. From [1], we find that 10% users experience a throughput lower than 0.7Mbps, 40% users experience a throughput lower than 1.3Mbps, and so on.

*2)* ***Existing Online Algorithms:*** To evaluate the performance of our proposed Lyapunov-based online algorithm, we also perform simulations using the following two typical existing online algorithms: Buffer-based algorithm [51] and Channel Prediction-based algorithm [65]. Specifically, buffer-based algorithm [51] introduces a linear mapping between buffer and bitrate, and selects the next segment bitrate based on the current buffer level: *a higher buffer level is mapped to a higher bitrate.* Channel prediction-based algorithm [65] proposes a channel prediction method, and selects the next segment bitrate based on the predicted channel capacity: *the highest bitrate that can be supported by the predicted channel capacity.*

Note that both algorithms in [51] and [65] were designed for the single-user scenario, and considered only the bitrate adaptation. In the multi-user scenario, we need to consider both bitrate adaptation and segment owner selection (i.e., whose segment to be downloaded) as discussed in Section 2.6. To this end, we introduce the following segment owner selection policy for these two algorithms in the multi-user scenario: *Each user $n$ will choose to download the next segment for another user $u \neq n$, if and only if (i) $b_n \geq \delta_{\text{TH}} \cdot B_n$, (ii) $b_n - b_u \geq \Delta_{\text{TH}}$, and (iii) $b_u = \min_{m \in \mathcal{N}} b_m$.* Intuitively, user $n$ will choose to help the user with the lowest buffer level, if his own buffer level is higher than a ratio threshold $\delta_{\text{TH}}$ and meanwhile is higher than the lowest buffer level by a threshold $\Delta_{\text{TH}}$. In our experiments, we will try different values of $\delta_{\text{TH}}$ and $\Delta_{\text{TH}}$, and choose the best ones.

### 2.7.2 Single-User Case

We first construct experiments for the single-user scenario (i.e., non-cooperative scenario), where the user plays a high-resolution video (bitrate 2.3Mbps). The total video length is 500 seconds, the segment length is 2 seconds, and the



(a)

(b)

(c)

(d)

Figure 2.5: Multi-user case: (a) average bitrate with 100% video users, (b) average bitrate with 60% video users, (c) average bitrate with 20% video users, (d) average bitrate increase under the Lyapunonv-based algorithm.

maximum buffer length at the user's device is 40 seconds. We use these experiments to illustrate the performance gap of our proposed Lyapunov-based online algorithm to the theoretical performance bound. We also use these experiments to compare the bitrate adaptation performance of our proposed algorithm with the existing online algorithms.

Figure 2.4 shows the average bitrate and social welfare under different average link capacities (extracted from the measured link throughput traces). Red curve/bar denotes the theoretical upperbound (benchmark). Blue curve/bar denotes the proposed Lyapunov-based online algorithm. Green curve/bar denotes the channel prediction-based algorithm in [65], and Pink curve/bar denotes the buffer-based algorithm in [51]. Each point in subfigures (a) and (b) denotes the average bitrate and social welfare generated in one experiment (corresponding to a particular choice of data trace), respectively. Subfigures (c) and (d) shows the average bitrate and average social welfare in 1000 experiments under different average link capacity ranges. For example, in the first bar group, we calculate the average bitrate and average social welfare achieved in all experiments with an average link capacity below 0.7Mbps.

We can see from (a) and (c) that our proposed algorithm (Blue) achieves an average bitrate higher than other two algorithms (with an average bitrate increase of 5% $\sim$ 30%), and is very close to the offline benchmark (Red). We can further see from (b) and (d) that our proposed algorithm achieves an average social welfare higher than other two algorithms (with an average gain of 10% $\sim$ 40%), and is very close to the theoretical upperbound (with an average gap less than 3%). Moreover, the social welfare gain decreases with the maximum link capacity. This is because with a larger link capacity, all algorithms approach to the upperbound, hence their differences become less significant.

The above experiments demonstrate that the bitrate adaptation mechanism in our algorithm is better than those in [51] and [65]. By Theorem 2.2, our algorithm asymptotically converges to the theoretical performance upperbound (with a controllable gap), while the other two algorithms represent some reasonable heuristics without a theoretical performance guarantee.

### 2.7.3 Multiple-User Case

Now we perform experiments for the multi-user scenario, where some users play videos (called video users), while others remain idle and can potentially help the encountered video users. For simplicity, we assume that all video users play the high-resolution videos (bitrate 2.3Mbps). The video configuration (such as video length, segment length, and buffer size) is same as those in the single-user case. We use these multi-user experiments to illustrate both the cooperation gain of the proposed crowdsourced system and the performance gain of the proposed algorithm.

In the following experiments, we consider a total of 50 users and randomly choose a subset of users as video users. We consider different network conditions, characterized by the *range* of the average link capacity. For example, a bad network condition corresponds to a range $[0, 0.7]$Mbps, under which each user will be randomly assigned by a real data trace with an average link capacity smaller than 0.7Mbps.

*1)* ***Average Bitrate:*** Figure 2.5 shows the average bitrates with different percentages of video users under different network conditions. For each video user percentage and network condition, we perform experiments with the three algorithms under ISF and NWF mobility traces, corresponding to different encountering scenarios (hence different cooperation probabilities). To fully characterize the cooperation gain, we also run the algorithms under two benchmark encountering scenarios: (i) a full cooperation scenario, where all users are always encountered with each other, and (ii) a non-cooperative scenario, where none of users are encountered. Sugfigures (a) to (c) show the average bitrates with 100%, 60%, and 20% video users, respectively. As illustrated in (a), the solid bar denotes the average bitrate under the non-cooperative scenario, and the hollow bar denotes the average bitrate under the full cooperation, in which the first (higher) line denotes the average bitrate under ISF (with a higher encountering probability) and the second (lower) line denotes the average bitrate under NWF (with a lower encountering probability). Subfigure (d) shows the average bitrate *increase* (i.e., the cooperation gain) with our proposed Lyapunov-based algorithm, comparing with the achieved bitrate under the non-cooperative scenario. The solid-dot lines denote the cooperation gain under the full cooperation scenario, the dash-square and dash-triangle lines denote the cooperation gain under the ISF and NWF, respectively.

From subfigure (a), we can see that when the percentage of (high-resolution) video users is very high (e.g., 100%), the increase of bitrate is very small under a low link capacity range (e.g., lower than 2.5Mbps), as in this case all users are lack of capacity, hence nobody can help other users significantly. Under a high link capacity range (e.g., $[0, 5]$Mbps and $[0, 8]$Mbps), the increase of bitrate becomes significant, as some users may have redundant capacities, hence can help others. From subfigures (b) and (c), we can see that when the percentage of video users is low (e.g., 60% or 20%), the bitrate increase is significant under all network conditions, mainly due to the contributions of the idle users.

Subfigure (d) summarizes the increase of bitrate under our proposed algorithm. We can see that with 100% video users, the increase of bitrate continuously increases with the link capacity, as a larger capacity gives the video users more opportunities to obtain redundant capacity and help others. With 20% video users, however, the increase of bitrate continuously decreases with the link capacity, as a very small capacity already leads to a considerably high bitrate (due to the contributions of a large population of idle users), hence the increase of bitrate is more significant under a small capacity (as the benchmark bitrate is smaller). With 60% video users, the increase of bitrate first increases with the link capacity (due to a similar reason in the 100% case), and then decreases with the link capacity (due to a similar reason in the 20% case). The maximum bitrate increase ratio under the full cooperation scenario can be up to $50\% \sim 230\%$ with 20% video users, $35\% \sim 60\%$ with 60% video users, and $4\% \sim 40\%$ with 100% video users. Moreover, the bitrate increase under the real data traces is bounded by the above maximum

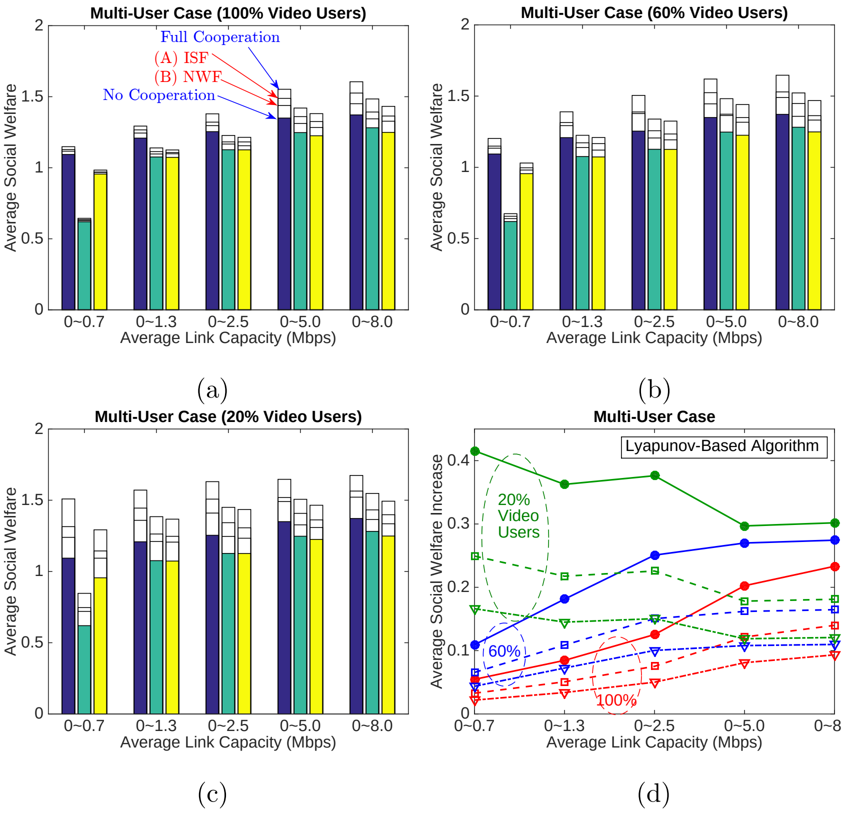


Figure 2.6: Multi-user case: (a) social welfare with 100% video users, (b) social welfare with 60% video users, (c) social welfare with 20% video users, (d) average social welfare increase under the Lyapunonv-based algorithm.

ration, and actually depends on the encountering probability. In our experiments, the bitrate increases under ISF and NWF can reach around 60% and 40% of the maximum bitrate increase, respectively.

*2) **Social Welfare:*** Figure 2.6 shows the average social welfares and welfare gains with different percentages of video users under different network conditions. The key informations and observations regarding the social welfare are similar as those regarding the average bitrate in Figure 2.5, hence we skip the detailed discussions and only present the results regarding the cooperation gain. Specifically, using our proposed algorithm, the maximum social welfare increase ratio (under the full cooperation scenario) can be up to 20% $\sim$ 40% with 20% video users, 10% $\sim$ 20% with 60% video users, and 5% $\sim$ 15% with 100% video users. The social welfare increase under ISF and NWF can reach 60% and 40% of the maximum welfare increase, respectively.

*3)* ***Algorithm Comparison:*** From Figure 2.5 (a) to (c) and Figure 2.6 (a) to (c), we can also evaluate the performance difference between our proposed algorithm and the algorithms in [51] and [65] in the multi-user scenario. Notice that the solid bars in Figure 2.5 and Figure 2.6 are equivalent to the corresponding bars in Figure 2.4, as the performance under the non-cooperative scenario in the multi-user scenario is equivalent to the performance in the single-user scenario. By comparing the difference between solid bars (for the single-user scenario) and the difference between hollow bars (for the multi-user cooperative scenario), we can find that the performance difference (between our algorithm and the algorithms in [51, 65]) become more significant in the multi-user scenario, especially when the video user percentage is small. For example, with 20% video users, the average bitrate increase of our algorithm reaches 20% $\sim$ 30%, and the average social welfare increase of our algorithm reaches 10% $\sim$ 50%. Such an increasing in the performance difference (between our algorithm and those in [51, 65]) is mainly due to the non-optimal segment owner selection in [51, 65]. In our algorithm, however, the segment owner selection and the bitrate adaptation are optimised jointly.

## 2.8 Chapter Summary

In this chapter, we proposed a crowdsourced streaming framework for multi-user cooperative video streaming over mobile networks. We analyzed the theoretical performance bound of the proposed crowdsourced streaming system, and designed the associated online algorithm for the practical implementation. We conducted extensive experiments with real data traces, and illustrated both the cooperation gain of the proposed crowdsourced system and the performance gain of the proposed algorithm. Adaptive bitrate streaming is a new technology trend of mobile video streaming, and the research on multi-user cooperative video streaming is becoming increasingly important. This chapter developed a unified cooperative framework, for both theoretical analysis and practical implementation.

# Chapter 3

# Communication Sharing: Incentive Mechanism

## 3.1 Introduction

### 3.1.1 Background and Motivation

Mobile video traffic accounted for around 55% of global mobile traffic in 2015, and is expected to grow at an annual rate of 62% between 2015 and 2020 [10]. The increasing video demand requires proper resource allocation methods to achieve desirable user's quality of experience (QoE) in increasingly congested wireless networks with limited radio resources. A key challenge to achieve this is that different users can have very different QoE requirements (e.g., depending on device screen sizes and user preferences) and channel conditions (e.g., 3G cellular, 4G cellular, or WiFi links). To resolve and exploit the heterogeneity among users and deal with the potential mismatch of video requirements and channel conditions at the individual user level, we have proposed a *crowdsourced mobile streaming* (CMS) model in Chapter 2. This model enables mobile users to form cooperative groups and share their network resources for more effective mobile video streaming.

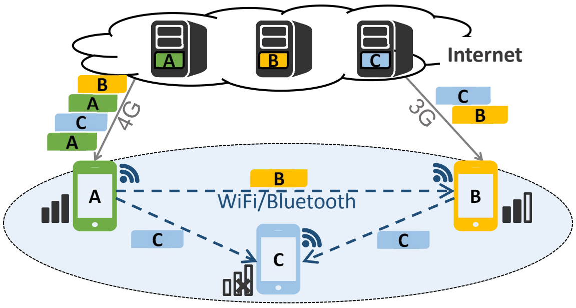


Figure 3.1: Crowdsourced mobile streaming.

The CMS model is very suitable for the adaptive bitrate video (ABR) streaming technology [14], a widely used video streaming technology in HTTP networks. In ABR, a video is partitioned into multiple video segments, and each video segment is encoded at multiple bitrates. A video user can choose the bitrate of each segment based on his preference and the real-time network condition. Hence, ABR-based video streaming provides a good amount of flexibility for cooperative downloading in the CMS system.

Figure 3.1 shows an example of the CMS model with three users, where each user watches a unique video hosted by the corresponding server. User C does not have a cellular connection to the Internet, so both user A and user B download user C's segments and forward to user C. User A also downloads a segment for user B, as he has a better downlink channel (4G) than user B (3G). In the CMS model, the downloading links from the Internet to users can be either cellular links or WiFi links, and the connections among users can be either WiFi Direct links or Bluetooth. In order to make the presentation easy to follow, we will refer to all the downloading links as "cellular links" and all the connections among users as "WiFi Direct links"; these are merely terminology choices that do not limit the applicability of the system.

The CMS model is different from the device-to-device (D2D) based [60, 64, 28] and peer-to-peer (P2P) based [92, 13, 62, 59] video streaming models, where users share their *downloaded* video segments with other users through D2D links and the Internet, respectively. In the CMS model, users share their cellular network resources (for segments downloading), hence it is mainly targeted at the much more common application scenario that different users watch *different* videos. Different from the bandwidth aggregation (BA) models that aggregate multiple users' bandwidth to serve one user's streaming need [99, 81, 98], the CMS model aggregates multiple users' bandwidth to satisfy all users' video streaming needs, enhancing the users' QoE through proper network resource allocation.

A major challenge for realizing the CMS model is that helping others will increase mobile users' cost, so the mobile users may not be willing to cooperate unless they receive proper incentives. In other words, the success of such a CMS model requires a proper *incentive mechanism* that motivates mobile users to crowdsource their network resources for cooperative video segments downloading.

### 3.1.2 Solution Approach and Contribution

In this chapter, we focus on the *incentive mechanism design* for the CMS model. Namely, we aim to design such mechanisms that offer enough compensation for each video user to download video segments for others, considering the user's own service request and downloading cost. The proposed mechanism needs to consider the following questions for each segment that each user (downloader) downloads:

- *Receiver Selection:* Whose segment will the downloader download?
- *Bitrate Adaption:* What bitrate (quality) will the receiver choose for the segment to be downloaded?
- *Cost Compensation:* How much will the downloader be compensated for

his downloading cost by the receiver?

It is challenging to design an effective incentive mechanism that addresses above questions, because of the users' private valuations for multi-bitrate encoded video segments as well as their asynchronous downloading behaviors. First, a user's valuation for a segment at a particular bitrate is the user's private information and can vary over time. The diverse and varying private valuation induces difficulties in evaluating users' contributions in cooperation and determining the proper incentive levels. Second, video scheduling in ABR streaming is segment based instead of time-slot based, so it is challenging to schedule the downloading cooperation among the users who request and download videos at different times.

Auction is widely used for allocating objects among the users who have private valuations. Hence, we propose auction-based incentive mechanisms for the CMS to handle the users' private valuation revelation. To address the asynchronous operations, we consider decentralized mechanisms: when a user (downloader) is ready to download new segments, he will initiate an auction to decide for whom to download at what bitrate with what price. In other words, the downloader acts as an auctioneer, and his nearby users (who request videos) act as bidders, bidding for the segment downloading opportunities.

Classical single-dimensional auction, where a bidder submits a single value indicating his willingness-to-pay, is not applicable in our crowdsourced model. This is because the video segments are encoded at multiple bitrates in ABR, so a bidder needs to specify multi-dimensional information in the bid, i.e., his intended bitrate and the price he is willing to pay for such a bitrate. This motivates us to consider a multi-dimensional auction in this chapter.

As a benchmark, we first propose a *single-object multi-dimensional auction* (SOMD) [31] based incentive mechanism framework for the CMS model,

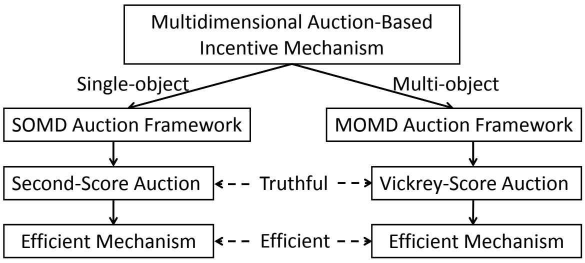


Figure 3.2: Theoretical framework of this chapter.

where an auctioneer allocates one segment in one auction. Based on the SOMD framework, we propose a second-score auction-based mechanism that ensures the truthful user valuation revelation in the CMS model. Through a proper design of the score function (to be discussed in Section 3.4.1), we derive the efficient mechanism that maximizes the social welfare.

However, such a single-object allocation may induce extensive signaling overhead because of the frequently initiated auctions, which may negatively affect the video streaming performance. This motivates us to consider a *multi-object multi-dimensional auction* that enables auctioneers to allocate multiple segments in one auction. Such a multi-object allocation introduces an additional dimension in the bidding process—the quantity (the number of the segments that a bidder desires), which is preferential dependent[1] of price. It has been shown in [23] that designing a multi-dimensional auction with preferential dependent dimensions is extremely difficult, but it turns out to be the problem that we need to solve. In this chapter, we propose a *multi-object multi-dimensional* (MOMD) auction framework, which enables bidders to bid for multiple objects (i.e., segments) with different bitrates in each auction. Within the MOMD framework, we design the allocation rule and payment rule, which leads to a truthful Vickrey-score auction. By a

[1] Dimension $x$ is preferentially dependent of dimension $y$ if the preference of $x$ depends on the preference of $y$ [71].

Table 3.1: Multi-user models in adaptive bitrate streaming

| Reference | Model | Model | | Method | | | Demo |
|---|---|---|---|---|---|---|---|
| | | Multi-Server | Multi-Video | Multi-Seg | Bitr-Adapt | Incent-Mech | |
| [60] | D2D | √ | × | √ | × | × | √ |
| [64, 28] | D2D | × | × | × | × | √ | × |
| [92, 62] | P2P | √ | × | √ | √ | × | √ |
| [13] | P2P | √ | × | √ | × | √ | × |
| [99, 98, 81] | BA | × | × | √ | × | √ | √ |
| Chapter 2 | CMS | √ | √ | × | √ | × | × |
| **This Chapter** | **CMS** | √ | √ | √ | √ | √ | √ |

proper design of the score function, we propose an efficient mechanism that maximizes the social welfare. Figure 3.2 illustrates the theoretical framework of this chapter.

The single-object and multi-object mechanisms assume that every user who is close to a downloader (and watches a video) will participate in the auction (when the downloader is ready to sell his downloading opportunities). Although the mechanisms maximize the social welfare in each auction (under a properly chosen score function), the long-term social welfare across multiple rounds of auction may not necessarily be maximized in some cases. For example, if a downloader's channel condition is very poor (at the time he initiates the auction), then it might be wise for the nearby users to refrain from bidding and wait for a different downloader (with a better channel condition) to become available. Therefore, we will further modify the proposed mechanisms, allowing users to refrain from bidding according to certain rules, which can improve the overall long-term system performance.

Our key contributions are summarized as follows:

- *Auction-Based Incentive Mechanisms in the CMS model:* We propose multi-dimensional auction based incentive mechanisms for the CMS model, supporting the asynchronous downloading and bitrate adapting of video users. The design of such mechanisms is challenging, as it needs to ensure that users truthfully report multi-dimensional preferentially dependent information.

- *Truthful and Efficient Auction*: For single-segment allocation, we propose a SOMD framework, based on which we propose a truthful and efficient mechanism that maximizes the social welfare in each auction. For multi-segment allocation, we propose an MOMD framework, based on which we propose the first mechanism achieving both truthfulness and efficiency in a multi-object multi-dimensional auction.

- *Modified Mechanism*: To enhance the long-term social welfare of video streaming services, we further improve the proposed mechanisms by allowing bidders to refrain from bidding according to their current situations. The simulation results show that such modification can successfully decrease rebuffer and bitrate degradation frequency along the entire video streaming.

- *Real-world Demonstration System*: We construct a demo system using Raspberry PI (a series of single-board computers) that enables the cooperation among multiple users watching multiple videos. Using the demo, we further analyze the real-world performances of the CMS.

- *Experiments and Performances*: Based on the modified auction mechanism, we perform experiments in both simulative system and demo system. Simulations with real traces show that crowdsourced mobile streaming outperforms noncooperative streaming by 48.6% (on average) in social welfare. Experiments over the demo system further show that those users who help others and those users who receive helps can increase their welfares by 15.5% and 35.4% (on average) via cooperation, respectively.

The rest of this chapter is as follows. We review related works in Section 3.2. We describe the system model in Section 3.3, and propose incentive mechanisms in Sections 3.4 and 3.5. We further propose a modified mechanism in 3.6. Then, in Section 3.7, we describe a demo system. In Section 3.8, we show experiment results. In Section 3.9, we conclude this chapter.

## 3.2 Related Work

### 3.2.1 Adaptive Bitrate Streaming

Most of early studies on ABR focused on single-user bitrate adaptation methods, such as buffer-based adaptation [82, 51], bandwidth-based adaptation [65], and hybrid buffer-bandwidth adaptation [100, 46, 93].

To better utilize the network resources, some recent works studied multi-user streaming models, which can be divided into four types [86]: *D2D models* [60, 64, 28], where users share their downloaded video segments with other users through D2D links; *P2P models* [92, 13, 62], where users download video segments from other users who have already downloaded it through the Internet; *BA models* [99, 98, 81], where multiple users aggregate their bandwidth to serve one user's video streaming need; *CMS model* [42], where multiple video users (who may watch different videos) form groups to share their cellular resources to serve all users' video streaming needs.

We summarize the key features of these works in Table I. Specifically, from the model's perspective, we compare two features: multi-server, "$\surd$" if videos can be downloaded from multiple servers (users with downloaded videos can also be regarded as servers); multi-video, "$\surd$" if users watch different videos. From the method's perspective, we compare three features: multi-seg, "$\surd$"

if multiple segments can be allocated in an allocation; bitr-adapt, "$\surd$" if bitrate adaptation is considered; incent-mech, "$\surd$" if incentive mechanism is considered. We also compare whether the studies involve real demonstration system or not.

In Chapter 2, we proposed a CMS model and derived the corresponding offline optimization problem. However, in practice, a properly designed incentive mechanism is always required to motivate user cooperation, as cooperations might lead to additional costs. This motivates the study of incentive mechanism in this chapter.

### 3.2.2 Multi-Dimensional Auction

A multi-dimensional auction enables bidders to reveal multi-dimensional information regarding the auctioned goods, such as price and quality. Che proposed a multi-dimensional auction framework [31], based on which Asker *et al.* in [16] and David *et al.* in [37] studied auction properties under specific score functions. As the multi-dimensional auction generalizes the single-dimensional auction, it has found wide applications in financial markets [45] and power procurement [30].

Most of the existing works on the multi-dimensional auction considered single-object allocation, where only one good is allocated in each auction. In [23], Bichler *et al.* showed that the multi-object extension in multi-dimensional auction is difficult because of the preferential dependence: bidders' preferences of the price depend on their preferences of the quantity. Specifically, with the preferential dependence, the widely used score function in the *additive* form (as in Definitions 1 and 3) fails to characterize the relationship between the price and the quality dimensions. If the score function is non-additive, the auction will be quite challenging to analyze. In [23], the authors proposed a continuous auction mechanism in the multi-object case, without the guarantee of either truthful bidding or efficient resource allocation. In addition, the continuous auction is unsuitable for video streaming applications, because such an auction incurs a large signaling overhead in practice as bidders have to submit bids repeatedly to reach an agreement.

In this chapter, instead of capturing all three dimensions (i.e., price, bitrate, and quantity) in the score function, we only capture price and bitrate dimensions using an additive form. We address the quantity dimension by enabling each bidder to submit a set of two-dimensional bids (bitrate and quality dimensions), each of which corresponds to the bid under a particular segment number (quantity dimension). This represents a new approach of the multi-object multi-dimensional auction design. As far as we know, this is the first work that achieves truthful bidding and efficient resource allocation in a multi-object multi-dimensional auction.

## 3.3 System Model

In a CMS model, we consider a set of mobile users $\mathcal{N} \triangleq \{1, 2, ..., N\}$, who download videos cooperatively. Each user watches a video that is encoded based on the ABR technology and is downloaded via cellular links to his mobile device.

### 3.3.1 Adaptive Bitrate Streaming

We consider a typical ABR streaming protocol [14] in the CMS model. Its key features are summarized as follows.

***Video Segmentation:*** A source video is partitioned into a sequence of small segments, each of which contains a piece of the source video with a fixed playback time (e.g., 10 seconds).

***Multi-Bitrate Encoding:*** A segment is encoded in multiple copies with different bitrates, so that a user can select the most suitable bitrate for each segment. Such a bitrate selection can be based on many factors, such as real time network conditions and individual preferences.

***Data Buffering:*** To smooth the playback, each downloaded segment is saved in a buffer at the user's device before playing. The video player on the user's device fetches segments from the buffer sequentially for playback. Due to the device's storage limit, the buffer has a limited maximum size.

For a user $n$, let $\beta_n > 0$ denote his video's fixed segment length (in terms of playback time), let $\mathcal{R}_n \triangleq \{R_n^1, R_n^2, ..., R_n^Z\}$ denote the corresponding finite bitrate set, and let $B_n > 0$ denote his maximum buffer size (in terms of playback time).

### 3.3.2 Crowdsourced Mobile Streaming

In the CMS model, users who are close-by form a mesh network and share their network resources. Through a proper scheduling mechanism, the group of users cooperatively download the requested segments of the entire group through cellular links and then forward segments to the actual requesting users (receivers) through WiFi Direct links. Different users can watch different videos in this framework.

We consider a continuous time model over a period of time $\mathcal{T} \triangleq [0,\ T]$, where $t = 0$ is the initial time and $t = T$ is the ending time. Let $h_n(t) > 0$ denote user $n$'s cellular link capacity at time $t \in \mathcal{T}$. Let $e_{n,m}(t) \in \{0, 1\}$ denote the encounter between users $n$ and user $m$ at time $t$, i.e., $e_{n,m}(t) = 1$ if user $n$ and user $m$ are encountered. Note that a user always encounters himself, i.e., $e_{n,n}(t) = 1$ for all $n, t$.

### 3.3.3 User Model

We first describe the welfare generated through the downloading operation between two users. Then, we define the social welfare of the system, which is the sum of the welfare generated by all downloading operations.

In the downloading operation between two users, a user $n \in \mathcal{N}$ downloads a sequence of a total of $\kappa$ segments with bitrates $\boldsymbol{r} = \{r_1, r_2, ..., r_\kappa\}$ for a user $m \in \mathcal{N}$, where $r_i > 0$ for all $i$. User $n$ and user $m$ can be the same user. The downloading of segment $i$ starts at $t_i^s$ and ends at $t_i^e$. The downloading timings and the channel condition satisfy the following relationship:

$$\int_{t_i^s}^{t_i^e} h_n(t)\mathrm{d}t = r_i \cdot \beta_m,\ i = 1, 2, ..., \kappa, \tag{3.1}$$

where the total downloaded volume within the downloading time is equal to the size of the downloaded segment.

This downloading operation (by user $n$ for user $m$) induces a cost for user $n$ and a utility for user $m$.

#### Cost of Downloader (User $n$)

Cost of the downloader is user $n$'s cost for downloading and transmitting video segments with bitrates $\boldsymbol{r} = \{r_1, r_2, ..., r_\kappa\}$. The cost function $C_{n,t}(\boldsymbol{r})$ consists of the cost on cellular links and the cost on WiFi Direct links: the cellular cost is the cost for downloading the segments (e.g., energy consumption or cellular data payment), while the WiFi Direct cost is the cost that user $n$ transmits the segments to user $m$ if $n \neq m$ (e.g., energy consumption). Let $c_{n,t}(r)$ be the cellular and the WiFi Direct cost for a single segment with bitrate $r$. We assume that the cost $c_{n,t}(r)$ is a non-decreasing linear function, i.e., $c_{n,t}(0) = 0$, $[c_{n,t}(r)]_r \geq 0$, and $[c_{n,t}(r)]_{rr} = 0$.[2] The linear model for

[2] Let $[\cdot]_x$ and $[\cdot]_{xx}$ denote the first and the second order derivatives with respect to variable $x$, respectively. Let $[\cdot]_{xy}$ denote the second-order partial derivative with respect to variable $x$ and variable $y$.

downloading and transmission energies has been widely considered in the existing works on video services and user-provided networks [47, 54], and the linear data payment is essentially a usage-based model commonly adopted by mobile network operators today. Relaxing the linearity assumption will not affect the auction mechanism and its corresponding properties, but will affect the particular form of the sufficient condition to satisify Assumption 3.1 (Proposition 3.5). We assume that the cost of different segments are independent of each other, so the cost $C_{n,t}(\boldsymbol{r})$ can be represented as follows:

$$C_{n,t}(\boldsymbol{r}) = \sum_{i=1}^{\kappa} c_{n,t}(r_i). \tag{3.2}$$

**Utility of Receiver (User $m$)**

Utility of the receiver is user $m$'s utility for receiving $\kappa$ video segments with bitrates $\boldsymbol{r} = \{r_1, r_2, ..., r_\kappa\}$. A user often desires to watch a high quality video without frequent video freezings (i.e., rebuffers) or quality degradations [46, 100, 93]. Hence, the utility depends on three factors: the video quality gain, the buffer filling gain, and the quality degradation loss. The buffer filling gain can be used to predict the rebuffering probability, since whether the exact rebuffering will occur or not is unknown when users make downloading decisions. Similar as in [43], we consider the quality degradation loss instead of the quality switching loss (that considers both the degradation and the upgrade losses), as humans are more sensitive to the degradation [66].

The utility function $U_{m,t}(\boldsymbol{r})$ is related to the receiver $m$'s desire for a high quality video $\theta_{m,t}$, his current buffer level $B^{\text{CUR}}_{m,t}$, and his previous segment bitrate $R^{\text{PRE}}_{m,t}$. Formally,

$$U_{m,t}(\boldsymbol{r}) \triangleq V^{\text{Q}}(\boldsymbol{r}, \theta_{m,t}) + V^{\text{B}}(\kappa, B^{\text{CUR}}_{m,t}) - L^{\text{QD}}(\boldsymbol{r}, R^{\text{PRE}}_{m,t}), \tag{3.3}$$

where the summation form of (3.3) is commonly used in existing ABR works, such as [93].

*a) Video Quality Gain* $V^{\mathrm{Q}}(\boldsymbol{r}, \theta_{m,t})$ is the user's gain in terms of the video segment quality. A user has a higher gain if he receives a segment with a higher bitrate. The user-dependent factor $\theta_{m,t}$ reflects user $m$'s desire for a high quality video. Let $v_m^{\mathrm{Q}}(r, \theta_{m,t})$ be the video quality gain function for a single segment with bitrate $r$, and this gain function is non-decreasing and concave [97], i.e., $[v_m^{\mathrm{Q}}(r, \theta_{m,t})]_r \geq 0$ and $[v_m^{\mathrm{Q}}(r, \theta_{m,t})]_{rr} \leq 0$. The quality gain is zero when the segment bitrate is zero (receives nothing), i.e., $v_m^{\mathrm{Q}}(0, \theta_{m,t}) = 0$, for any $\theta_{m,t}$. Moreover, a user with a higher $\theta_{m,t}$ has a higher desire to increase the bitrate, so $[v_m^{\mathrm{Q}}(r, \theta_{m,t})]_r$ is non-decreasing in $\theta_{m,t}$, i.e., $[v_m^{\mathrm{Q}}(r, \theta_{m,t})]_{r\theta} \geq 0$. Suppose that the quality gain of each segment is independent of that of the others, i.e.,

$$V_m^{\mathrm{Q}}(\boldsymbol{r}, \theta_{m,t}) = \sum_{i=1}^{\kappa} v_m^{\mathrm{Q}}(r_i, \theta_{m,t}). \tag{3.4}$$

*b) Buffer filling gain* $V^{\mathrm{B}}(\kappa, B_{m,t}^{\mathrm{CUR}})$ is the user's gain in terms of filling the playback buffer [82], which is a gain related to the segment number $\kappa$ only. A user will have a higher gain if he receives more segments in an allocation, as this leads to a reduced chance of video freezing. The gain of each additional segment decreases in the number of segments. This is because if a user has already been allocated a larger number of segments, he is less willing to obtain an additional one due to the reduced probability of rebuffering. For a user with a lower current buffer size, he is more willing to have new segments, so he will have a higher buffer filling gain for the allocated segments. Let $B_{m,t}^{\mathrm{CUR}}$ denote user $m$'s real-time buffer level at time $t$, which is measured in terms of the playback time. For notation convenience, we define a buffer filling gain gap between total $\kappa + 1$ segments and total $\kappa$ segments under buffer level $B_{m,t}^{\mathrm{CUR}}$, as $\Delta(\kappa, B_{m,t}^{\mathrm{CUR}}) = V_m^{\mathrm{B}}(\kappa + 1, B_{m,t}^{\mathrm{CUR}}) - V_m^{\mathrm{B}}(\kappa, B_{m,t}^{\mathrm{CUR}})$. Summarizing the above discussions, the function $V_m^{\mathrm{B}}(\kappa, B_{m,t}^{\mathrm{CUR}})$ satisfies the following inequalities:

$$[V_m^{\mathrm{B}}(\kappa, B_{m,t}^{\mathrm{CUR}})]_{B_{m,t}^{\mathrm{CUR}}} \leq 0, \tag{3.5}$$

$$\Delta(\kappa, B^{\text{CUR}}_{m,t}) \geq 0,\ \Delta(\kappa+1, B^{\text{CUR}}_{m,t}) - \Delta(\kappa, B^{\text{CUR}}_{m,t}) < 0. \tag{3.6}$$

*c) Quality Degradation Loss* $L^{\text{QD}}(\boldsymbol{r}, R^{\text{PRE}}_{m,t})$ is the user's loss when the video degrades from a higher bitrate to a lower bitrate. The user will have a higher degradation loss if the degradation gap is larger. Let $l^{\text{QD}}_m(\hat{r}, r)$ be the degradation loss due to the fact that a newly downloaded segment degrades from the previous bitrate $\hat{r}$ (of the previous segment) to the current bitrate $r$ with the gap $\Delta r = \hat{r} - r$:

$$l^{\text{QD}}_m(\hat{r}, r) = \begin{cases} 0, & \hat{r} < r, \\ \tilde{l}^{\text{QD}}_m(\Delta r), & \textit{otherwise}. \end{cases} \tag{3.7}$$

The positive part $\tilde{l}^{\text{QD}}_m(\Delta r)$ linearly increases with $\Delta r$ [93], i.e.,

$$[\tilde{l}^{\text{QD}}_m(\Delta r)]_{\Delta r} \geq 0,\ \ [\tilde{l}^{\text{QD}}_m(\Delta r)]_{\Delta r \Delta r} = 0. \tag{3.8}$$

Let $r_0 = R^{\text{PRE}}_{m,t}$ be the bitrate of the segment that user $m$ receives immediately before the new downloading segments. The loss $L^{\text{QD}}_m(\boldsymbol{r}, R^{\text{PRE}}_{m,t})$ of the segments with bitrates $\boldsymbol{r}$ is the sum of the degradation loss of all the segments. Formally,

$$L^{\text{QD}}_m(\boldsymbol{r}, R^{\text{PRE}}_{m,t}) = \sum_{i=1}^{\kappa} l^{\text{QD}}_m(r_{i-1}, r_i). \tag{3.9}$$

#### Social Welfare

In the downloading operation by user $n$ for user $m$, the generated welfare is defined as the difference between the user $m$'s utility and the user $n$'s cost:

$$W_{nm,t}(\boldsymbol{r}) = U_{m,t}(\boldsymbol{r}) - C_{n,t}(\boldsymbol{r}). \tag{3.10}$$

The social welfare of the system is the sum of the welfares that are generated through all the downloading operations among all the users.

### 3.3.4 Problem Formulation

We aim to design an incentive mechanism in the CMS model that can reveal user's private information and maximizes the social welfare. The mechanism should help each user to decide how to allocate the downloading opportunities of $K$ segments to near-by users: *(i) who is the receiver of each of the segments, (ii) what is the bitrate of each of the segments, and (iii) what is the payment of each of the segment receivers?*

We will design auction mechanisms to address the issue of user private information. More specifically, we propose a single-object ($K = 1$) and a multi-object ($K \geq 1$) auction-based mechanism in Section 3.4 and Section 3.5, respectively.

## 3.4 Single-Object Auction-Based Mechanism

We adopt an auction-based incentive mechanism, in which users allocate segment downloading opportunities using auctions. At each decision epoch of any user (who is ready to download segments), he acts as an auctioneer and initiates an auction for all nearby users for deciding the next $K$ segments ($K = 1$ in this section) to be downloaded. This framework operates in an asynchronous and decentralized manner, in the sense that each user initiates an auction independently and asynchronously from other users. To clarify, a double auction (that involves multiple auctioneers in an auction) is not suitable for the CMS model. This is because if implementing a double auction, under the auctioneers' asynchronous downloading operations, the auctioneers who are ready for downloading earlier have to wait for those who are ready later, which can lead to a significant downloading resource waste.

We propose a multi-dimensional auction based framework, in which the bidders reveal their *intended bitrates* and *intended prices* through submitting multi-dimensional bids on the $K$ segments to be downloaded. Without loss of generality, we consider an auction initiated by a downloader (auctioneer) $n$ to his encountered users. We assume that the auction period (including the auction operation period and the segment downloading period) is short enough, so that user $n$'s encountered users do not change during the auction period. Such an assumption is supported by the fact that a segment often has a small size (e.g., 10 seconds), and the corresponding total downloading and device-to-device transmission time is relatively small (comparing with the user's mobility time scale). Let $\mathcal{N}_n$ denote the set of user $n$'s encountered users:

$$\mathcal{N}_n \triangleq \{m \in \mathcal{N} \mid e_{n,m}(t) = 1, t \in [t_0, t_0 + \epsilon]\}, \tag{3.11}$$

where $[t_0, t_0 + \epsilon]$ is the auction period. Let $|\mathcal{N}_n|$ denote the total number of users in set $\mathcal{N}_n$. Note that the downloader will also join the auction as a *virtual* bidder (i.e., $e_{n,n}(t) = 1$) to fulfill his own service requirement. The bidder $m$'s private information is his real-time utility function, i.e., $U_{m,t}(\cdot)$, depending on his desire for high quality video $\theta_{m,t}$, current buffer level $B^{\text{CUR}}_{m,t}$, and previous segment bitrate $R^{\text{PRE}}_{m,t}$ (Section 3.3.3). We also assume that these functions and parameters do not change during a single auction.

In this section, we focus on single-object multi-dimensional auction mechanism design, where the auctioneer allocates one segment in each auction, i.e., $K = 1$.

### 3.4.1 Auction-Based Incentive Mechanism

#### SOMD Auction Framework

In the SOMD auction framework, each bidder submits a two-dimensional bid comprising *bitrate* and *price*. According to the bids, the auctioneer allocates the single segment to a single winner. Formally,

**Framework 3.1** (SOMD Auction Framework)**.**

1. *The auctioneer $n$ announces auction rules, including the* allocation rule $\Gamma(\cdot)$ *and the* payment rule $\Pi(\cdot)$*;*
2. *Each bidder $m \in \mathcal{N}_n$ submits a bid $\boldsymbol{b}^m = (r^m, p^m)$ to maximize his own expected payoff. Let $\boldsymbol{b} = (\boldsymbol{b}^m, \forall m \in \mathcal{N}_n)$ denote the bids from all the bidders;*
3. *The auctioneer $n$ determines the winner $\sigma^\dagger$ and the winner's payment $\pi^\dagger$ according to the announced rules:*

$$\sigma^\dagger = \Gamma(\boldsymbol{b}), \quad \pi^\dagger = \Pi(\boldsymbol{b}). \tag{3.12}$$

*The auctioneer will download a segment for winner $\sigma^\dagger$ at the bitrate specified in the winner's bid, i.e, $r^\dagger = r^{\sigma^\dagger}$.*

Here, $\pi^\dagger$ is the actual payment from the winner, which may not be equal to the price $p^{\sigma^\dagger}$ submitted by the winner. Given an auction outcome $(\sigma^\dagger, \pi^\dagger, r^\dagger)$, the auctioneer $n$'s payoff is

$$P_n(\pi^\dagger, r^\dagger) = \pi^\dagger - C_{n,t}(r^\dagger), \tag{3.13}$$

and the receiver (winner) $\sigma^\dagger$'s payoff is

$$P_{\sigma^\dagger}(\pi^\dagger, r^\dagger) = U_{\sigma^\dagger,t}(r^\dagger) - \pi^\dagger, \tag{3.14}$$

where $C_{n,t}(\cdot)$ is the downloader's cost defined in (3.2), and $U_{\sigma^\dagger,t}(\cdot)$ is the receiver's utility defined in (3.3).

### Second-Score Auction

The winning rule $\Gamma(\cdot)$ and the payment rule $\Pi(\cdot)$ are two key elements in auction design. In a single-dimensional auction, the auctioneer can determine the winner by simply sorting all bidders' prices and choosing the bidder with the highest price. In a multi-dimensional auction here, however, the auctioneer cannot determine the winner by simply choosing the bidder with the highest price. This is because the bitrate of bidder will affect the auctioneer's downloading cost, and hence the auctioneer's payoff.

To this end, we introduce a *score function* to determine the winner and the payment. The key idea is to transform a multi-dimensional bid $(r, p)$ into a single-dimensional score $\phi(r, p)$, so that the auctioneer can sort bidders according to their scores and determine the winner by choosing the highest score bidder. In this chapter, we adopt the class of score functions in [37].

**Definition 3.1** (Single-Object Score Function)**.** *Given the bitrate $r^m$ and the price $p^m$ submitted by a bidder $m$, the single-object score function is defined as*

$$\phi(r^m, p^m) = p^m - s(r^m), \tag{3.15}$$

*where $s(\cdot)$ is a non-decreasing function with $s(0) = 0$.*

Intuitively, such a score function increases with the bidder's price and decreases with the bidder's bitrate, capturing the fact that the auctioneer prefers a higher price and a lower bitrate. Note that (3.15) corresponds to a class of score functions, as we have not specified the concrete form of function $s(\cdot)$. A key contribution of our work is to design the function $s(\cdot)$ properly in order to achieve desirable outcomes such as efficiency (social welfare maximization).

We implement a *second-score* (multi-dimensional) auction [31], where the winner is the bidder with the highest score, and the winner's payment is the price that "derives" the second highest score under the winner's bitrate. Formally,

**Mechanism 3.1** (Second-Score Auction)**.** *The second-score auction is a special case of Framework 3.1, where the allocation and bidding rules are defined as follows:*

1. Allocation Rule*: The winner* $\sigma^\dagger$ *is the bidder with the highest score, i.e.,*

$$\sigma^\dagger = \arg\max_{m \in \mathcal{N}_n} \phi(r^m, p^m); \tag{3.16}$$

2. Payment Rule*: The winner's payment* $\pi^\dagger$ *is the price that derives the second highest score under his bitrate* $r^\dagger = r^{\sigma^\dagger}$*, i.e.,*

$$\pi^\dagger - s(r^\dagger) = \max_{m \in \mathcal{N}_n / \sigma^\dagger} \phi(r^m, p^m). \tag{3.17}$$

Next, we first analyze any bidder's optimal price and bitrate strategies in Section 3.4.2. Based on the optimal strategies, we propose an efficient mechanism through a proper choice of score function in Section 3.4.3.

### 3.4.2 Truthfulness and Optimal Bitrate

In the second-score auction, we will show that each bidder will submit his bid (i.e., price and bitrate) according to Proposition 3.1 and 3.2 to maximize his expected payoff, with proofs in Appendices 3.10.1 and 3.10.2, respectively.

**Proposition 3.1** (Truthfulness)**.** *Given any bitrate bidding strategy* $r^m$*, the optimal price bidding strategy* $p^m$ *of a bidder* $m$ *is his true utility under the selected bitrate* $r^m$*, i.e.,*

$$p^m = U_{m,t}(r^m). \tag{3.18}$$

**Proposition 3.2** (Optimal Bitrate)**.** *The optimal bitrate bidding strategy* $r^m$ *of a bidder* $m$ *is given by*

$$r^m = \arg\max_{r \in \mathcal{R}_m} \left( U_{m,t}(r) - s(r) \right). \tag{3.19}$$

Propositions 3.1 and 3.2 propose the optimal strategy of each bidder $m$ in the second-score auction. To maximize his own profit, each bidder should select the bitrate $r$ that maximizes the difference between his utility $U_{m,t}(r)$ and the $s(r)$, and select the price $p$ that is equal to his utility under the optimized bitrate. Due to the finite choices of bitrate, an optimal solution always exists, and each bidder is able to calculate the optimal solution based on his local information and the auctioneer's announced information with a low computation complexity.

### 3.4.3 Efficiency

Notice that Propositions 3.1 and 3.2 hold for any score functions in the form of (3.15). On the other hand, the specific choice of $s(r)$ determines bidders' optimal strategies and auction's allocation and payment, so auctioneers can choose the score function to achieve desirable auction outcomes.

Here, we propose the efficient mechanism that maximizes the social welfare. We first define an efficient score function:

**Definition 3.2** (Single-Object Efficient Score Function)**.** *An efficient score function is in the form of*

$$\phi(r, p) = p - C_{n,t}(r), \tag{3.20}$$

*where $C_{n,t}(r)$ is the auctioneer's downloading cost.*

Under the score function of (3.20), we next show that the second-score auction implements the efficient mechanism.

**Theorem 3.1** (Efficiency)**.** *Under the optimal bidding behavior specified in Propositions 3.1 and 3.2, the second-score auction with the efficient score function in* (3.20) *implements the efficient mechanism that maximizes the social welfare.*

*Proof.* Based on Proposition 3.1 and 3.2, each bidder $m$ submits bid $(r^m, p^m)$, where $r^m = \arg\max_{r \in \mathcal{R}_m} (U_{m,t}(r) - C_n(r))$ and $p^m = U_{m,t}(r^m)$. In other words, each bidder submits the bitrate $r^m$ that maximizes his score. This leads to

$$\phi(r^m, p^m) = \max_{r \in \mathcal{R}_m} \left(U_{m,t}(r) - C_n(r)\right). \tag{3.21}$$

In second-score auction, the winner $\sigma^\dagger$ is the bidder with the highest score, i.e.,

$$\sigma^\dagger = \arg\max_{m \in \mathcal{N}_n} \phi(r^m, p^m). \tag{3.22}$$

The winning bitrate $r^\dagger$ is the bitrate submitted by the winner $\sigma^\dagger$, i.e., $r^\dagger = r^{\sigma^\dagger} = \arg\max_{r \in \mathcal{R}_{\sigma^\dagger}} \big(U_{\sigma^\dagger,t}(r) - C_n(r)\big)$. Hence, the social welfare under $\sigma^\dagger$ and $r^\dagger$ is as follows:

$$U_{\sigma^\dagger,t}(r^\dagger) - C_n(r^\dagger) = \max_{m \in \mathcal{N}_n} \max_{r \in \mathcal{R}_m} \left(U_{m,t}(r) - C_n(r)\right), \tag{3.23}$$

which implies that the social welfare is maximized. □

Note that the exact downloading capacity is unknown beforehand, which leads to an unknown cost function $C_{n,t}(r)$ in (3.20) when an auctioneer initiates an auction. Hence, in practice, an auctioneer needs to estimate his downloading capacity based on his historical information using methods such as the one in [65]. The design and optimization of such an estimation is outside the scope of this chapter. In later simulations, we assume that an auctioneer calculates his cost function based on the average capacities of his previous several downloading operations. Although the estimation accuracy will affect the mechanism performance, bidders and auctioneers make decisions based on not only the cost $C_{n,t}(r)$ but also bidders' utilities $U_{m,t}(r)$ (which involves bidders' buffer level information). Hence, under the extreme case where capacities vary dramatically, the consideration of the buffer levels can alleviate the performance degradation caused by inaccurate estimation.

## 3.5 Multi-Object Auction-Based Mechanism

To reduce the possibly excessive signaling overhead caused by the frequently auctions, in this section, we consider the more general case of multi-object multi-dimensional auction mechanism design, where the auctioneer allocates multiple segments in each auction, i.e., $K \geq 1$. For nontation simplicity, we will write the bidder set $\mathcal{N}_n$ as $\mathcal{M} = \{1, 2, ..., M\}$, where $M$ is the total number of bidders in the set $\mathcal{N}_n$.

### 3.5.1 Auction-Based Incentive Mechanism

**MOMD Auction Framework**

In the MOMD auction framework, bidders submit multi-dimensional bids, revealing their intended *bitrate* and *price* under each segment that might be allocated. Based on the bids, the auctioneer allocates the (downloading opportunities of) $K$ segments to multiple bidders. An MOMD auction operates as follows:

**Framework 3.2.** *[MOMD Auction Framework]*

1. *The auctioneer n announces auction rules, including the* segment number $K$*, the* allocation rule $\Gamma(\cdot)$*, and the* payment rule $\Pi(\cdot)$*;*
2. *Each bidder $m \in \mathcal{M}$ submits a bid $\boldsymbol{b}^m = (\boldsymbol{R}^m, \boldsymbol{p}^m)$ to maximize his own expected payoff. Let $\boldsymbol{b} = (\boldsymbol{b}^m, \forall m \in \mathcal{M})$ denote the bids from all the bidders. Here,*
   - *Bitrate matrix*

$$\boldsymbol{R}^m = \begin{bmatrix} \boldsymbol{r}_1^m \\ \boldsymbol{r}_2^m \\ \vdots \\ \boldsymbol{r}_K^m \end{bmatrix} = \begin{bmatrix} r_{11}^m & 0 & ... & 0 \\ r_{21}^m & r_{22}^m & ... & 0 \\ \vdots & \vdots & \ddots & \vdots \\ r_{K1}^m & r_{K2}^m & ... & r_{KK}^m \end{bmatrix}, \tag{3.24}$$

*where $r^m_{\kappa i} \in \boldsymbol{R}^m$ is the bitrate of the $i^{th}$ segment when bidder $m$ is allocated a total of $\kappa$ segments.*

- *Price Vector*

$$\boldsymbol{p}^m = (p^m_1, p^m_2, ..., p^m_K), \tag{3.25}$$

*where $p^m_\kappa$ is the total price (willingness-to-pay) when bidder $m$ is allocated a total of $\kappa$ segments.*

3. *The auctioneer $n$ determines the* allocation set*, i.e., the winner of* each segment*, $\boldsymbol{\sigma}^\dagger \triangleq \{\sigma^\dagger_1, \sigma^\dagger_2, ..., \sigma^\dagger_K\}$, and the* payment set*, i.e., the price that* each bidder *needs to pay, $\boldsymbol{\pi}^\dagger \triangleq \{\pi^\dagger_1, \pi^\dagger_2, ..., \pi^\dagger_M\}$, according to the rules:*

$$\boldsymbol{\sigma}^\dagger = \Gamma(\boldsymbol{b}), \quad \boldsymbol{\pi}^\dagger = \Pi(\boldsymbol{b}). \tag{3.26}$$

*Accordingly, the downloading bitrate of each segment is equal to the submitted bitrate of the corresponding winner, denoted by $\boldsymbol{r}^\dagger \triangleq \{r^\dagger_1, r^\dagger_2, ..., r^\dagger_K\}$.*

Notice that both the allocation set and the bitrate set have the size of $K$, as these two sets enumerate the receiver and the bitrate for *each segment*, respectively; however, the size of the payment set is $M$, and each element corresponds to the payment from *a bidder*. To facilitate the later discussions, we define a revised allocation set $\boldsymbol{\sigma}^\ddagger$ and a revised bitrate set $\boldsymbol{r}^\ddagger$, both of which have the size of $M$. More specifically, starting from allocation set $\boldsymbol{\sigma}^\dagger$, we can compute the number of segments allocated to bidder $m$, denoted as $\widetilde{\kappa}_m$. With this we can define the revised allocation set as $\boldsymbol{\sigma}^\ddagger = \{\widetilde{\kappa}_1, \widetilde{\kappa}_2, ..., \widetilde{\kappa}_M\}$, where $\sum_{m=1}^M \widetilde{\kappa}_m = K$. Similarly, we define the revised bitrate set as $\boldsymbol{r}^\ddagger = \{\widetilde{\boldsymbol{r}}_1, \widetilde{\boldsymbol{r}}_2, ..., \widetilde{\boldsymbol{r}}_M\}$, where vector $\widetilde{\boldsymbol{r}}_m$ is the bitrate set for the $\widetilde{\kappa}_m$ segments allocated to bidder $m$, i.e., $\widetilde{\boldsymbol{r}}_m = \boldsymbol{r}^m_{\widetilde{\kappa}_m}$ (i.e., the $\widetilde{\kappa}_m$th row of bitrate bid matrix $\boldsymbol{R}^m$).

Based on the auction results, the auctioneer $n$'s payoff is the sum of the difference between each bidder's payment and $n$'s downloading cost for this bidder's segments, i.e.,

$$P_n(\boldsymbol{\pi}^{\dagger}, \boldsymbol{r}^{\ddagger}) = \sum_{m=1}^{M} [\pi_m^{\dagger} - C_{n,t}(\widetilde{\boldsymbol{r}}_m)]. \tag{3.27}$$

Bidder $m$'s payoff is the difference between his utility and his payment, i.e.,

$$P_m(\pi_m^{\dagger}, \widetilde{\boldsymbol{r}}_m) = U_{m,t}(\widetilde{\boldsymbol{r}}_m) - \pi_m^{\dagger}. \tag{3.28}$$

#### Vickrey-Score Auction

In a multi-dimensional auction, the vector bids may not be sorted easily, and this introduces difficulties for determining the allocation set and the payment set. We again introduce a *score function* to address this problem. Different from single-object case in Section 3.4, here we will transform the bids into sequences of *marginal scores*, of which the auctioneer can sort and make decisions.

We first define the score function as follows.

**Definition 3.3** (Multi-Object Score Function)**.** *Given the bitrate $\boldsymbol{R}^m$ and the price $\boldsymbol{p}^m$ submitted by a bidder $m$, for any number of allocated segments $\kappa$, the multi-object score function $\phi(\boldsymbol{r}_\kappa^m, p_\kappa^m)$ is given by*

$$\phi(\boldsymbol{r}_\kappa^m, p_\kappa^m) = p_\kappa^m - s(\boldsymbol{r}_\kappa^m), \tag{3.29}$$

*where $s(\cdot)$ is a component-wise non-decreasing function and $s(\mathbf{0}) = 0$.*

The score function in (3.29) involves one row in the bitrate matrix in (3.24) and one component in the price vector in (3.25). Hence, for each bidder $m$, we will compute $K$ scores, i.e., $\phi(\boldsymbol{r}_\kappa, p_\kappa), \forall \kappa = 1, ..., K$. Based on this, we can further compute the *marginal score sequence* for each bidder $m$: $\boldsymbol{S}^m = \{S_1^m, S_2^m, ...S_K^m\}$, where the $\kappa^{th}$ marginal score reflects bidder $m$'s score increase when the total allocated segment number to bidder $m$ increases from

$\kappa - 1$ to $\kappa$. Formally,

$$S_{\kappa}^{m} = \begin{cases} \phi(\boldsymbol{r}_1^m, p_1^m), & \kappa = 1, \\ \phi(\boldsymbol{r}_{\kappa}^m, p_{\kappa}^m) - \phi(\boldsymbol{r}_{\kappa-1}^m, p_{\kappa-1}^m), & 2 \leq \kappa \leq K. \end{cases} \tag{3.30}$$

We impose the following assumption on marginal scores:

**Assumption 3.1** (Marginal Score)**.** *For any bidder $m \in \mathcal{M}$, the marginal score sequence $\boldsymbol{S}^m$ is non-negative and non-increasing in $\kappa$, where:*

$$S_{\kappa}^{m} \geq S_{\kappa+1}^{m} \geq 0, \quad \kappa = 1, 2, ..., K-1. \tag{3.31}$$

Assumption 3.1 implies that an additional segment induces a larger score (i.e., a positive marginal score $S_{\kappa+1}^m \geq 0$), and the score increase (i.e., the marginal score) is non-increasing with the allocated segment number $\kappa$ (i.e., $S_{\kappa}^m \geq S_{\kappa+1}^m$). In Section 3.5.4, we provide a sufficient condition under which Assumption 3.1 is always satisfied.

Inspired by the VCG mechanism [90], we propose a Vickrey-score auction, where we allocate the $K$ segments to the $K$ highest marginal scores, and choose the payments reflecting the score damages of the winners to the system. Next we will first define the proposed mechanism, and then provide a numerical illustrating example.

For a bidder $m$, let sequence $\hat{\boldsymbol{S}}^{-m}$ denote the $K$ highest marginal scores *except* bidder $m$'s:

$$\hat{\boldsymbol{S}}^{-m} \triangleq \{\hat{S}_1^{-m}, \hat{S}_2^{-m}, ..., \hat{S}_K^{-m}\}, \tag{3.32}$$

where $\hat{S}_k^{-m}$ is the $k^{th}$ highest value among all the bidders' marginal scores *except* bidder $m$'s. We further let $\boldsymbol{S}^{\dagger}$ denote the $K$ highest marginal scores among all bidders:

$$\boldsymbol{S}^{\dagger} \triangleq \{S_1^{\dagger}, S_2^{\dagger}, ..., S_K^{\dagger}\}, \tag{3.33}$$

where $S_k^{\dagger}$ is the $k^{th}$ highest value among all the bidders' marginal scores. The Vickrey-score auction is as follows:

**Mechanism 3.2** (Vickrey-Score Auction). *The Vickrey-score auction is a special case of Framework 3.2, where the allocation and payment rules are defined as follows:*

- *Allocation Rule: The segment k's receiver $\sigma_k^\dagger$ is the bidder corresponding to the $k^{th}$ highest marginal score, i.e.,*

$$S_i^{\sigma_k^\dagger} = S_k^\dagger, \tag{3.34}$$

*where $S_i^{\sigma_k^\dagger}$ refers to the $i^{th}$ marginal score of bidder $\sigma_k^\dagger$.*

- *Payment Rule: If bidder $m$ wins $\widetilde{\kappa}_m$ segments, then his payment $\pi_m^\dagger$ corresponds to the score damage caused by this bidder under his submitted bitrate, i.e.,*

$$\pi_m^\dagger - s(\boldsymbol{r}_{\widetilde{\kappa}_m}^m) = \sum_{i=1}^{\widetilde{\kappa}_m} \hat{S}_{K-\widetilde{\kappa}_m+i}^{-m}. \tag{3.35}$$

**Example 3.1.** *Consider an auction with $M = 3$ users and $K = 4$ segments, where we have the following marginal score sequences: $\boldsymbol{S}^1 = \{8,\ 7,\ 5,\ 2\}$, $\boldsymbol{S}^2 = \{9,\ 6,\ 3,\ 2\}$, and $\boldsymbol{S}^3 = \{4,\ 4,\ 3,\ 1\}$. Hence, we have the sorted sequences:*

$$\boldsymbol{S}^\dagger = \{9,\ 8,\ 7,\ 6\};\ \hat{\boldsymbol{S}}^{-1} = \{9,\ 6,\ 4,\ 4\};$$
$$\hat{\boldsymbol{S}}^{-2} = \{8,\ 7,\ 5,\ 4\};\ \hat{\boldsymbol{S}}^{-3} = \{9,\ 8,\ 7,\ 6\}.$$

*The four numbers in vector $\boldsymbol{S}^\dagger$ corresponds to the marginal scores of user 1 ($8$ and $7$) and user 2 ($9$ and $6$). Hence, according to the proposed Vickrey-score auction: user $1$ wins two segments, and user $2$ wins two segments. The payments of user 1 and user 2 are:*

$$\pi_1^\dagger = \sum_{i=1}^{2} \hat{S}_{4-2+i}^{-1} + s(\boldsymbol{r}_2^1) = \underbrace{4+4}_{score\ damage} + s(\boldsymbol{r}_2^1);$$
$$\pi_2^\dagger = \sum_{i=1}^{2} \hat{S}_{4-2+i}^{-2} + s(\boldsymbol{r}_2^2) = \underbrace{5+4}_{score\ damage} + s(\boldsymbol{r}_2^2).$$

*Take user* 1 *as an example: without user* 1*, user* 3 *will win* 2 *segments with scores* 4 *and* 4*, so these scores are the score damage caused by user* 1*. Hence, user* 1 *has to pay the price that compensates this damage as shown above.*

### 3.5.2 Truthfulness and Optimal Bitrate

In the Vickrey-score auction, we prove that each bidder will submit his bid (i.e., price and bitrate) according to Proposition 3.3 and Proposition 3.4 to maximize his expected payoff, with proofs in Appendices 3.10.3 and 3.10.4, respectively.

**Proposition 3.3** (Truthfulness)**.** *Given any bitrate matrix $\boldsymbol{R}^m$, the optimal price vector $\boldsymbol{p}^m$ of a bidder $m$ is his true utility under the selected bitrate matrix $\boldsymbol{R}^m$, i.e.,*

$$p^m_\kappa = U_{m,t}(\boldsymbol{r}^m_\kappa), \quad \kappa = 1, 2, ..., K. \tag{3.36}$$

**Proposition 3.4** (Optimal Bitrate)**.** *For any number of allocated segments $\kappa$ to bidder $m$, the optimal bitrate vector $\boldsymbol{r}^m_\kappa$ is the optimal solution $\boldsymbol{r}^\star$ of the following optimization problem:*

$$\begin{aligned} \underset{\boldsymbol{r}}{\text{maximize}} \quad & U_{m,t}(\boldsymbol{r}) - s(\boldsymbol{r}) \\ \text{subject to} \quad & r_i > 0, \quad i = 1, ..., \kappa, \\ & r_i = 0, \quad i = \kappa + 1, ..., K, \\ \text{variables} \quad & r_i \in \mathcal{R}_m, \quad i = 1, ..., \kappa. \end{aligned} \tag{3.37}$$

*The constraints restrict the allocated segment number to be $\kappa$.*

### 3.5.3 Efficiency

In this section, we propose the efficient score function that maximizes the social welfare.

**Definition 3.4** (Multi-Object Efficient Score Function). *An efficient score function is in the form of*

$$\phi(\boldsymbol{r}, p) = p - C_{n,t}(\boldsymbol{r}), \tag{3.38}$$

*where $C_{n,t}(\boldsymbol{r})$ is the downloading cost of the auctioneer.*

If each bidder submits the bid based on the optimal price in Proposition 3.3 and the optimal bitrate in Proposition 3.4, we prove that the Vickrey-score auction with the efficient score function maximizes the social welfare.

**Theorem 3.2** (Efficiency). *Under the optimal bidding behavior specified in Propositions 3.3 and 3.4, the Vickrey-score auction with the efficient score function in* (3.38) *implements the efficient mechanism that maximizes the social welfare.*

*Proof.* In the Vickrey-score auction with an efficient score function, when bidding according to Proposition 3.3 and 3.4, any bidder $m$'s bid will induce a score $\phi_{\kappa}^{m,n}$ for being allocated $\kappa$ segments, i.e., $\phi_{\kappa}^{m,n} = \max_{\boldsymbol{r}_\kappa} \left(U_{m,t}(\boldsymbol{r}_\kappa) - C_{n,t}(\boldsymbol{r}_\kappa)\right)$, where $\boldsymbol{r}_\kappa$ denotes the bitrate vector that satisfies the constraint of $\kappa$ segments. Here, $\phi_{\kappa}^{m,n}$ is essentially the maximum welfare that can be generated through the downloading by auctioneer $n$ for bidder $m$ under a particular segment number $\kappa$. Let $\boldsymbol{\sigma} = \{\kappa_1, \kappa_2, ..., \kappa_M\}$ denote an allocation set, where $\kappa_m$ is the number of segments allocated to bidder $m$. In the Vickrey-score auction, the auctioneer chooses the allocation set $\boldsymbol{\sigma}^* = \arg\max_{\boldsymbol{\sigma}} \sum_{m=1}^{M} \phi_{\kappa_m}^{m}$, i.e., picking the set of allocation that maximizes the welfare generated between auctioneer $n$ and bidders, that is, the social welfare. □

Finally, we comment on the applicability of the proposed Vickrey-score auction in existing video streaming systems, where the bitrate adaptation method has been specified. In this case, if each bidder chooses the bidding price according to Proposition 3.3 and use an existing bitrate adaptation method (e.g., [82, 51, 65, 100, 46, 93]), the Vickrey-score auction with the efficient score function is conditionally efficient.

**Corollary 3.1** (Conditional efficiency)**.** *Given any fixed bitrate $\boldsymbol{R}^m$ for bidder $m$, Vickrey-score auction with the efficient score function maximizes the social welfare under the fixed bitrates.*

The proof of Corollary 3.1 is similar as the proof of Theorem 3.2 and hence is omitted.

### 3.5.4 Conditions for Satisfying Assumption 3.1

By now we have proved several desirable properties of the Vickrey-score auction under Assumption 3.1. In this section, we will specify sufficient conditions, under which Assumption 3.1 is satisfied. As an example, we will focus on the efficient score function in (3.38) in the rest of the discussions. Our discussions can also be generalized to other choices of score functions.

The rest of this subsection is divided into two parts. First, we prove some additional properties of a bidder's optimal bitrate matrix. Next, we characterize sufficient conditions of the cost function $C_{n,t}(\cdot)$ and the utility function $U_{m,t}(\cdot)$ (defined in 3.3.3) in Proposition 3.5, under which Assumption 3.1 is satisfied.

Starting from Proposition 3.4, we prove that a bidder's optimal bitrate matrix has two features, as shown in Lemma 3.1 and 3.2. Note that both lemmas are based on the efficient score function in (3.38), where the optimal bitrate vector in each row $\kappa$ is given:

$$\boldsymbol{r}_{\kappa}^{m} = \arg\max_{\boldsymbol{r}_{\kappa}} \left( U_{m,t}(\boldsymbol{r}_{\kappa}) - C_{n,t}(\boldsymbol{r}_{\kappa}) \right). \tag{3.39}$$

Here, $\boldsymbol{r}_{\kappa} = \{r_{\kappa 1}, r_{\kappa 2}, ..., r_{\kappa\kappa}\}$ denotes the vector with $\kappa$ non-zero elements, and $U_{m,t}(\boldsymbol{r}_{\kappa}) = V_{m,t}^{\mathrm{Q}}(\boldsymbol{r}_{\kappa}) + V_{m,t}^{\mathrm{B}}(\kappa) - L_{m,t}^{\mathrm{QD}}(\boldsymbol{r}_{\kappa}, R_{m,t}^{\mathrm{PRE}})$. For presentation convenience, we define a function $g_{mn,t}(r) = v_{m,t}^{\mathrm{Q}}(r) - c_{n,t}(r)$. Since the value of $V^{\mathrm{B}}_{m,t}(\kappa, B^{\mathrm{CUR}}_{m,t})$ depends on segment number $\kappa$ but not the value of $\boldsymbol{r}_\kappa$, the optimal vector $\boldsymbol{r}_\kappa$ can also be represented as:

$$\boldsymbol{r}^m_\kappa = \arg\max_{\boldsymbol{r}_\kappa} \left( \sum_{i=1}^{\kappa} g_{mn,t}(r_{\kappa i}) - L^{\mathrm{QD}}_{m,t}(\boldsymbol{r}_\kappa, R^{\mathrm{PRE}}_{m,t}) \right). \tag{3.40}$$

Any bidder's optimal bitrate matrix has the following features:

**Lemma 3.1** (Identical Bitrate)**.** *Under the efficient score function in* (3.38)*, any bidder m's optimal bitrate matrix $\boldsymbol{R}^m$ satisfies that in any row $\kappa$, the non-zero bitrate elements $r^m_{\kappa i}$ $(i \le \kappa)$ are identical, hence can be written as $r^m_{\kappa i} = r^m_\kappa$, $\forall i \le \kappa$.*

The detailed proof is given in Appendix 3.10.5, and here are the intuitions. First, in (3.40), the order of the non-zero elements in vector $\boldsymbol{r}^m_\kappa$ only affects function $L^{\mathrm{QD}}_{m,t}(\boldsymbol{r}, R^{\mathrm{PRE}}_{m,t})$, which is minimized when the elements are in the ascending order. Hence, the non-zero elements in the optimal vector $\boldsymbol{r}^m_\kappa$ has be in the ascending order. This means that the bitrate degradation may only happen at the first segment, i.e.,

$$\boldsymbol{r}^m_\kappa = \arg\max_{\boldsymbol{r}_\kappa} \left( \sum_{i=1}^{\kappa} g_{mn,t}(r_{\kappa i}) - l^{\mathrm{QD}}(R^{\mathrm{PRE}}_{m,t}, r_{\kappa 1}) \right). \tag{3.41}$$

Second, there always exists an optimal bitrate (denoted by $r^*$) that maximizes $g_{mn,t}(r)$. If $R^{\mathrm{PRE}}_{m,t} \le r^*$, then $r^m_{\kappa i} = r^*$ for all $i = 1, 2, ..., \kappa$. If $R^{\mathrm{PRE}}_{m,t} \ge r^*$, we can obtain $r^m_{\kappa 1} \ge r^*$ by checking the partial derivate of the objective function in (3.41) with respect to $r_{\kappa 1}$. Moreover, the concave function $g_{mn,t}(r)$ is non-increasing with $r$ for $r \ge r^m_{\kappa 1} \ge r^*$, so $g_{mn,t}(r^m_{\kappa 2}), ..., g_{mn,t}(r^m_{\kappa\kappa})$ are maximized when bitrates $r_{\kappa 2}, ..., r_{\kappa\kappa}$ are minimized under the constraint that $r^m_{\kappa 1} \le r^m_{\kappa 2} \le ... \le r^m_{\kappa\kappa}$, which implies $r^m_{\kappa 1} = r^m_{\kappa 2} = ... = r^m_{\kappa\kappa}$.

**Lemma 3.2** (Non-Increasing Bitrate)**.** *Under the efficient score function in* (3.38)*, any bidder m's optimal bitrate matrix $\boldsymbol{R}^m$ satisfies that the bitrate $r^m_\kappa$ for row $\kappa$ defined in Lemma 3.1 is non-increasing in the row index $\kappa$: $r^m_\kappa \ge r^m_{\kappa+1}$ for all $\kappa = 1, 2, ..., K-1$.*

According to Lemma 3.1, the optimal common non-zero bitrate $r_\kappa^m$ for each row $\kappa$ is derived as follows:

$$r_\kappa^m = \arg\max_r \left(\kappa \cdot g_{mn,t}(r) - l^{\text{QD}}(R_{m,t}^{\text{PRE}}, r)\right). \tag{3.42}$$

Intuitively, if $R_{m,t}^{\text{PRE}} \leq r^*$, $r_i^m = r^*$ for all $i = 1, 2, ..., \kappa$. If $R_{m,t}^{\text{PRE}} \geq r^*$, as $\kappa$ increases, the impact of $\kappa \cdot g_{mn,t}(r)$ on the optimization problem increases, so $r_\kappa^m$ gradually decreases in $\kappa$ to approach $r^*$. The detailed proof is in Appendix 3.10.6.

Based on Lemma 3.1 and 3.2, we show the sufficient conditions of $C_{n,t}(\cdot)$ and $U_{m,t}(\cdot)$ for satisfying Assumption 3.1.

**Proposition 3.5** (Sufficient Conditions for Assumption 1)**.** *The marginal scores are non-negative for all $m, n, t$, if*

$$v_m^{\text{Q}}(r, \theta) \geq c_{n,t}(r), \;\forall r, \theta. \tag{3.43}$$

*The marginal scores are non-increasing in $\kappa$ (i.e., the number of allocated segments) for all $m, n, t$, if*

$$2K \cdot c_{n,t}(R_m^Z) + l_m^{\text{QD}}(R_m^Z, 0) \leq |\tilde{\Delta}|, \tag{3.44}$$

*where $\tilde{\Delta}$ is the minimum value that satisfies*

$$0 > \tilde{\Delta} \geq \Delta(\kappa + 1, B_{m,t}^{\text{CUR}}) - \Delta(\kappa, B_{m,t}^{\text{CUR}}), \;\forall \kappa, B_{m,t}^{\text{CUR}}. \tag{3.45}$$

Intuitively, to satisfy Assumption 3.1, the video quality gain of each allocated segment should be no less than the downloading cost of that segment to ensure the non-negative marginal scores, and the buffer filling gain should be concave enough (i.e., $|\tilde{\Delta}|$ should be large enough) to ensure the non-increasing marginal scores. The proof is shown in Appendix 3.10.7.

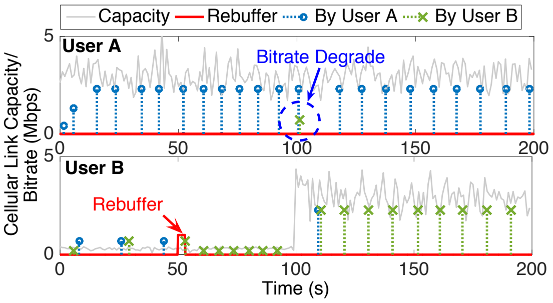


Figure 3.3: The video streaming of users A and B under mechanism 3.2.

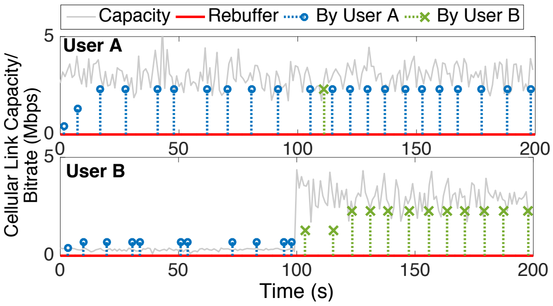


Figure 3.4: The video streaming of users A and B under mechanism 3.3.

## 3.6 Mechanism Modification

In Sections 3.4 and 3.5, we proposed two auction-based incentive mechanisms for single segment and multiple segments downloading, respectively. By implementing the efficient score function, the mechanisms can maximize the social welfare in each auction. However, since the social welfare maximization is performed in each auction independently, the long-term social welfare across multiple rounds of auctions may not necessarily achieve the maximum in some cases.

One scenario worth considering is where the link capacities of some users are substantially poorer than others. Hence utilizing the downloading opportunities of these users might actually hurt the overall performance. Figure 3.3 shows the video scheduling processes of such a scenario with two users: user

A and user B. Here $x$-axis corresponds to the video streaming time horizon (of 200 seconds), and $y$-axis corresponds to cellular network capacity (for the gray continuous curves). User A has an average capacity of 3Mpbs along the whole streaming interval (200 seconds), while user B has an average capacity of 0.3Mbps during the first 100 seconds and an average capacity of 3Mbps during the latter 100 seconds. The stems with circles and crosses are the segments that are downloaded by user A and user B, respectively, and the heights of these stems represent the corresponding segment bitrates. Note that the cellular link capacities and the bitrates are measured in the same unit of Mbps. Available bitrate set is $\{0.2, 0.4, 0.7, 1.3, 2.3\}$Mbps.

With the proposed auction mechanism, as shown in Figure 3.3, two unexpected results happen due to the low capacity of user B during the first 100 seconds: i) bitrate degradation; ii) rebuffer. For example, although user A achieves a video bitrate of 2.3Mbps most of the time, a bitrate degradation to 0.7Mbps happens at about second 100 when user B downloads for user A. The reason is that user B has a quite low link capacity, so users A chooses a lower bitrate (when asking user B to help downloading) to avoid rebuffer. Similar situation happens when user B downloads for himself at about second 60. Moreover, as user B partially relies on the downloading by himself during the first 100 seconds, he experiences rebuffer at second 50. The rebuffer continues until the corresponding segment has been downloaded at the end of second 53.

The observation in Figure 3.3 motivates us to modify our proposed mechanism to increase the long-term social welfare by avoiding unexpected bitrate degradation and rebuffer. The basic idea is that any bidder $m$ can “skip” the available network downloading resources from an auctioneer $n$ by refraining from bidding if both of the following conditions are satisfied: (i) the link capacity of auctioneer $n$ is low so that the downloading (by auctioneer $n$) for user $m$ will result in rebuffer; (ii) the link capacity of auctioneer $n$ is lower than the downloading capacity that allocates to user $m$, which is the sum of the capacities that each of user $m$'s encountered users allocates to user $m$ (under the assumption that user $m$'s encountered user $i \in \mathcal{N}_m$ equally allocates his capacity to his encountered users $\mathcal{N}_i$). Mathematically, We introduce coefficients $\alpha^{\text{LINK}}$ and $\alpha^{\text{BUF}}$ to adjust bidder's willingness of refraining from bidding: a smaller coefficient indicates a smaller willingness to skip the current resources.

**Mechanism 3.3** (Modification of Bidding Participation in Mechanism 3.2)**.** *To improve the long-term social welfare, we modify Mechanism 3.2 by allowing bidders to refrain from bidding if necessary. Specifically, after an auctioneer $n$ announces the start of the auction with the allocation and payment rules, a bidders should refrain from bidding if both of the following inequalities are satisfied:*

$$h_n(t) < \alpha^{\text{BUF}} \cdot \frac{R^{\text{PRE}}_{m,t} \cdot \beta_m}{B^{\text{CUR}}_{m,t}}, h_n(t) < \alpha^{\text{LINK}} \cdot \sum_{i \in \mathcal{N}_m} \frac{h_i(t)}{|\mathcal{N}_i|}, \tag{3.46}$$

*where $|\mathcal{N}_i|$ denotes the total number of user $i$'s encountered users. This means that only a subset of set $\mathcal{M}$ may choose to partipate in the bidding process. The rest of the auction is the same as Mechanism 3.2.*

Notice that the values of the coefficients $\alpha^{\text{BUF}}$ and $\alpha^{\text{LINK}}$ will impact on the social welfare, hence should be chosen carefully through experimental studies. Under the experiment setting similar as that in Figure 3.3, we evaluate each of the coefficient pairs $\alpha^{\text{LINK}} \in [0, 2]$ and $\alpha^{\text{LINK}} \in [0, 2]$ for 1000 randomly generated link capacity scenarios, and find that choosing $\alpha^{\text{LINK}} = 0.5$ and $\alpha^{\text{BUF}} = 1$ will lead to the largest long-term social welfare on average in this experiment. Hence, in our later experiments, we set $\alpha^{\text{LINK}} = 0.5$ and $\alpha^{\text{BUF}} = 1$.

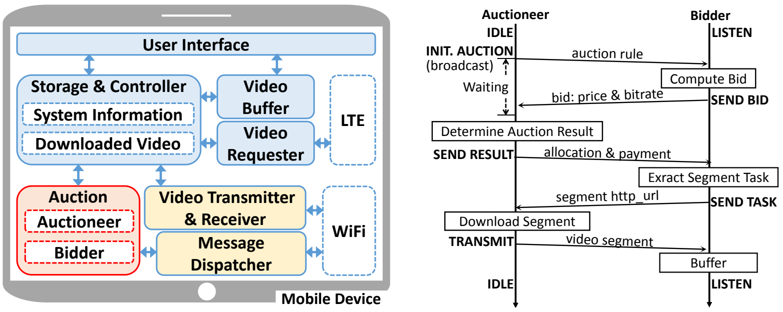
After the modification, Figure 3.4 shows the performance of the same two users (as in Figure 3.3) under the modified Mechanism 3.3, and we notice that

Table 3.2: Comparison between unmodified and modified mechanisms.

| User B's Average Capacity (Mbps) | 0.15 | 0.3 | 0.45 | 1.5 | 3.0 |
|---|---|---|---|---|---|
| Social Welfare Improvement (%) | 16.9 | 13.2 | 9.6 | 0.0 | 0.0 |
| Rebuffer Reduction (%) | 1.6 | 0.7 | 0.9 | 0.0 | 0.0 |
| Bitrate Degrade Reduction (%) | 22.1 | 5.9 | 0.2 | 0.0 | 0.0 |

the quality degradation and rebuffer do not occur (under the same experiment settings). Moreover, the modification does not have much impact on the scheduling when both users have relatively high average capacities (i.e., the last 100 seconds). Overall, the modification increases the long-term average social welfare by 6.17%.

We further perform comparisons between the *unmodified* Mechanism 3.2 and the *modified* Mechanism 3.3 over 1000 randomly generated network scenarios. In the experiments, user A and user B watch two different 100-second videos. The average link capacity of user A is 3Mbps, while the average capacity of user B varies from 0.15Mbps to 3Mbps (listed in Table 3.2). The rest of the settings are the same as in Figure 3.3 and 3.4. Table 3.2 shows the average results over the 1000 experiment rounds. As shown in the table, when user B's capacity is low (i.e., 0.15, 0.3, and 0.45 Mbps), the modification increases the social welfare as well as reduces the rebuffer ratio (i.e., the ratio of the total rebuffer time to the total video length) and the bitrate degradation ratio (i.e., the ratio of the bitrate degradation amount to the sum of the bitrates of all the received video segments). As user B's capacity becomes large (i.e., 1.5 and 3 Mbps), the modified and unmodfied mechanisms achieve the same performance. This is an expected result because, when both users have high capacities, the unmodified mechanism already has no rebuffer and bitrate degradation and hence no need for modification.



(a) (b)

Figure 3.5: Demonstration system: (a) system architecture; (b) signaling.

## 3.7 Demonstration System

We implement the CMS model on Raspberry PI Model B+ with the Wheezy-Raspbian operating system. In the demonstration system, Raspberry PIs correspond to the mobile devices, which are equipped with monitors (for video playing), LTE USB modems (for LTE connections), and WLAN adapters (for WiFi connections). The devices can dynamically join and leave the cooperative group and there is no need for a centralized control. After joining the cooperative group, the mobile devices download video segments via LTE and forward messages as well as video segments to other devices (if needed) through WiFi connections. Figure 3.5 (a) illustrates the system architecture with the following modules. *User Interface* displays videos to human. *Storage & Controller* stores system information and downloaded videos, and offers other modules necessary control signals. *Video Requester* pulls video segments from servers through LTE links, and *Video Buffer* fetches and stores the segments that are for the device's own video consumption. *Auction* implements our proposed auction mechanism, mainly consisting of *Auctioneer* and *Bidder* modules. When the device acts as an auctioneer, *Auctioneer* module is active and is in charge of the information announcement and auction determination. When the device acts as a bidder, *Bidder* module is active and is in charge of the bid calculation and submission. *Message Dispatcher* transmits and receives auction information, such as auction announcement and bid submission, through WiFi connections. *Transmitter* & *Receiver* transmits the downloaded segment to others and receives the segments downloaded by others through WiFi connections.

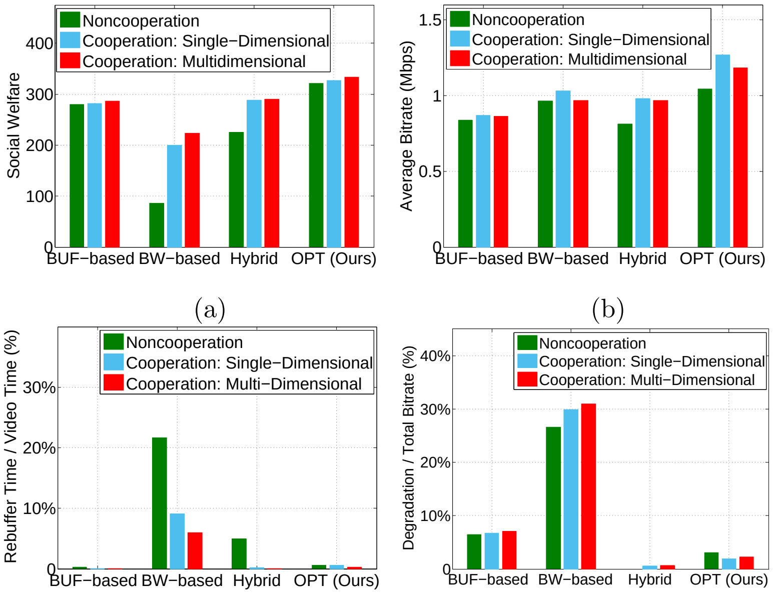
Figure 3.5 (b) shows the signaling between auctioneer's *Auctioneer* module and bidders' *Bidder* modules. The auctioneer first initiates the auction, then, the bidders compute and submit their bids. Since information exchange (e.g., auction initiation and bidding) takes time (due to message passing), we introduce a waiting time (100ms) between the auction initiation and the auction result determination to ensure that all the bids are received before determining auction results. After the auction result determination, the auctioneer announces the results to all the bidders. The winners will send the required segment URL to the auctioneer, and the auctioneer will download the segments and pass to the winners accordingly.

## 3.8 Experiments and performance

The experiments in this section are based on the modified multi-object auction mechanism (Mechanism 3.3). Note that the multi-object mechanism includes the single-object mechanism as a special case by letting $K = 1$.

### 3.8.1 Method Comparison

In this section, we compare our proposed auction scheme with existing methods using real cellular link capacity traces obtained from BesTV. We perform the comparison results for 500 randomly generated network scenarios and show the average results. For each network scenario, we consider 3 users whose cellular link capacities are randomly generated based on the statistics



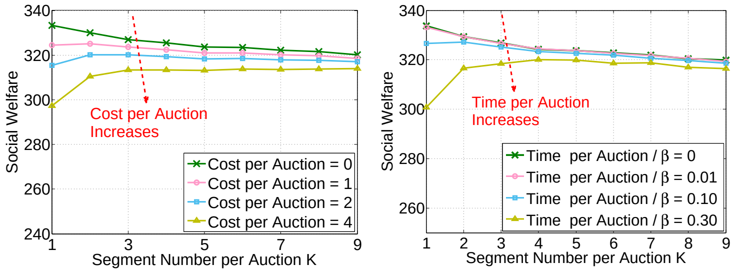
(a) (b)

(a) (b)

Figure 3.6: Comparisons: (a) social welfare; (b) average bitrate; (c) rebuffer; (d) quality degradation.

extracted from real traces, and each user is interested in watching a 100-second video. The available bitrates for all three users' videos are $\{0.2, 0.4, 0.7, 1.3, 2.3\}$Mbps, and the common segment length $\beta = 10$s.

We compare our mechanism with existing methods in two aspects: (i) comparison among three cooperative scenarios—noncooperation, cooperation with single-dimensional (Vickrey) auction [90], and cooperation with multi-dimensional (our proposed Vickrey-score) auction; (ii) bitrate adaptation comparison among buffer-based method ($BUF$-$based$) [51], bandwidth-based method ($BW$-$based$) [65], hybrid buffer-bandwidth method ($Hybrid$) [46], and our optimal bitrate method ($OPT$). For now we do not consider the impact of auction overhead (i.e., auction time and energy consumption), hence it is optimal to choose $K = 1$ segment due to its maximum flexibility to the users. We will consider the impact of overhead and the proper choice of $K$ in Section



(a) (b)

Figure 3.7: Auction overhead: (a) energy consumption; (b) time consumption.

3.8.2.

Figure 3.6 shows the results. For comparison (i), under each of the cooperative scenarios, we take the average among all four methods. Compared with noncooperation, cooperation with multi-dimensional auction increases the social welfare by 48.6%, increases the average bitrate by 8.9%, and reduces the rebuffer by 73.7%. Compared with the cooperation with single-dimensional auction, the cooperation with multi-dimensional auction reduces the rebuffer by 61.4% (as the multi-dimensional auction considers the bitrate adaptation) and increases the social welfare by 3.9%. For comparison (ii), under the scenario of the cooperation with multi-dimensional auction, our mechanism has the highest social welfare (outperforming the other methods by 24.8% on average), the highest bitrate (outperforming the other methods by 25.8% on average), a relatively low rebuffer time (0.26 second on average for a 100-second video), and a relatively low quality degradation (with a degradation ratio[3] of 2.5% on average).

### 3.8.2 Auction Overhead

Now we study the impact of the auction overhead and the proper choice of $K$. Auction mechanism mainly induces two kinds of overheads: energy consumption and time consumption. By increasing the segment number $K$ per auction, both the energy and the time spent on the auctions in a fixed video scheduling cycle (e.g., 100 seconds in our experiment) reduce due to less auctions. We evaluate these two kinds of auction overheads separately. The simulation setting is similar to that of Figure 3.6, except we will change the value of $K$.

For energy consumption, we assume that there is a fixed *cost per auction*, as in Figure 3.7 (a). When the cost per auction is zero, social welfare decreases with the segment number $K$ due to the difficulty in accurately predicting future channel conditions when auctioning a larger number of segments in a single auction. As the cost per auction increases, the social welfare decreases, but a larger $K$ may be better than $K = 1$ because of its smaller total overhead. For the time consumption, we assume that there is a fixed *time per auction* as in Figure 3.7 (b), and we consider different ratios between this time per auction with the video segment length $\beta$. As time per auction increases, social welfare decreases, and a larger $K$ becomes better than $K = 1$ because of its smaller time waste.

### 3.8.3 Realistic Performance over the Demo System

We further perform experiments over the demo system introduced in Section 3.7. The bitrates set is {0.5, 1.0, 2.2, 5.0}Mbps, and the segment length $\beta = 10s$.

[3] The quality degradation ratio is defined as the ratio of the bitrate degradation volume to the sum of the bitrates of all the received video segments. For example, for a sequence of received segments with bitrates $\{1.3, 0.7, 1.3\}$, the degradation ratio is computed as $(1.3 - 0.7)/(1.3 + 0.7 + 1.3) = 18.2\%$.

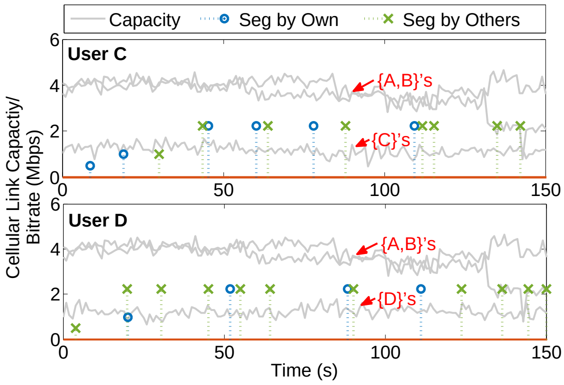


Figure 3.8: Scheduling: user C and user D.

Table 3.3: Welfare comparison.

| | Noncooperation | Cooperation |
|---|---|---|
| A and B | 0% | 15.5% |
| C and D | 49.1% | 84.5% |
| Social Welfare | 49.1% | 100% |

### Welfare Increase for High and Low Capacity Users

In this experiment, four users {A,B,C,D} form a group in a CMS model: user A and B do not watch videos and have cellular link capacities around 3.5Mbps; user C and D watch two different videos and have cellular capacities around 1.2Mbps.

Figure 3.8 shows the video scheduling results of users C and D in one experiment. The meanings of curves and stems are similar as that in Figures 3.3. In Figure 3.8, although user C and D have link capacities around 1.2Mbps, they can download videos at the bitrate of 2.2Mbps most of the time and do not suffer from rebuffer due to the help from users A and B. Table 3.3 shows users' average normalized welfare over four experiment rounds. We normalize the social welfare (i.e., the sum of all the users' welfares) in cooperation as 100%. Without cooperation, users A and B receive zero social welfare, as

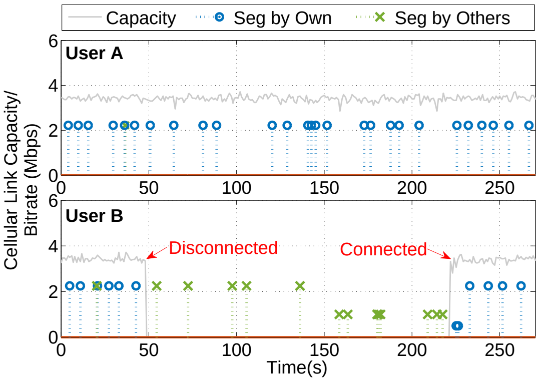


Figure 3.9: Scheduling: user $B$ is disconnected during 50 $\sim$ 220s.

they do not watch videos. Users C and D receive less than 50% of the cooperative total social welfare. Under cooperation, users A and B receive 15.5% of the social welfare due to the payments from the auction (subtracting their own costs for helping other users). For user C and D, their welfare increases 35.4% compared with noncooperation due to the service enhancement. The overall social welfare also increases 50.9% compared with noncooperation.

### Video Streaming Stability

We consider two users, A and B, both of which watch different videos and have cellular capacities around 3.6Mbps. User A is always connected to the Internet, while user B is disconnected from the Internet between 50 to 220 seconds. Figure 3.9 demonstrates the result of an experiment. The notations are similar to that of Figure 3.8. Although user B's video bitrate decreases from 2.2Mbps to 1.0Mbps during the time he is disconnected from the Internet, he is still able to watch the video with the help from user A. This demonstrates the practical benefit of the CMS model.

## 3.9 Chapter Summary

The CMS model enables mobile users to share their downloading capacities for cooperative video streaming. The success of this system requires an effective incentive mechanism that motivates user cooperations. In this chapter, we propose truthful and efficient mechanisms that maximize the social welfare. To the best of our knowledge, the multi-segment mechanism is the first mechanism achieving both truthfulness and efficiency in a multi-object multidimensional auction, overcoming the known challenge of the preferential dependent bidding dimensions (including video segment quality, quantity, and bidders' willingness-to-pay) [23]. We further construct a demo system to evaluate the real world performance of the CMS model.

## 3.10 Appendix

### 3.10.1 Proof of Proposition 3.1

Given any bitrate $r^m$, a bidder $m$'s score will be $\phi^m = U_{m,t}(r^m) - s(r^m)$ if bidding truthfully using price $p^m = U_{m,t}(r^m)$, and $\phi' = p' - s(r^m)$ if bidding untruthfully using price $p' \neq U_{m,t}(r^m)$. We will show that bidder $m$ cannot obtain a higher payoff by bidding $\phi' \neq \phi^m$ (or $p' \neq p^m$), which implies that bidding with price $p^m$ (truthful bidding) is a weakly dominant strategy for bidder $m$.

According to Mechanism 3.1, regardless of what price that bidder $m$ bids, he will obtain a zero payoff if he loses the auction, and he will obtain a payoff of $P_m(p^\dagger, r^\dagger)$ if he wins,

$$\begin{aligned} P_m(p^\dagger, r^\dagger) &= U_{m,t}(r^\dagger) - p^\dagger \\ &= U_{m,t}(r^m) - (\max\{\boldsymbol{\phi}_{\mathcal{N}_n/m}\} + s(r^m)) \\ &= \phi^m - \max\{\boldsymbol{\phi}_{\mathcal{N}_n/m}\}, \end{aligned} \tag{3.47}$$

where $\max\{\boldsymbol{\phi}_{\mathcal{N}_n/m}\}$ is the maximum score other than bidder $m$'s. Hence, if bidder $m$ loses (or wins) under both $\phi'$ and $\phi^m$, he will gain the same payoffs under both the scores. If he loses under $\phi'$ and wins under $\phi^m$, he will gain a zero payoff under $\phi'$, and gains a payoff of $\phi^m - \max\{\boldsymbol{\phi}_{\mathcal{N}_n/m}\} > 0$ under $\phi^m$. If he wins under $\phi'$ and loses under $\phi^m$, he will gain a zero payoff under $\phi^m$, and gains a payoff of $\phi^m - \max\{\boldsymbol{\phi}_{\mathcal{N}_n/m}\} < 0$ under $\phi'$, where the negative payoff is due to the fact that bidder $m$ loses under $\phi^m$. In each of the cases above, bidder $m$ cannot obtain a higher payoff by bidding $\phi' \neq \phi^m$.

### 3.10.2 Proof of Proposition 3.2

The key idea is to show that, for any bid $(\bar{r}^m, \bar{p}^m)$, there always exists a bid $(\hat{r}^m, \hat{p}^m)$, which leads to an expected payoff of bidder $m$ that is no smaller than the bid $(\bar{r}^m, \bar{p})$ does. Such a bid $(\hat{r}^m, \hat{p}^m)$ satisfies two properties: i) the bitrate $\hat{r}^m$ is computed based on (3.19); ii) the price $\hat{p}^m$ is chosen such that $\phi(\hat{r}^m, \hat{p}^m) = \phi(\bar{r}^m, \bar{p}^m)$, which means that both bids $(\hat{r}^m, \hat{p}^m)$ and $(\bar{r}^m, \bar{p}^m)$ lead to the same score and hence the same winning probability. If bidder $m$ loses the auction under such a score, then the payoff will be zero under both bids. If bidder $m$ wins the auction under such a score, then $(\hat{r}^m, \hat{p}^m)$ leads to a larger payoff than $(\bar{r}^m, \bar{p}^m)$, i.e.,

$$\begin{aligned} & U_{m,t}(\hat{r}^m) - (\max\{\boldsymbol{\phi}_{\mathcal{N}_n/m}\} + s(\hat{r}^m)) \\ & \qquad \geq U_{m,t}(\bar{r}^m) - (\max\{\boldsymbol{\phi}_{\mathcal{N}_n/m}\} + s(\bar{r}^m)). \quad (3.48) \end{aligned}$$

Inequality (3.48) holds because $\hat{r}^m$ satisfies equation (3.19).

### 3.10.3 Proof of Proposition 3.3

Suppose all bids *except* those of bidder $m$'s are fixed, so the $K$ highest marginal scores *except* bidder $m$'s, $\hat{\boldsymbol{S}}^{-m} = \{\hat{S}_1^{-m}, \hat{S}_2^{-m}, ..., \hat{S}_K^{-m}\}$ (in the non-increasing order), are fixed. We further assume that bidder $m$'s bitrate matrix

$\boldsymbol{R}^m$ is fixed.

If bidding truthfully, bidder $m$ will submit a price $\boldsymbol{p}^m = (p_1^m, p_2^m, ..., p_K^m)$, where $p_\kappa^m = U_{m,t}(\boldsymbol{r}_\kappa^m)$ for all $\kappa$. Under such a price, bidder $m$ will win $\kappa_m^\dagger$ segments, and has a payoff $P_m$:

$$P_m = U_{m,t}(\boldsymbol{r}_{\kappa_m^\dagger}^m) - (\sum_{i=1}^{\kappa_m^\dagger} \hat{S}_{K-\kappa_m^\dagger+i}^{-m} + s(\boldsymbol{r}_{\kappa_m^\dagger}^m)). \tag{3.49}$$

Let $\boldsymbol{S}^m = \{S_1^m, S_2^m, ..., S_K^m\}$ denote bidder $m$'s marginal score vector (derived from his bids) under the truthful bidding, where $\boldsymbol{S}^m$ satisfies Assumption 3.1. Because of the truthfulness, the marginal score summation satisfies $\sum_{i=1}^{\kappa} S_i^m = p_\kappa^m - s(\boldsymbol{r}_\kappa^m) = U_{m,t}(\boldsymbol{r}_k^m) - s(\boldsymbol{r}_k^m)$ for all $\kappa$. Moreover, the marginal scores of those bidders who win should be no smaller than the marginal scores of those bidders who do not win. Bidder $m$ (with marginal scores $\boldsymbol{S}^m$) wins $\kappa_m^\dagger$ segments, so any of the first $\kappa_m^\dagger$ marginal scores (winning marginal scores) in $\boldsymbol{S}^m$ should be no smaller than any of the last $\kappa_m^\dagger$ marginal scores (losing marginal scores) in $\hat{\boldsymbol{S}}^{-m}$. The bidders except bidder $m$ (with marginal scores $\hat{\boldsymbol{S}}^{-m}$) win $K - \kappa_m^\dagger$ segments, so any of the first $K - \kappa_m^\dagger$ marginal scores (winning marginal scores) in $\hat{\boldsymbol{S}}^{-m}$ should be no smaller than any of the last $K - \kappa_m^\dagger$ marginal scores (losing marginal scores) in $\boldsymbol{S}^m$. Formally,

$$S_i^m \geq \hat{S}_j^{-m}, \ i \leq \kappa_m^\dagger, j \geq K - \kappa_m^\dagger + 1, \tag{3.50}$$

$$\hat{S}_j^{-m} \geq S_i^m, \ i \geq \kappa_m^\dagger + 1, j \leq K - \kappa_m^\dagger. \tag{3.51}$$

If bidding untruthfully, bidder $m$ will submit a price $\bar{\boldsymbol{p}}^m = (\bar{p}_1^m, \bar{p}_2^m, ..., \bar{p}_K^m)$. Under such a price, bidder $m$ will win $\bar{\kappa}_m^\dagger$ segments, and has a payoff $\bar{P}_m$:

$$\bar{P}_m = U_{m,t}(\boldsymbol{r}_{\bar{\kappa}_m^\dagger}^m) - (\sum_{i=1}^{\bar{\kappa}_m^\dagger} \hat{S}_{K-\bar{\kappa}_m^\dagger+i}^{-m} + s(\boldsymbol{r}_{\bar{\kappa}_m^\dagger}^m)). \tag{3.52}$$

According to above discussions, we show that bidder $m$ cannot obtain a higher payoff by submitting $\bar{\boldsymbol{p}}^m \neq \boldsymbol{p}^m$, i.e., we will show $P_m - \bar{P}_m \geq 0$. Considering three possible situations:

- If $\kappa_m^\dagger = \bar{\kappa}_m^\dagger$, then $P_m - \bar{P}_m = 0$.
- If $\kappa_m^\dagger > \bar{\kappa}_m^\dagger$ (loses segments by untruthful bidding), then

$$P_m - \bar{P}_m = \sum_{i=\bar{\kappa}_m^\dagger+1}^{\kappa_m^\dagger} S_i^m - \sum_{i=1}^{\kappa_m^\dagger-\bar{\kappa}_m^\dagger} \hat{S}_{K-\kappa_m^\dagger+i}^{-m} \geq 0. \tag{3.53}$$

- If $\kappa_m^\dagger < \bar{\kappa}_m^\dagger$ (gains segments by untruthful bidding), then

$$P_m - \bar{P}_m = -\sum_{i=\kappa_m^\dagger+1}^{\bar{\kappa}_m^\dagger} S_i^m + \sum_{i=1}^{\bar{\kappa}_m^\dagger-\kappa_m^\dagger} \hat{S}_{K-\bar{\kappa}_m^\dagger+i}^{-m} \geq 0. \tag{3.54}$$

Inequalities (3.53) and (3.54) are obtained based on (3.50) and (3.51).

### 3.10.4 Proof of Proposition 3.4

For any bidder $m$, we will show that given any bid $(\bar{\boldsymbol{R}}^m, \bar{\boldsymbol{p}}^m)$, there always exists a bid $(\tilde{\boldsymbol{R}}^m, \tilde{\boldsymbol{p}}^m)$ that leads to an expected payoff (for bidder $m$) that is no smaller than that achieved by bid $(\bar{\boldsymbol{R}}^m, \bar{\boldsymbol{p}}^m)$. The bid $(\tilde{\boldsymbol{R}}^m, \tilde{\boldsymbol{p}}^m)$ satisfies two properties: i) bitrate $\tilde{\boldsymbol{R}}^m$ is obtained from Proposition 3.4, ii) the marginal score vector of the bid $(\tilde{\boldsymbol{R}}^m, \tilde{\boldsymbol{p}}^m)$ is the same as that of the bid $(\bar{\boldsymbol{R}}^m, \bar{\boldsymbol{p}}^m)$, which implies that the two bids will win the same number of segments, denoted by $\kappa_m^\dagger$.

If $\kappa_m^\dagger = 0$, bidder $m$'s payoff is zero under both the bids. If $\kappa_m^\dagger > 0$, bidder $m$'s payoff under $(\tilde{\boldsymbol{R}}^m, \tilde{\boldsymbol{p}}^m)$ and $(\bar{\boldsymbol{R}}^m, \bar{\boldsymbol{p}}^m)$ are as follows:

$$\tilde{P}_m = U_{m,t}(\tilde{\boldsymbol{r}}_{\kappa_m^\dagger}^m) - (\sum_{i=1}^{\kappa_m^\dagger} \hat{S}_{K-\kappa_m^\dagger+i}^{-m} + s(\tilde{\boldsymbol{r}}_{\kappa_m^\dagger}^m)), \tag{3.55}$$

$$\bar{P}_m = U_{m,t}(\bar{\boldsymbol{r}}_{\kappa_m^\dagger}^m) - (\sum_{i=1}^{\kappa_m^\dagger} \hat{S}_{K-\kappa_m^\dagger+i}^{-m} + s(\bar{\boldsymbol{r}}_{\kappa_m^\dagger}^m)). \tag{3.56}$$

As bitrate $\boldsymbol{R}^m$ is derived through maximizing $U_{m,t}(\boldsymbol{r})-s(\boldsymbol{r})$ under the segment number constraints, i.e.,

$$U_{m,t}(\tilde{\boldsymbol{r}}_{\kappa_m^\dagger}^m) - s(\tilde{\boldsymbol{r}}_{\kappa_m^\dagger}^m) \geq U_{m,t}(\bar{\boldsymbol{r}}_{\kappa_m^\dagger}^m) - s(\bar{\boldsymbol{r}}_{\kappa_m^\dagger}^m), \forall \kappa_m^\dagger \tag{3.57}$$

Hence, $\tilde{P}_m \geq \bar{P}_m$. This completes the proof of Proposition 3.4.

### 3.10.5 Proof of Lemma 3.1

First, we prove that the non-zero elements in the optimal bitrate $\boldsymbol{r}_\kappa^m$ should be in the ascending order, i.e., $r_{\kappa 1}^m \leq r_{\kappa 2}^m \leq ... \leq r_{\kappa\kappa}^m$. We prove this through contradiction. Suppose the optimal bitrate $\boldsymbol{r}_\kappa^m$ is not in the ascending order. By reordering the elements in $\boldsymbol{r}_\kappa^m$ in ascending order, we obtain a new vector $\bar{\boldsymbol{r}}_\kappa$. Note that $V_{m,t}^{\mathrm{Q}}(\cdot)$ and $V_{m,t}^{\mathrm{B}}(\cdot)$ in the bidder's utility $U_{m,t}(\cdot)$ are independent of bitrate order, and the auctioneer $n$'s downloading cost $C_{n,t}(\cdot)$ is also independent of the bitrate order. The degradation loss $L_{m,t}^{\mathrm{QD}}(\cdot)$, however, is minimized when the non-zero elements in $\boldsymbol{r}_\kappa$ are in the ascending order. Hence, we have the following inequality

$$U_{m,t}(\bar{\boldsymbol{r}}_\kappa) - C_{n,t}(\bar{\boldsymbol{r}}_\kappa) \geq U_{m,t}(\boldsymbol{r}_\kappa^m) - C_{n,t}(\boldsymbol{r}_\kappa^m), \tag{3.58}$$

which contradicts the definition of the optimal bitrate, i.e.,

$$\boldsymbol{r}_\kappa^m = \arg\max_{\boldsymbol{r}_\kappa} \left(U_{m,t}(\boldsymbol{r}_\kappa) - C_{n,t}(\boldsymbol{r}_\kappa)\right). \tag{3.59}$$

This proves the ascending order of non-zero bitrate elements.

Next, we prove that the non-zero elements in optimal bitrate $\boldsymbol{r}_\kappa^m$ should be identical, i.e., $r_{\kappa i}^m = r_\kappa^m \; \forall i \leq \kappa$. To simplify the presentation, we define a function $g_{mn,t}(r) = v_{m,t}^{\mathrm{Q}}(r) - c_{n,t}(r), r \in \mathcal{R}_m$. Among the finite bitrate set $\mathcal{R}_m$, there exists an optimal bitrate, denoted by $r^* \in \mathcal{R}_m$, that maximizes the concave function $g_{mn,t}(r)$. As we have shown that the non-zero bitrates will be in the ascending order, the only possible bitrate degradation is the degradation from $R_{m,t}^{\mathrm{PRE}}$ (i.e., the bitrate of the last segment from the previous auction) to $r_{\kappa 1}^m$ (the bitrate of the first allocated segment in this auction). Hence, we can rewrite the bitrate vector optimization problem as follows:

$$\boldsymbol{r}_\kappa^m = \arg\max_{\boldsymbol{r}_\kappa} \left( \sum_{i=1}^{\kappa} g_{mn,t}(r_{\kappa i}) - l^{\mathrm{QD}}(R_{m,t}^{\mathrm{PRE}}, r_{\kappa 1}) \right). \tag{3.60}$$

We show $r_{\kappa i}^m = r_\kappa^m$, $\forall i \leq \kappa$ in the following two cases:

- If $R_{m,t}^{\text{PRE}} < r^*$, then we know that bitrate $r^*$ maximizes $g_{mn,t}(r)$ and minimizes $l^{\text{QD}}(R_{m,t}^{\text{PRE}}, r)$ among the feasible set $r \in \mathcal{R}_m$. Hence, the optimal bitrate vector $\boldsymbol{r}_\kappa^m$ satisfies $r_{\kappa 1}^m = r_{\kappa 2}^m = ... = r_{\kappa\kappa}^m = r^*$.
- If $R_{m,t}^{\text{PRE}} \geq r^*$, then we prove the identical bitrate result as follows. First, we have bitrates $r^* \leq r_{\kappa 1}^m$ for the following reasons. If $r_{\kappa 1}^m$ is the optimal value for $r_{\kappa 1}$, then the partial derivative of the objective function in (3.60) with the respect to $r_{\kappa 1}$ at $r_{\kappa 1} = r_{\kappa 1}^m$ should be zero, i.e., $[g_{mn,t}(r_{\kappa 1}^m)]_{r_{\kappa 1}} - [l^{\text{QD}}(R_{m,t}^{\text{PRE}}, r_{\kappa 1}^m)]_{r_{\kappa 1}} = 0$. Hence, $[g_{mn,t}(r_{\kappa 1}^m)]_{r_{\kappa 1}} \leq 0$, which implies that $r^* \leq r_{\kappa 1}^m$ holds because $g_{mn,t}(\cdot)$ is concave and is maximized at $r^*$. Second, the concave function $g_{mn,t}(r)$ is non-increasing in $r$ for $r \geq r_{\kappa 1}^m \geq r^*$. We have proved that $r_{\kappa 1}^m \leq r_{\kappa 2}^m \leq ... \leq r_{\kappa\kappa}^m$, so $g_{mn,t}(r_{\kappa 2}^m), g_{mn,t}(r_{\kappa 3}^m), ..., g_{mn,t}(r_{\kappa\kappa}^m)$ are maximized when bitrates $r_{\kappa 2}, r_{\kappa 3}, ..., r_{\kappa\kappa}$ are minimized under the constraint that $r_{\kappa 1}^m \leq r_{\kappa 2}^m \leq ... \leq r_{\kappa\kappa}^m$, which implies $r_{\kappa 2}^m = ... = r_{\kappa\kappa}^m = r_{\kappa 1}^m$.

### 3.10.6 Proof of Lemma 3.2

According to Lemma 3.1, the optimal common non-zero bitrate $r_\kappa^m$ for each row $\kappa$ is derived as follows:

$$r_\kappa^m = \arg\max_r \left(\kappa \cdot g_{mn,t}(r) - l^{\text{QD}}(R_{m,t}^{\text{PRE}}, r)\right). \tag{3.61}$$

We prove that $r_\kappa$ is non-increasing in row index $\kappa$ by checking two cases:

- If $R_{m,t}^{\text{PRE}} < r^*$, then bitrate $r^*$ maximizes $\kappa \cdot g_{mn,t}(r) - l^{\text{QD}}(R_{m,t}^{\text{PRE}}, r)$ for any $\kappa$. Hence, $r_1^m = r_2^m = ... = r_K^m = r^*$.
- If $R_{m,t}^{\text{PRE}} \geq r^*$, then bitrate $r^* \leq r_\kappa^m = r_{\kappa 1}^m$ (proved in Property 1). The optimal bitrate $r_\kappa^m$ and $r_{\kappa+1}^m$ satisfy:

$$r_\kappa^m = \arg\max \left(\kappa \cdot g_{mn,t}(r) - l^{\text{QD}}(R_{m,t}^{\text{PRE}}, r)\right); \tag{3.62}$$

$$r^m_{\kappa+1} = \arg\max\left(\kappa \cdot g_{mn,t}(r) - l^{\mathrm{QD}}(R^{\mathrm{PRE}}_{m,t}, r) + g_{mn,t}(r)\right). \tag{3.63}$$

Function $g_{mn,t}(\cdot)$ is concave and non-increasing in $r$ when $r \in \mathcal{R}_m$ and $r \geq r^*$. Suppose $r^m_\kappa < r^m_{\kappa+1}$. Then based on the definition of $r^m_\kappa$, we have

$$\begin{aligned}\kappa \cdot g_{mn,t}(r^m_{\kappa+1}) - l^{\mathrm{QD}}(R^{\mathrm{PRE}}_{m,t}, r^m_{\kappa+1}) + g_{mn,t}(r^m_{\kappa+1})& \\ < \ \kappa \cdot g_{mn,t}(r^m_\kappa) - l^{\mathrm{QD}}(R^{\mathrm{PRE}}_{m,t}, r^m_\kappa) + g_{mn,t}(r^m_\kappa),&\end{aligned} \tag{3.64}$$

which contradicts to the definition of the optimal bitrate $r^m_{\kappa+1}$. Hence, $r^m_\kappa \geq r^m_{\kappa+1}$ for all $\kappa = 1, 2, ..., K-1$.

This completes the proof for Lemma 3.2.

### 3.10.7 Proof of Proposition 3.5

Considering efficient score function in (3.38), if a bidder $m$ bids according to Proposition 3.3 and 3.4, then the bidder's score for being allocated a total of $\kappa$ segments is given:

$$\begin{aligned}\phi^{m,n,t}_\kappa = &\underset{\boldsymbol{r}}{\text{maximize}} && U_{m,t}(\boldsymbol{r}) - C_{n,t}(\boldsymbol{r}) \\ &\text{subject to} && r_i > 0, \quad i = 1, ..., \kappa, \\ & && r_i = 0, \quad i = \kappa+1, ..., K, \\ &\text{variables} && r_i \in \mathcal{R}_m, \quad i = 1, ..., \kappa.\end{aligned} \tag{3.65}$$

Hence, the conditions on the marginal scores in Assumption 3.1 can be written as equivalent conditions of $\phi^{m,n,t}_\kappa$ as follows:

- Non-negative marginal score:

$$\phi^{m,n,t}_{\kappa+1} - \phi^{m,n,t}_\kappa \geq 0, \forall \kappa = 1, 2, ..., K-1 \tag{3.66}$$

- Non-increasing marginal score:

$$\phi^{m,n,t}_\kappa - \phi^{m,n,t}_{\kappa-1} \geq \phi^{m,n,t}_{\kappa+1} - \phi^{m,n,t}_\kappa, \ \forall \kappa = 1, 2, ..., K-1 \tag{3.67}$$

Next, we show that inequalities (3.43) and (3.44) are the sufficient conditions for satisfying (3.66) and (3.67), respectively.

***Non-negative***: If $v_m^{\mathrm{Q}}(r, \theta) \geq c_{n,t}(r)$ for all $r$ and $\theta$, then $g_{mn,t}(r) = v_{m,t}^{\mathrm{Q}}(r) - c_{n,t}(r) \geq 0$ always holds. Based on Lemma 3.1, the score $\phi_\kappa^{m,n,t}$ can be represented as follows:

$$\phi_\kappa^{m,n,t} = \max_r \left(\kappa \cdot g_{mn,t}(r) - l^{\mathrm{QD}}(R_{m,t}^{\mathrm{PRE}}, r)\right). \tag{3.68}$$

Let $r_\kappa^m$ and $r_{\kappa+1}^m$ be the optimal non-zero common bitrates for rows $\kappa$ and $\kappa+1$, respectively. Based on Lemma 3.1 and 3.2,

$$\begin{aligned}\phi_{\kappa+1}^{m,n,t} = (\kappa+1) \cdot g_{mn,t}(r_{\kappa+1}^m) - l^{\mathrm{QD}}(R_{m,t}^{\mathrm{PRE}}, r_{\kappa+1}^m)\\ \geq (\kappa+1) \cdot g_{mn,t}(r_\kappa^m) - l^{\mathrm{QD}}(R_{m,t}^{\mathrm{PRE}}, r_\kappa^m)\\ \geq \kappa \cdot g_{mn,t}(r_\kappa^m) - l^{\mathrm{QD}}(R_{m,t}^{\mathrm{PRE}}, r_\kappa^m) = \phi_\kappa^{m,n,t},\end{aligned} \tag{3.69}$$

which proves that $\phi_{\kappa+1}^{m,n,t} - \phi_\kappa^{m,n,t} \geq 0$.

***Non-increasing***: The non-increasing marginal score requirement is equivalent to the following one:

$$\begin{aligned}(\kappa+1)g_{mn,t}(r_{\kappa+1}^m) - 2\kappa g_{mn,t}(r_\kappa^m) + (\kappa-1)g_{mn,t}(r_{\kappa-1}^m)\\ - l^{\mathrm{QD}}(R_{m,t}^{\mathrm{PRE}}, r_{\kappa+1}^m) - 2l^{\mathrm{QD}}(R_{m,t}^{\mathrm{PRE}}, r_\kappa^m) - l^{\mathrm{QD}}(R_{m,t}^{\mathrm{PRE}}, r_{\kappa-1}^m) \leq |\tilde{\Delta}|.\end{aligned} \tag{3.70}$$

Based on Lemma 3.1 and 3.2, we derive the conditions for satisfying inequality (3.70) in the following two cases:

- If $R_{m,t}^{\mathrm{PRE}} < r^*$, the bitrate $r_\kappa^m = r_{\kappa-1}^m = r_{\kappa-2}^m = r^*$. Hence, inequality (3.70) is directly satisfied.
- If $R_{m,t}^{\mathrm{PRE}} \geq r^*$, then the inequality (3.70) is satisfied if:

  $$2(\kappa+1)(c_{n,t}(r_\kappa^m) - c_{n,t}(r_{\kappa-1}^m)) + l^{\mathrm{QD}}(R_{m,t}^{\mathrm{PRE}}, r_\kappa^m) \leq |\tilde{\Delta}|, \tag{3.71}$$

  because $g_{mn,t}(\cdot)$ is concave and non-increasing in $r$ when $r \geq r^*$, and $l^{\mathrm{QD}}(\cdot)$ is non-increasing in $r$. Since $c_{n,t}(r_\kappa^m) - c_{n,t}(r_{\kappa-1}^m) \leq c_{n,t}(R_m^Z) -$

$c_{n,t}(0) = c_{n,t}(R_m^Z)$, $l^{\mathrm{QD}}(R_{m,t}^{\mathrm{PRE}}, r_\kappa^m) \le l_m^{\mathrm{QD}}(R_m^Z, 0)$, and $\kappa + 1 \le K$, we have the sufficient condition for non-increasing marginal score:

$$2K \cdot c_{n,t}(R_m^Z) + l_m^{\mathrm{QD}}(R_m^Z, 0) \le |\tilde{\Delta}|. \tag{3.72}$$

# Chapter 4

# Communication, Computation, and Caching Sharing

## 4.1 Introduction

### 4.1.1 Background and Motivation

Humans have being increasingly enjoying multimedia services over the Internet, thanks to the fast development of communication and information technologies. For example, an US adult spent 5.9 hours daily on average on multimedia services over the Internet in 2017, while this value was 3.2 hours in 2010 [67]. Such an increasing of multimedia demand is more obvious in mobile networks. For example, an US adult spent 3.3 hours daily on average on mobile multimedia in 2017, while this value was 0.4 hour in 2010 [67].

Despite the large mobile multimedia demand, providing high quality-of-experience (QoE) multimedia services over mobile networks is challenging due to two main reasons. First, the multimedia service tasks always request a large amount of *communication resources* (e.g., stream downloading and uploading), *computation resources* (e.g., stream decoding and encoding), and *caching resources* (e.g., stream storage), named "3C resources". On the other hand, comparing with wired devices and networks, mobile devices and networks always limit in their 3C resources. Second, different mobile users can have very different service requirements (e.g., high quality or low quality videos depending on the device capabilities and the user preferences) and mobile device and network resources (e.g., 3G or 4G cellular links), which leads to challenges for effective QoE provision. This also leads to the potential mismatch of service requirement and resource supply at a single user level.

To provide high QoE mobile multimedia services, a promising approach is to enable device-to-device (D2D) based resource sharing among mobile devices, and to allocate their 3C resources efficiently for their cooperative multimedia service task executions.

Many of the existing works focused on the sharing of one resource [55, 85, 32, 34, 57, 33]. For example, the user-provided networking in [55, 85] focus on the sharing of communication resource, the ad hoc computation offloading in [32, 34] focus on the sharing of computation resource, and the ad hoc content sharing in [57, 33] focus on the sharing of caching resource. We refer to these models as 1C sharing models, since each of them focuses on one type of the 3C resources. Some other recent works further considered the sharing of two types of the 3C resources, which we call the 2C sharing model. Typical examples of 2C sharing include the distributed data analysis in [84, 38], which focus on the sharing of computation and caching resources.

Despite the success of the earlier 1C/2C resource sharing models, there are still significant potential benefits of exploiting the joint 3C resource sharing framework. Such a 3C sharing framework can further improve the resource utilization efficiency, by offering more flexibilities in terms of device cooperation and resource scheduling. Regarding the device cooperation, a joint 3C framework can enable devices performing different tasks to cooperate with each other, which leads to an increased number of participating devices and

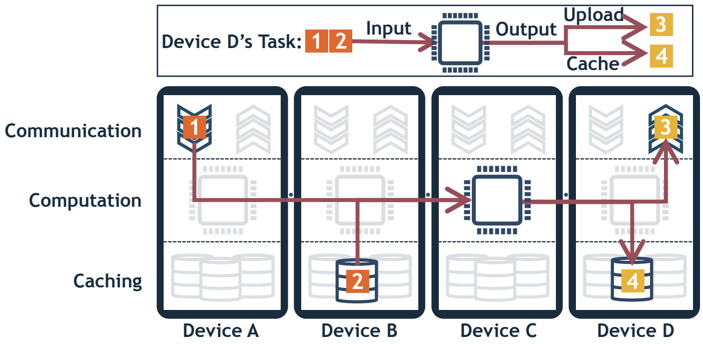


Figure 4.1: An example of the general 3C framework.

hence more cooperation opportunities. Regarding the resource scheduling, the joint optimization of 3C resources can lead to a more efficient resource allocation.

### 4.1.2 Solution Approach and Contribution

In this chapter, we present the first study regarding the general 3C resource sharing framework, which aims to generalize existing mobile device resource sharing (1C/2C) models and provide additional network design and optimization flexibilities. A key feature of this new 3C sharing framework is that it is centered around the characterization of the resource requirements of the tasks initialized by mobile devices, i.e., "resource-centric", instead of emphasizing on the classification of these tasks, i.e., "task-centric". In other words, any of the tasks (e.g., content retrieving, data analysis, or uploading) is modeled by the resources that it requests, so that various types of tasks requesting any combination of the 3C resources can coexist in the same framework.

Figure 4.1 illustrates a simple example of the proposed 3C framework, where four devices {A, B, C, D} connect with each other via D2D and share their 3C resources to complete tasks. In this example, device D initializes a task that involves the following procedures: (i) retrieving contents "1" and

"2" (either downloading from the Internet or fetching from some devices' caches), (ii) performing computation, and (iii) outputting contents "3" (to the Internet) and "4" (to device D's local cache). With the 3C framework, devices can share their communication (downlink and uplink), computation (CPU), and caching resources. In this example, device A and B are responsible for obtaining the inputting contents and delivering them to device C for computation, then device C performs the computation and sends the outputting contents to device D, and finally device D further uploads content "3" and caches content "4" in its local cache.

To show the benefits of the 3C framework concretely, we focus on the energy consumption of mobile devices, and solve an energy consumption minimization problem under the 3C framework. Note that the proposed methodology can also be applied to other system optimization objectives (e.g., delay minimization or QoS maximization). A common feature of these optimization problems under the 3C framework is the introduction of integer variables due to the consideration of caching sharing (e.g., who caches which content). The existence of the integer variables introduces difficulties in analyzing and solving the proposed problem. Moreover, tasks are often correlated with each other (e.g., due to the delays generated by resource sharing), which further complicates the problem solving. We solve the energy minimization problem systematically and discuss the energy reduction due to the 3C sharing both analytically and numerically. Our key contributions are summarized as follows:

- *General 3C Resource Sharing Framework:* We propose a general 3C sharing framework and the corresponding "resource-centric" mathematical formulation. This framework generalizes many existing 1C/2C resource sharing models, and improves the resource utilization efficiency by encouraging more devices participating and more flexible resource scheduling.

- *Energy Efficiency Optimization:* We focus on the energy consumption of mobile devices under the 3C framework, and formulate and solve an energy consumption minimization problem. The problem is difficult as it is an integer non-convex optimization problem. We first transform it to an integer linear programming problem, and then proposed a linear programming heuristic algorithm, which can produce an output that is empirically close to the optimal solution.

- *Theoretical Performance Analysis:* We analyze the energy consumption reduction due to the 3C resource sharing analytically. We show that if the 3C framework can double the number of cooperative devices (comparing with 1C models), it can reduce the energy by a maximum of about 20% of the energy consumed in noncooperation case (where devices do not cooperate).

- *Simulation and Performance Evaluation:* Comparing with existing 1C/2C sharing approaches, 3C sharing reduces the total energy by 83.8% when the D2D energy consumption is negligible, and the energy reduction is still 27.5% when the D2D energy per unit time is twice as large as the cellular energy per unit time. As for the computational complexity, when the network size is moderate (e.g., 27 devices), the heuristic algorithm reduces the computation time by 78.6% at the expense of an optimality gap of 11.2%.

The rest of this chapter is organized as follows. Section 4.2 reviews the related work, and Section 4.3 presents the 3C framework. In Section 4.4, we formulate the energy efficiency optimization problem, and present the problem transformation and heuristic algorithm design. In Section 4.5, we analyze the energy reduction due to the 3C framework. We then perform simulations on optimal and heuristic solutions comparison as well as 1C/2C and 3C approaches comparison in Section 4.6, and conclude in Section 4.7.

## 4.2 Literature Review

There have been extensive studies working on the 1C/2C sharing models. Due to the limited space, we are only able to briefly discuss some representative works that are most closely related to this study.

Most of the existing works considered 1C models. For example, Iosifidis *et al.* [55] and Syrivelis *et al.* [85] proposed user-provided network models for communication resource sharing, where nearby devices share their Internet connectivity for cooperative downloading. Militano *et al.* [68] proposed a uploading resource sharing model, where devices form effective coalitions to share their uploading resources. Chi *et al.* [32] and Chen *et al.* [34] proposed ad hoc computation offloading models for computation resource sharing, where nearby mobile devices share their computation resources for data processing. Jiang *et al.* [57] and Chen *et al.* [33] proposed ad hoc content sharing models for cached content sharing, where mobile devices share their cached contents through D2D connections.

Some recent works further considered 2C models. For example, Stojmenovic *et al.* [84] and Destounis, *et al.* [38] considered distributed data analysis models, where some mobile devices share their cached data and other devices share their computation resources to process the shared data.

There are two limitations of the existing 1C/2C models: (i) due to commonly adopted "task-centric" approach, devices with different types of tasks cannot cooperate (e.g., devices in user-provided network cannot cooperate with devices in ad hoc computation offloading), which restricts the pool of cooperative devices; and (ii) some tasks may request all of the 3C resources, which cannot be handled by these existing 1C/2C models. In comparison, our proposed 3C framework addresses the above two limitations and improve resource utilization efficiency by providing more device cooperation and resource scheduling flexibilities.

## 4.3 A General 3C Sharing Framework

### 4.3.1 System Model

We consider three key elements in the 3C framework: devices, tasks, and contents.

- **Device set** $\mathcal{N} = \{1, 2, ..., N\}$: The devices form a mesh network (through D2D connections) for cooperative task execution. For any device $n \in \mathcal{N}$, let $\mathcal{E}(n)$ denote the set of devices connected with device $n$ through D2D connections. Note that $n \in \mathcal{E}(n)$.
- **Task set** $\S = \{1, 2, .., S\}$: The devices initialize these tasks, where a device may initialize one or more tasks.
- **Content set** $\mathcal{K} = \{1, 2, ..., K\}$: The contents can be the inputting or outputting of the tasks. They can be downloaded from the Internet, cached by the devices, or produced by computations. A content $k \in \mathcal{K}$ has a size of $L_k$ (in bits).

Next we provide detailed explanations of devices and tasks.

#### Device Model

A device is characterized as follows.

**Definition 4.1** (Device Model)**.** *Each device $n \in \mathcal{N}$ corresponds to a collection of tasks and resources, denoted by*

$$\boldsymbol{Q}_n = (\boldsymbol{s}_n, Q_n^{down}, Q_n^{cpu}, Q_n^{up}, \boldsymbol{Q}_n^{ca}), \tag{4.1}$$

*where each notation refers to a feature of the device:*

$\boldsymbol{s}_n$ - *the vector of its initializing tasks' indexes, where the dimension is the number of tasks it initializes*

$Q_n^{down}$ - *downloading capacity (in bits per second),*

$Q_n^{cpu}$ - *computation capacity (in CPU cycles per second),*

$Q_n^{up}$ - *uploading capacity (in bits per second),*

$\boldsymbol{Q}_n^{ca}$ - *the vector of cached contents of dimension $K$, where for any content $k \in \mathcal{K}$, $Q_{nk}^{ca} = 1$ if $n$ has cached content $k$, and $Q_{nk}^{ca} = 0$ otherwise.*

*Let $c_n^{down}$, $c_n^{cpu}$, and $c_n^{up}$ denote the energy consumption of the downloading, computation, and uploading operations per unit second, respectively.*

Next we define the model of the D2D connections.

**Definition 4.2** (Device-to-Device Model)**.** *For any two different devices $n, m \in \mathcal{N}$, let $Q_{n \to m}^{d2d}$ denote the D2D transmission capacity (bits per second) from device $n$ to device $m$, and let $c_{n \to m}^{d2d}$ denote the corresponding D2D transmission energy per second.*

### Task Model

Each task $s \in \S$ is represented by the task model shown in Figure 4.2. Specifically, each task has a *computation* module (which can have a zero computation requirement if the task does not involve any computation). The computation module requests some inputting contents (downloading from the Internet or fetching from devices' caches) and produces some outputting contents (for uploading to the Internet or caching at the task owner's device).

**Definition 4.3** (Task Model)**.** *Each task $s \in \S$ is denoted by*

$$\boldsymbol{D}_s = (u_s, \boldsymbol{D}_s^{in}, D_s^{cpu}, \boldsymbol{D}_s^{up}, \boldsymbol{D}_s^{ca}), \tag{4.2}$$

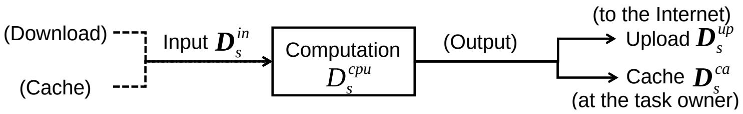

(Download)
(Cache)
Input $\boldsymbol{D}_s^{in}$
Computation
$D_s^{cpu}$
(Output)
(to the Internet)
Upload $\boldsymbol{D}_s^{up}$
Cache $\boldsymbol{D}_s^{ca}$
(at the task owner)

Figure 4.2: The task model.

*where each notation refers to a feature of the task:*

$u_s$ - *task owner (i.e., the device initializes this task),*

$\boldsymbol{D}_s^{in}$ - *the vector of the inputting contents,*

$D_s^{cpu}$- *computation requirement (in total CPU cycles),*

$\boldsymbol{D}_s^{up}$ - *the vector of the uploading contents,*

$\boldsymbol{D}^{ca}$ - *the vector of the caching contents.*

*All three* $\boldsymbol{D}_s^{in}$, $\boldsymbol{D}_s^{up}$, *and* $\boldsymbol{D}_s^{ca}$ *have the same size of* $K$. *For any* $k \in \mathcal{K}$, $D_{sk}^X = 1$ *(*$X \in in, up, ca$*) if content* $k$ *is requested by task* $s$ *for inputting, uploading, or caching, respectively, and* $D_{sk}^X = 0$ *otherwise.)*

Each task can be divided into several subtasks:

**Definition 4.4** (Subtask)**.** *Each task consists of three subtasks: inputting, computation, and uploading*[1] *subtasks.*

We consider a general 3C framework, where the inputting, computation, and uploading subtasks (of a same task) can be performed by different devices. Moreover, multiple contents requested by the same task can be inputted from or uploaded by different devices, leading to the maximum resource sharing flexibility.

[1] The contents to be cached, i.e., $\boldsymbol{D}^{ca}$, are cached at the task owner, so we do not regard it as a separate subtask. It is possible to consider the extension where the outputting content can be cached at other devices. This will add additional complexity of the model without significantly changing the result. We will not consider this due to space limit.

### 4.3.2 Problem Statement

One key concern of mobile devices is their energy consumptions, hence as a concrete example, we focus on an energy minimization under the 3C framework in this chapter.

Next we first introduce decision variables, constraints, and energy calculations. This enables us to present the energy minimization problem.

#### Decision Variables

We consider a set of *binary* decision variables with possible values from $\{0, 1\}$ as follows. A variable equals 1 if its corresponding description is true, and equals 0 otherwise.

$x^{in}_{s,k\to n}$ - device $n$ inputs task $s$' content $k$,
$x^{down}_{s,k\to n}$ - device $n$ downloads task $s$' content $k$,
$x^{cpu}_{s\to n}$ - device $n$ performs task $s$' computation,
$x^{up}_{s,k\to n}$ - device $n$ uploads task $s$' content $k$,
$z^{in}_{s,k\to i,j}$- task $s$' inputting content $k$ is delivered from $i$ to $j$,
$z^{up}_{s,k\to i,j}$- task $s$' uploading content $k$ is delivered from $i$ to $j$,
$z^{ca}_{s,k\to i,j}$- task $s$' caching content $k$ is delivered from $i$ to $j$.

Here $i$ and $j$ refer to devices from set $\mathcal{N}$.

#### Constraints

The system should satisfy allocation constraints, capacity constraints, network flow balancing constraints, and worst delay constraints as follows.

***Allocation constraints:*** Any task $s$' computation subtask should be allocated to only one device:

$$\sum_{n\in\mathcal{N}} x^{cpu}_{s\to n} = 1, \ \forall s \in \mathcal{S}. \tag{4.3}$$

The inputting of a requested content should be allocated to one device, i.e.,

$$\sum_{n\in\mathcal{N}} x^{in}_{s,k\to n} = D^{in}_{sk},\ \forall s\in\mathcal{S}, k\in\mathcal{K}. \tag{4.4}$$

The uploading of a requested content should be allocated to one device, i.e.,

$$\sum_{n\in\mathcal{N}} x^{up}_{s,k\to n} = D^{up}_{sk},\ \forall s\in\mathcal{S}, k\in\mathcal{K}. \tag{4.5}$$

***Capacity constraints:*** The device responsible for inputting a content should either has the content in its local cache or will download it from the Internet:

$$x^{in}_{s,k\to n} \le Q^{ca}_{nk} + \sum_{s\in\mathcal{S}} x^{down}_{s,k\to n},\ \forall s\in\mathcal{S}, n\in\mathcal{N}, k\in\mathcal{K}. \tag{4.6}$$

We consider the sum of the downloading subtask indicators, i.e., $\sum_{s\in\mathcal{S}} x^{down}_{s,k\to n}$, as the device will own the content once it downloads content $k$ for any of the tasks $s\in\mathcal{S}$.

***Network flow balancing constraints:*** These constraints are related to the content delivery through multi-hop transmissions. For a particular content at any device, the incoming number of the copies of the content (i.e., the number of the copies of the content that it receives and generates) should be equal to the outgoing ones (i.e., the number of the copies of the content that it transmits and consumes).

Taking the inputting content transmission variable $z^{in}_{s,k\to i,j}$ as an example. For any task $s\in\mathcal{S}$ and content $k\in\mathcal{K}$, the network flow balancing constraint at a device $i\in\mathcal{N}$ is

$$\sum_{j\in\mathcal{E}(i)} z^{in}_{s,k\to j,i} + x^{in}_{s,k\to i} D^{in}_{sk} = \sum_{j\in\mathcal{E}(i)} z^{in}_{s,k\to i,j} + x^{cpu}_{s\to i} D^{in}_{sk}. \tag{4.7}$$

The left-hand side of (4.7) is the incoming number of task $s$' inputting content $k$, including the number of the content copies that device $i$ receives from its nearby devices and the number of the content copies it generates for inputting (which equals one if $x^{in}_{s,k\to i} = 1$ and $D^{in}_{sk} = 1$). The right-hand side of (4.7) is the outgoing number of task $s$' inputting content $k$, including the number of the content copies that device $i$ transmits to its nearby devices and the number of the content copies it consumes to perform computation (which equals one if $x_{s\to i}^{cpu}=1$ and $D_{sk}^{in}=1$).

Applying similar arguments to caching and uploading, we obtain the following constraints:

$$\sum_{j\in\mathcal{E}(i)} z_{s,k\to j,i}^{ca} + x_{s\to i}^{cpu} D_{sk}^{ca} = \sum_{j\in\mathcal{E}(i)} z_{s,k\to i,j}^{ca} + \mathbf{1}_{i=u_s} D_{sk}^{ca}; \tag{4.8}$$

$$\sum_{j\in\mathcal{E}(i)} z_{s,k\to j,i}^{up} + x_{s\to i}^{cpu} D_{sk}^{up} = \sum_{j\in\mathcal{E}(i)} z_{s,k\to i,j}^{up} + x_{s,k\to i}^{up} D_{sk}^{up}. \tag{4.9}$$

Operator $\mathbf{1}_{i=u_s}=1$ if $i=u_s$, and $\mathbf{1}_{i=u_s}=0$ if $i\neq u_s$.

***Worst Delay Constraints:*** The worst delay is the maximum delay that may happen due to the resource sharing. We first explain the worst case delay constraints, and then compute the worst delays.

First, to execute a task $s$, the *worst* (maximum) delay of downloading[2], computation, and uploading subtasks, denoted by $T_s^X$ ($X\in\{down, cpu, up\}$), should be smaller than the corresponding delay bounds, respectively:

$$T_s^{down} \le \bar{T}_s^{down}, T_s^{cpu} \le \bar{T}_s^{cpu}, T_s^{up} \le \bar{T}_s^{up},\ \forall s\in\mathcal{S}. \tag{4.10}$$

By using these separate delay constraints, we want to characterize the delay of each of the subtasks of a task. In addition, we ignore the D2D transmission delay for simplification.[3]

Then, we show how to calculate the worst delays ($T_s^{down}$, $T_s^{cpu}$, and $T_s^{up}$). The first step is to calculate a device's time spending on completing all the subtasks allocated to it. Specifically, let $\tau_n^{down}$, $\tau_n^{cpu}$, and $\tau_n^{up}$ denote device

[2] Inputting contents may come from downloading from the Internet or fetching from caches. For simplicity, we assume that fetching from caches is instantaneous, so we can focus on the delay caused by downloading.

[3] In reality, such a delay is usually relatively small. For example, Wi-Fi Direct (`https://www.wi-fi.org/`) has a transmission speed of up to 250Mbps.

$n$'s total time spending on completing all the downloading, computation, and uploading subtasks allocated to it, respectively:

$$\tau_n^{down} = \frac{\sum_{s=1}^{S} \sum_{k=1}^{K} x_{s,k\to n}^{down} L_k}{Q_n^{down}}, \tag{4.11}$$

$$\tau_n^{up} = \frac{\sum_{s=1}^{S} \sum_{k=1}^{K} x_{s,k\to n}^{up} L_k}{Q_n^{up}}, \tau_n^{cpu} = \frac{\sum_{s=1}^{S} x_{s\to n}^{cpu} D_r^{cpu}}{Q_n^{cpu}}. \tag{4.12}$$

The second step is to calculate the worst delays. Using (4.11) as an example, we explain how to compute the worst downloading delay $T_s^{down}$. We will first discuss the maximum downloading time that a device $n$ can impose on a task that it downloads for, and then discuss how a task $s$ (requesting downloading from multiple devices) computes its worst downloading delay $T_s^{down}$.

For any device $n$, it may download contents for multiple tasks, and the multiple tasks may be ready for downloading at different times.[4] For simplicity, we assume that, if a device is downloading for multiple tasks at the same time, the device divides its total downloading capacity among the multiple tasks according to their total downloading volumes. For example, a device $n$ is downloading two contents (with sizes 1MB and 2MB, respectively) for task A, and one content (with a size 6MB) for task B. Then, the device $n$ allocates its fraction of the downloading capacity that allocated to task A and task B is $(1+2)/(1+2+6) = 1/3$ and $6/(1+2+6) = 2/3$ of its downloading capacity to task A and task B, respectively. Under such settings, the maximum downloading time that device $n$ can impose on a task is its total time spending on downloading, i.e., $\tau_n^{down}$. This happens when all the tasks (allocated to device $n$) is ready for downloading at the same time, under which these tasks will share the device $n$'s capacity during the entire downloading process.

[4] The case of different ready times is more obvious for computation and uploading subtasks. Specifically, the computation of a task is ready for execution only when the corresponding downloading has finished, and this time could be different for different tasks. Similar for uploading subtasks.

For any task $s$, it can obtain multiple contents from different devices. The worst downloading delay that task $s$ experiences is the maximum downloading time $\tau_n^{down}$ among the device set $\{n|\sum_{k=1}^{K} x_{s,k:\to n}^{in}(1-Q_{nk}^{ca})>0\}$. This set refers to the set of devices that are responsible for task $s$' inputting but have not cached the contents yet (so they have to download the contents). Formally,

$$T_s^{down} = \max_{\{n|\sum_{k=1}^{K} x_{s,k:\to n}^{in}(1-Q_{nk}^{ca})>0\}} \tau_n^{down}. \tag{4.13}$$

A similar idea applies to the calculation of the worst computation and uploading delays. Specifically, the worst computation delay is the maximum computation time of the device who performs task $s$' computation:

$$T_s^{cpu} = \sum_{n=1}^{N} x_{s\to n}^{cpu} \tau_n^{cpu}. \tag{4.14}$$

The worst uploading delay is the maximum uploading time $\tau_n^{up}$ among all the devices who perform task $s$' uploading:

$$T_s^{up} = \max_{\{n|\sum_{k=1}^{K} x_{s,k\to n}^{up}>0\}} \tau_n^{up}. \tag{4.15}$$

To clarify, these delay constraints are non-convex, since they contain quadratic forms that are not positive semidefinite, which makes it difficult to solve the energy minimization problem (to be presented in Section 4.3.2).

#### Energy Calculations

The energy for executing a task $s$ consists of the energy for downloading, computation, uploading, and D2D transmission. Formally,

$$E_s = E_s^{down} + E_s^{cpu} + E_s^{up} + E_s^{d2d}. \tag{4.16}$$

Each of these four terms is linear with the time that the devices executing the corresponding operations:[5]

$$E_s^{down} = \sum_{n=1}^{N} c_n^{down} \frac{\sum_{k=1}^{K} x_{s,k\to n}^{down} L_k}{Q_n^{down}},$$

$$E_s^{cpu} = \sum_{n=1}^{N} c_n^{cpu} \frac{x_{s\to n}^{cpu} D_s^{cpu}}{Q_n^{cpu}}, E_s^{up} = \sum_{n=1}^{N} c_n^{up} \frac{\sum\limits_{k=1}^{K} x_{s,k\to n}^{up} L_k}{Q_n^{up}},$$

$$E_s^{d2d} = \sum_{i=1}^{N}\sum_{j=1}^{N} c_{i\to j}^{d2d} \frac{\sum\limits_{k=1}^{K} (z_{s,k\to i,j}^{in} + z_{s,k\to i,j}^{up} + z_{s,k\to i,j}^{ca}) L_k}{Q_{i\to j}^{d2d}}.$$

As an example, we explain task $s$' downloading energy $E_s^{down}$. It is the sum of the energies consumed by various devices for downloading for tasks $s$, where each device's downloading energy equals to the product of its energy coefficient and the downloading time.

### Problem Formulation

We want to minimize the energy consumption of the 3C framework under the proposed constraints. Formally, we want to solve the following optimization problem:

$$\begin{aligned} \text{minimize} \quad & \sum_{s\in\mathcal{S}} E_s \\ \text{subject to} \quad & (4.3) \sim (4.10) \\ & \boldsymbol{x}, \boldsymbol{z} \in \{0,1\} \end{aligned} \tag{OPT}$$

Note that such an energy minimization problem (OPT) is a typical example of showing the benefits of the 3C framework. The proposed mathematical formulation can also be applied to other system optimization objectives: for example, we could also minimize the worst delays subject to an energy consumption constraint together with other constraints.

[5] The linearity assumption has been adopted in the existing 1C/2C models considering energy consumption, e.g., [55, 85, 32, 34, 68].

Table 4.1: Existing models that the 3C framework can generalize.

| Shared Resource | Examples and Task Models $\boldsymbol{D}_s$ |
|---|---|
| Downloading | (a) User-provided networks (e.g., [55, 85])<br>$(u_s, \boldsymbol{D}_s^{data}, 0, \mathbf{0}, \boldsymbol{D}_s^{data})$ |
| Uploading | (b) Ad hoc content uploading (e.g., [68])<br>$(u_s, \boldsymbol{D}_s^{data}, 0, \boldsymbol{D}_s^{data}, \mathbf{0})$ |
| Content | (c) Ad hoc content sharing (e.g., [57, 33])<br>$(u_s, \boldsymbol{D}_s^{data}, 0, \mathbf{0}, \boldsymbol{D}_s^{data})$ |
| Computation | (d) Ad hoc computation offloading (e.g., [32, 34])<br>$(u_s, \boldsymbol{D}_s^{in}, D_s^{cpu}, \mathbf{0}, \boldsymbol{D}_s^{out})$ |
| Hybrid | (e) Distributed data analysis (e.g., [84, 38])<br>$(u_s, \boldsymbol{D}_s^{in}, D_s^{cpu}, \mathbf{0}, \boldsymbol{D}_s^{out})$ |

Problem (OPT) is non-convex due to the delay constraints. In Section 4.4, we transform Problem (OPT) into an ILP problem, which can be solved by standard optimizers. We further propose a heuristic algorithm with a lower computation complexity.

### 4.3.3 Generalization of Existing Models in the Literature

Through properly choosing various parameters, the proposed 3C framework can generalize many of the existing 1C and 2C models, as illustrated in Table 4.1. Among these models, the notation $\boldsymbol{D}_s^{data}$ (in (a), (b), and (c)) denotes the contents that are requested by the corresponding operations.

Figure 4.3 illustrates the distributed data analysis model (e) as a special case of our proposed 3C framework. Figure 4.3(a) corresponds to the model in [38]: two data source nodes S1 and S2 forward data to a computation node C for data analysis, then node C forwards the computation outputting to a destination D. Through specifying the task model as in Figure 4.3(b), our proposed model generalizes the model in 4.3(a), and can achieve the optimal

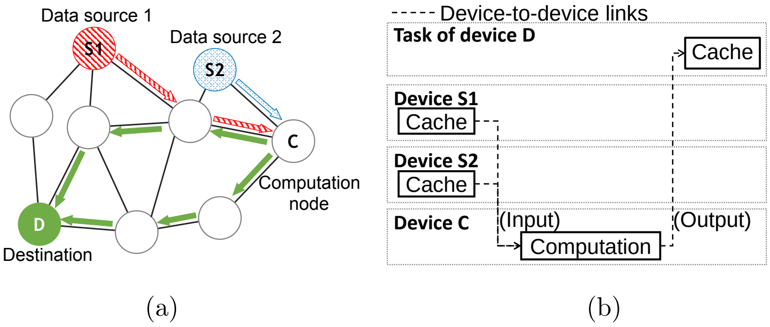


Figure 4.3: An example of distributed data analysis: (a) existing model [38]; (b) the generalization by the 3C framework.

resource allocation by solving Problem (OPT).

## 4.4 Energy Minimization with 3C Sharing

In this section, we focus on the energy minimization problem (OPT), which is an integer non-convex optimization problem. To solve this problem, we first transform it into an ILP problem, which can be solved by standard optimizers. However, an ILP problem is an NP-complete problem (Theorem 18.1 in [79]), so its computation time dramatically increases as the number of devices and tasks increases. Hence, we further propose a heuristic algorithm, which solves a series of problems, each of which is a LP relaxation (relaxing integer variables to continuous ones) of the original problem (OPT). This heuristic algorithm is guaranteed to produce an integer solution within the feasible region of the original problem (OPT), and has a lower computational complexity.

### 4.4.1 Linear Transformation of Problem (OPT)

We first transform the integer non-convex problem (OPT) to an ILP problem. The non-convexity is mainly due to the delay constraints, because the delays are formulated in quadratic forms that are not positive semidefinite. Hence, the key focus will be how to transform the delay constraints to linear ones, after which Problem (OPT) becomes an ILP.

The key transforming idea is as follows. Consider a constraint in the form of $\tau \times y \leq \bar{T}$, where the continuous variable $\tau \geq 0$, the discrete variable $y \in \{0, 1\}$, and the fixed parameter $\bar{T} \geq 0$. We can transform the constraint into an equivalent into a linear form $\tau - (1 - y) \times M \leq \bar{T}$, where $M$ is any number that satisfies $\tau - M \leq 0$. To see the equivalence of the two constraints, we consider two possible values of $y$. If $y = 1$, both the original and the transformed constraints are $\tau \leq \bar{T}$; if $y = 0$, both constraints are true for any value of $\tau$. Hence, the two constraints are equivalent.

Next we use this key idea to explain the transformation of downloading delay constraints. The transformations of computation and uploading delay constraints are similar. Finally, we present the transformed problem.

#### Transforming downloading delay constraints

We will transform the delay constraint $T_s^{down} \leq \bar{T}_s^{down}$ in (4.10) to a linear one, where $T_s^{down}$ is defined in (4.13). More specifically, this constraint can be written as

$$\tau_n^{down} y_{s \to n}^{down} \leq \bar{T}_s^{down}, \forall s \in \mathcal{S}, n \in \mathcal{N}, \tag{4.17}$$

where $\tau_n^{down}$ is a linear function of variables $x_{s,k \to n}^{down}$ as in (4.11). The variable $y_{s \to n}^{down} \in \{0, 1\}$ indicates whether a task $s$' downloading is being allocated to device $n$,[6] and satisfies the following conditions:

$$y_{s \to n}^{down} \leq \min\{\sum_{k=1}^{K} x_{s,k \to n}^{in}(1 - Q_{nk}^{ca}), 1\}, \tag{4.18}$$

$$y_{s \to n}^{down} \geq x_{s,k \to n}^{in}(1 - Q_{nk}^{ca}), \forall k \in \mathcal{K}. \tag{4.19}$$

[6] If $y_{s \to n}^{down} = 1$ (i.e., device $n$ downloads for task $s$), device $n$'s downloading time $\tau_n^{down}$ should satisfy the delay constraint, i.e., $\tau_n^{down} \leq \bar{T}_s^{down}$; if $y_{s \to n}^{down} = 0$, device $n$'s downloading time does not need to.

Specifically, when $x^{in}_{s,k\to n}(1-Q^{ca}_{nk})=0$ for all $k$ (i.e., device $n$ does not download any content for task $s$), we have $0\le y^{down}_{s\to n}\le 0$, i.e., $y^{down}_{s\to n}=0$; otherwise, when there exists a $k$ such that $x^{in}_{s,k:\to n}(1-Q^{ca}_{nk})=1$ (i.e., device $n$ downloads contents for task $s$), we have $0< x^{in}_{s,k:\to n}(1-Q^{ca}_{nk})\le y^{down}_{s\to n}\le 1$, i.e., $y^{down}_{s\to n}=1$.

Then, we transform constraints (4.17) based on the previously mentioned transforming idea. We will choose a parameter $M^{down}_n$ that satisfies $\tau^{down}_n - M^{down}_n \le 0$, and the constraint is given by

$$\tau^{down}_n-(1-y^{down}_{s\to n})M^{down}_n\le \bar{T}^{down}_s, \forall s\in\mathcal{S}, n\in\mathcal{N}, \tag{4.20}$$

where $\tau^{down}_n\ge 0$ is a linear function of $x^{down}_{s,k\to n}$ as in (4.11).

#### Transforming computation and uploading delay constraints

Based on similar ideas, for the computation delay constraint, we will choose a parameter $M^{cpu}_n$ that satisfies $\tau^{cpu}_n-M^{cpu}_n\le 0$, and the equivalent constraint is given by

$$\tau^{cpu}_n-(1-x^{cpu}_{s\to n})\cdot M^{cpu}_n\le \bar{T}^{cpu}_s, \forall s\in\mathcal{S}, n\in\mathcal{N}, \tag{4.21}$$

where $x^{cpu}_{s\to n}$ denotes whether device $n$ computes for task $s$.

For the uploading delay constraint, we will choose a parameter $M^{up}_n$ that satisfies $\tau^{up}_n-M^{up}_n\le 0$, and the corresponding equivalent constraint is given by

$$\tau^{up}_n-(1-y^{up}_{s\to n})\cdot M^{up}_n\le \bar{T}^{up}_s, \forall s\in\mathcal{S}, n\in\mathcal{N}, \tag{4.22}$$

where $y^{cpu}_{s\to n}$ denotes whether device $n$ uploads for task $s$:

$$y^{up}_{s\to n}\le \min\left\{\sum_{k=1}^{K} x^{up}_{s,k\to n}, 1\right\}, \tag{4.23}$$

$$y^{up}_{s\to n}\ge x^{up}_{s,k\to n}, \forall k\in\mathcal{K}. \tag{4.24}$$

**The Linear Transformation of Problem (OPT)**

Once replacing the delay constraint (4.10) with (4.18)∼(4.24), we transform Problem (OPT) into the following problem:

$$\begin{aligned} \underset{\boldsymbol{x},\boldsymbol{z},\boldsymbol{y}}{\text{minimize}} \quad & \sum_{s\in\mathcal{S}} E_s \\ \text{subject to} \quad & (4.3) \sim (4.9), (4.18) \sim (4.24) \\ \text{variables} \quad & \boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y} \in \{0,1\} \end{aligned} \qquad \text{(OPT-LINEAR)}$$

Problem (OPT-LINEAR) is an ILP, which can be solved by standard optimizers, e.g., Gurobi (`http://www.gurobi.com`).

We want to emphasize again that the transformed delay constraints (4.18)∼(4.24) are equivalent to the original constraints (4.10) if the newly introduced parameters are large enough. This is captured by the following Lemma 4.1.

**Lemma 4.1.** *Problem (OPT-LINEAR) is equivalent to Problem (OPT) if the following is true for all $n \in \mathcal{N}$:*

$$M_n^{down} \geq \tau_n^{down}, M_n^{cpu} \geq \tau_n^{cpu}, M_n^{cpu} \geq \tau_n^{up}. \tag{4.25}$$

Note that $\tau_n^{down}$, $\tau_n^{cpu}$, and $\tau_n^{up}$ are functions of decision variables, which are unknown before solving the optimization problem. To ensure that (4.25) holds, we can set the parameters $M_n^{down}$, $M_n^{cpu}$, and $M_n^{up}$ to be the maximum delay bounds (i.e.,, $\max\{\bar{T}_s^{down}, \forall s\}$, $\max\{\bar{T}_s^{cpu}, \forall s\}$, and $\max\{\bar{T}_s^{up}, \forall s\}$, respectively, for any $n \in \mathcal{N}$). (i.e.,$\tau_n^{down}$, $\tau_n^{cpu}$, and $\tau_n^{up}$) cannot exceed them.

Solving Problem (OPT-LINEAR) using standard optimizers works well when the network size (e.g., the number of devices and tasks) is reasonably small.[7] However, as the system size increases, the corresponding computation time dramatically increases, because Problem (OPT-LINEAR) (an ILP) is an NP-complete problem [79]. To address this complexity issue, we propose a heuristic algorithm based on the original Problem (OPT).

[7] In simulations in Section 4.6, directly solving Problem (OPT-LINEAR) has a low computation time when the device number is smaller than 20.

### 4.4.2 A Heuristic Algorithm of of Solving Problem (OPT)

The key idea is to iteratively solve a series of modified versions of Problem (OPT), where we remove the delay constraints and relax the integer variables to continuous ones (i.e., LP relaxation [79], so that the modified problems are LP problems). At the end of each iteration, the algorithm will check whether the removed delay constraints are satisfied. If not, the algorithm will prevent some tasks from being allocated to certain devices (in order to address the violated delay constraints), and solve a new version of modified problem. The algorithm iterates until all the delay constraints are satisfied. Note that we do not check whether the variables satisfy the integer constraints in the algorithm. Later we will show that, however, the algorithm is guaranteed to produce integer solutions that are feasible for Problem (OPT).

Next we first describe the modified problem, then we propose the heuristic algorithm.

#### A Modified Problem of Problem (OPT)

Comparing with the original Problem (OPT), the modification involves removing delay constraints, relaxing integer variables, and adding control parameters that can prevent certain tasks from being allocated to certain devices. We first introduce the control parameters, then propose the modified problem.

In order to prevent particular subtasks from being allocated to certain devices, we introduce the following binary control parameters $\tilde{N}^{in}_{s,k\to n}$, $\tilde{N}^{cpu}_{s\to n}$, and $\tilde{N}^{up}_{s,k\to n}$:

$$x^{in}_{s,k\to n} \leq \tilde{N}^{in}_{s,k\to n},\ x^{cpu}_{s\to n} \leq \tilde{N}^{cpu}_{s\to n},\ x^{up}_{s,k\to n} \leq \tilde{N}^{up}_{s,k\to n}. \tag{4.26}$$

Take $\tilde{N}^{in}_{s,k\to n}$ as an example: if it equals zero, then (4.26) indicates that $x^{in}_{s,k\to n}$ can only be zero, so the content $k$ of task $s$ cannot be allocated to device $n$;

if it equals one, then $x^{in}_{s,k\to n}$ can be either zero or one, so the allocation is not prevented. The same idea applies to $\tilde{N}^{cpu}_{s\to n}$ and $\tilde{N}^{up}_{s,k\to n}$.

Then, we introduce the modified problem of Problem (OPT) by removing delay constraints (4.10), relaxing integer variables (i.e., relaxing $\boldsymbol{x}, \boldsymbol{z} \in \{0,1\}$ to be $\boldsymbol{x}, \boldsymbol{z} \in [0,1]$), and adding control constraints (4.26):

$$\begin{aligned} &\underset{\boldsymbol{x},\boldsymbol{z}}{\text{minimize}} && \sum_{s\in\mathcal{S}} E_s \\ &\text{subject to} && (4.3) \sim (4.9), (4.26) \\ &\text{variables} && \boldsymbol{x}, \boldsymbol{z} \in [0,1] \end{aligned} \tag{OPT-RELAX}$$

To clarify, there are many versions of Problem (OPT-RELAX), each of which corresponds to a set of parameter choices of $\tilde{N}^{in}_{s,k\to n}$, $\tilde{N}^{cpu}_{s\to n}$, and $\tilde{N}^{up}_{s,k\to n}$. Moreover, Problem (OPT-RELAX) is an LP problem, which can be solved using various methods such as Simplex method [35].

### A Heuristic Algorithm to solve Problem (OPT)

The heuristic algorithm will iteratively solve multiple versions of Problem (OPT-RELAX) as follows. At the beginning, no allocation is prevented, i.e., $\tilde{N}^{in}_{s,k\to n} = \tilde{N}^{cpu}_{s\to n} = \tilde{N}^{up}_{s,k\to n} = 1$ for all $s$, $k$, and $n$. We first optimize the corresponding Problem (OPT-RELAX) (e.g., using Simplex method [35]), and check whether the optimal solution satisfies the delay constraints in (4.10). If yes, then the obtained optimal solution of solving Problem (OPT-RELAX) is the optimal solution of Problem (OPT); if not, we need to revise the control parameters in Problem (OPT-RELAX) (i.e., setting some $\tilde{N}^{in}_{s,k\to n}$, $\tilde{N}^{cpu}_{s\to n}$, or $\tilde{N}^{up}_{s,k\to n}$ to be zeros), with details discussed in the next paragraph. We optimize Problem (OPT-RELAX) iteratively until obtaining a solution that satisfies the delay constraints in (4.10). The algorithm is given in Algorithm 2.

We now discuss how the algorithm chooses the proper version of Problem

(OPT-RELAX) to solve by setting $\tilde{N}^{in}_{s,k\to n}$, $\tilde{N}^{cpu}_{s\to n}$, or $\tilde{N}^{up}_{s,k\to n}$ for inputting (in lines 6-12 of Algorithm 2),[8] computation (in lines 13-17), and uploading (in lines 18-22) subtasks, respectively. We first introduce the basic idea, then explain the special settings of the inputting subtask.

The basic idea of preventing some allocations for the inputting, computation, and uploading subtasks is as follows. For a particular subtask of a task $s$, if its corresponding delay (after solving a version of Problem (OPT-RELAX)) is larger than the delay bound, then the algorithm will (i) find the device $\hat{n}$ that induces the maximum delay, (ii) at device $\hat{n}$, find the task $\bar{s}$ with the tightest delay bound among all the tasks that are allocated to device $\hat{n}$ (excluding device $\hat{n}$'s own tasks),[9] (iii) prevent task $\bar{s}$ from being allocated to device $\hat{n}$ by setting the corresponding parameters $\tilde{N}^{in}_{s,k\to n}$, $\tilde{N}^{cpu}_{s\to n}$, or $\tilde{N}^{up}_{s,k\to n}$ to be zero.

Next we discuss the special settings of the inputting (downloading) subtask. Specifically, device $\hat{n}$ may download the same content for both itself and other devices, but it should not prevent the content downloading of its own tasks. So we define a *non-preventable* set $\boldsymbol{I}$ (in line 8), which contains the tasks that request a same content as device $\hat{n}$ does. Only the tasks outside the non-preventable set can be prevented (in line 9). In addition, only the contents that have not be cached (have to be downloaded) are prevented (in lines 10 and 11), because cached contents do not induce delays.

### Properties of Algorithm 2

We first make an assumption that is often satisfied in practice, and then characterize several properties of the proposed heuristic algorithm, including its

[8] The inputting subtask prevention corresponds to downloading delays, because the downloading is the operation inducing inputting delays.

[9] This is used to ensure that the heuristic algorithm always has an output, which is the noncooperation case (where each device performs its tasks on its own). This will be discussed in the next subsection.

guaranteed output of a feasible solution output, the performance guarantee, and the computation complexity.

**Assumption 4.1** (Feasible Noncooperation Case)**.** *Noncooperation (i.e., each of the devices performs its tasks on its own) is within the feasible region of Problem (OPT).*

This assumption implies that each device is capable of executing its tasks on its own. If Assumption 2 is violated, some tasks may become infeasible to complete, as cooperation is not always guaranteed in practice.

Under this assumption, Algorithm 2 can guarantee to output a feasible solution of Problem (OPT).

**Proposition 4.1** (Guarantee of Feasible Output)**.** *Algorithm 2 is guaranteed to produce an integer solution that is within the feasible region of Problem (OPT).*

The proof is given in Appendix 4.8.1 . The key idea is that the optimal solution of Problem (OPT-RELAX) is always integer, even though there is no integer constraint in that problem. To emphasize, the key part that we relax (in the heuristic algorithm) is the integer variables, where the relaxation is used to reduce the computation time (transforming an ILP problem (OPT) to an LP problem (OPT-RELAX)). However, such LP relaxation does not affect the integer output of the heuristic algorithm, as we can prove that solving (OPT-RELAX) with $\boldsymbol{x}, \boldsymbol{z} \in [0, 1]$ (e.g., using Simplex method) is guarantee to output an integer solution, i.e., $\boldsymbol{x}, \boldsymbol{z} \in \{0, 1\}$.

We now show the performance guarantee of Algorithm 2.

**Proposition 4.2** (Performance Guarantee)**.** *The energy consumption of the heuristic algorithm output is no larger than that of the noncooperation case. When there is no delay constraint, the heuristic algorithm output is an optimal solution of the original problem (OPT).*

The proof is given in Appendix 4.8.2. Specifically, when there is no delay constraint, we can show that the optimal solution of (OPT-RELAX) is the optimal solution of (OPT). As a result, the output of the heuristic algorithm is optimal. We will evaluate the performance of this heuristic algorithm under the settings with delay constraints in Section 4.6.2.

Regarding the complexity of Algorithm 2, its maximum iteration time is as follows:

**Proposition 4.3** (Maximum Iteration Time)**.** *The maximum iteration time of this heuristic algorithm is* $S \times (N-1)$*, where* $S$ *is the task number and* $N$ *is device number.*

The proof is given in Appendix 4.8.3. Recall that Problem (OPT) is NP-complete. In comparison, the heuristic algorithm has a maximum of $S \times (N-1)$ iterations, each of which solves an LP problem (OPT-RELAX) that is a P-complete problem. This implies that the heuristic algorithm can terminate in polynomial time, while solving Problem (OPT) cannot be done in polynomial time in general.

## 4.5 Energy Reduction Due to 3C Sharing

The proposed 3C framework is "resource-centric" instead of "task-centric", so that it provides additional flexibilities in terms of device cooperation. More specifically, it promotes cooperation opportunities through enabling devices performing different tasks to cooperate. In this section, we study how much a 3C framework can reduce the energy consumption through a specific problem setting, comparing with 1C models. We first introduce system settings, then discuss the energy reduction due to the 3C framework.

### 4.5.1 System Settings

In order to derive the closed-form solutions of the energy reduction, we consider specific device and task models as follows. We consider a random graph model $G(N,p)$ [20], where there are $N$ devices in the graph and every two devices are connected randomly and independently with a probability $p$. Suppose that the network is large and sparse, so that $N$ approaches infinite with $Np$ being a constant.[10] These devices initialize a set of tasks. Since we focus on the comparison between 1C models and 3C framework, we assume that each task only needs one of the 3C resources.

The devices are heterogeneous in terms of their owned resources. Specifically, each device $n$ owns some resources $Q_n^{down}$, $Q_n^{cpu}$, and $Q_n^{up}$. The capacities $Q_n^X$ ($X \in \{down, cpu, up\}$) is independent and identically distributed (i.i.d.) with the cumulative distribution function (cdf) $F_Q^X(x)$ and the probability density function (pdf) $f_Q^X(x)$. The support of capacity $Q_n^X$ is $(\underline{Q}^X, \overline{Q}^X)$, hence $F_Q^X(\underline{Q}^X) = 0$ and $F_Q^X(\overline{Q}^X) = 1$. For the convenience of analysis, we assume that the energy coefficients of the devices are homogeneous, i.e., $c_n^X = c^X, X \in \{down, cpu, up\}$, $\forall n$. In addition, each device $n$ uniformly and randomly caches $M^{ca}$ contents in its cache,[11] i.e., $\sum_{k=1}^{K} Q_{nk}^{ca} = M^{ca}$ for all $n$. For the convenience of analysis, we assume that all the contents have the same size that is normalized to one, i.e., $L_k = 1, \forall k$.

We aim to study the system under the general distribution function, which is quite challenging to do. Hence we further make the following simplifying assumption for the rest of Section 4.5. These assumptions lead to a larger energy reduction that the framework can achieve (when considering single-hop cooperations), comparing with relaxing these assumptions. A more realistic

[10] This assumption is consistent with the phenomenon that most real networks are sparse, i.e., the number of devices that a device may connect with is significantly smaller than the total number of devices [20].

[11] The uniformly and randomly caching is a widely used benchmark in proactive caching [21], leading to a performance that is no better than than using optimal caching strategies.

case (with these assumptions relaxed) is evaluated empirically in Section 4.6.

**Assumption 4.2.** *1) the D2D transmission energy is relatively small and can be ignored; 2) there is no delay constraint; 3) devices can only cooperate with their one-hop neighbors.*

Under Assumption 4.2, for any device $m$, the optimal allocation of any of its tasks $s \in \boldsymbol{s}_m$ is as follows. Regarding the inputting subtask, for a content requested by task $s$, if any device $n \in \mathcal{E}(m)$ has cached it, then the content will be inputted from device $n$. If none of the devices in set $\mathcal{E}(m)$ has cached it, then the device who has the highest downloading capacity among set $\mathcal{E}(m)$ downloads the content. Regarding the computation and uploading subtasks, they will be allocated to the devices with the highest computation and uploading capacities among the devices $\mathcal{E}(m)$, respectively.

### 4.5.2 Energy Reduction Due to the 3C Framework

In this subsection, we study how much a 3C framework can reduce the energy consumption through providing more cooperation opportunities.

We explain the basic analysis idea using the following simple example. Suppose among the entire device population, $\alpha N$ devices initialize downloading tasks and participate in user-provided network, and another $\alpha N$ devices initialize computation tasks and participate in ad hoc computation offloading, where $\alpha$ refers to a fraction of devices, and we assume that these two set of devices do not overlap with each other. Under the 1C models (e.g., [55] and [32]), only devices with the same type of tasks (hence requesting the same type of resource) can cooperate; in our proposed 3C framework, all $2\alpha N$ devices can cooperate with each other, so that the number of devices sharing each of the downloading and computation resources is doubled, respectively. We are interested in calculating the energy gap between these two cooperative scenarios, showing the energy reduction due to 3C framework. To clarify, the above scenario is only a simple example, our analysis will cover not only the communication and computation resources but also the caching resource.

Since that each of the tasks only requests one kinds of the 3C resources, we can analyze the tasks requesting each of the 3C resources separately. Specifically, as in the above example, we can calculate the energy reduction of the tasks requesting downloading resource and the tasks requesting computation resource separately, and the entire energy reduction will be the sum of the two energy reductions.

Next we will compute the energy reduction of the tasks requesting each of the 3C resources one by one. We will first discuss the tasks requesting communication/computation resource (both of which are capacity-based resources), and then discuss the tasks requesting caching resource.

#### Communication/Computation

In the following analysis, we focus on the tasks requesting a particular resource (i.e., downloading, uploading, or computation). Hence, for presentation simplicity, the term "resource" in Section 4.5.2 only refers to the particular resource, and we omit the corresponding resource-specific super-scripts and sub-scripts. Without the loss of generality, we normalized the energy coefficient of the particular resource to be one, i.e., $c = 1$.

We will first formulate the expected energy consumption, then introduce the analysis idea. In the random graph $G(N, p)$, suppose each device joins the cooperative system with a probability $\alpha \in [0, 1]$. Under these, each task requesting the particular resource will have an expected energy consumption denoted by $W(\alpha, Np)$.[12] Under the 1C model, let us denote the probability that each device joins the corresponding 1C model (that shares the particular resource) as $\alpha^{1C} \in [0, 1]$; under the 3C framework, the corresponding

[12] Under the homogeneous distribution settings in Section 4.5.1, all the tasks will have the same expected energy consumption, so we only need to study the expected energy consumption of a task.

probability is $\alpha^{3C} = \min\{r\alpha^{1C}, 1\}$, where $r \geq 1$ is a coefficient reflecting the ratio of the increased cooperation opportunities. We will compute the energy reduction $\Delta W(r, \alpha^{1C}, Np) \triangleq W(\alpha^{1C}, Np) - W(\alpha^{3C}, Np)$ for $r \geq 1$.

First, we calculate the expected energy, i.e., $W(\alpha, Np)$, under particular $\alpha$ and $Np$. As we have explained, under Assumption 4.2, any device's task will be allocated to its neighbor who has the highest capacity. By using the order statistic result [15], the probability density function of the highest capacity among total $n$ devices is given by

$$f_{(n)}(x) = n(F(x))^{n-1} f(x). \tag{4.27}$$

In the random graph, for a device with a degree $m$, the probability that $\hat{N}$ of its neighbors join in the cooperative system is $P(\hat{N}|m) = C_m^{\hat{N}} \alpha^{\hat{N}} (1-\alpha)^{m-\hat{N}}$, and the corresponding distribution of the highest capacity among these $\hat{N}$ devices and itself is $f_{(\hat{N+1})}(x)$. Taking the expectation over $\hat{N} = \{0, ..., m\}$, the expected energy consumption of this device's task is given by

$$\hat{W}_m(\alpha, Np) = \sum_{\hat{N}=0}^{m} P(\hat{N}|m) \int_{\underline{Q}}^{\overline{Q}} \frac{1}{x} f_{(\hat{N}+1)}(x) dx. \tag{4.28}$$

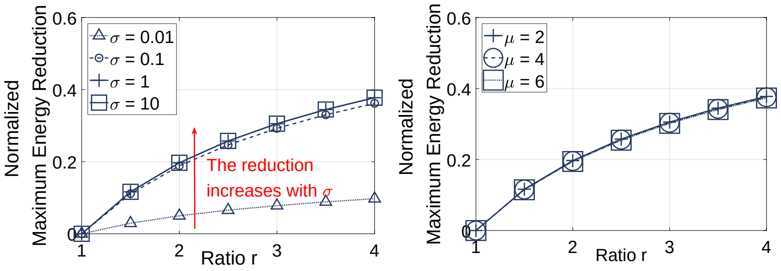
Taking the expectation of $\hat{W}_m(\alpha, Np)$ over all degrees $m = \{0, ..., \infty\}$ [20], the expected energy of a task is

$$W(\alpha, Np) = \frac{1}{\overline{Q}} + \int_{\underline{Q}}^{\overline{Q}} e^{Np(F(x)-1)\alpha} F(x) x^{-2} dx, \tag{4.29}$$

with the detailed proof given in Appendix 4.8.4.

Then, we discuss how much the 3C framework can reduce the energy consumption under a coefficient $r \geq 1$. We are interested in the best (the maximum energy reduction) that the 3C framework can achieve for any $\alpha^{1C}$ and $p$ under an $r$, i.e., $\max_{\alpha^{1C},p} \Delta W(r, \alpha^{1C}, Np)$.[13] The maximum energy reduction

[13] It is less interesting to discuss the worst case (the minimum energy reduction). This is because under the worst case, the energy reduction is always zero, i.e., $\min_{\alpha^{1C},p} \Delta W(r, \alpha^{1C}, Np) = 0$, which is achieved under $p = 0$, $p = 1$, $\alpha^{1C} = 0$, or $\alpha^{1C} = 1$ (where either noncooperation or full cooperation occurs in both the 1C and 3C approaches).



(a) (b)

Figure 4.4: Maximum energy reduction normalized by the energy consumption in noncooperation: (a) $\mu = 2$; (b) $\sigma = 10$.

caused by the 3C framework is as follows, with a proof provided in Appendix 4.8.5:

**Theorem 4.1** (Maximum Energy Reduction of Communication/Computation)**.** *Under a coefficient $r \geq 1$, the maximum energy reduction due to the 3C framework is given by*

$$\max_{\alpha^{1C},p} \Delta W(r, \alpha^{1C}, p) = \int_{\underline{Q}}^{\overline{Q}} \left( e^{\frac{N\tilde{p}(F(x)-1)}{r}} - e^{N\tilde{p}(F(x)-1)} \right) F(x) x^{-2} dx, \tag{4.30}$$

*where $\tilde{p}$ satisfies*

$$\int_{\underline{Q}}^{\overline{Q}} (F(x)-1) F(x) \left( e^{\frac{N\tilde{p}(F(x)-1)}{r}} - r e^{N\tilde{p}(F(x)-1)} \right) dx = 0. \tag{4.31}$$

Such a maximum energy reduction depends on the capacity distribution $F(x)$. Next, we show a concrete example.

**Example 4.1.** *Let us consider the truncated normal distribution, denoted by $F(x) = F(x; \mu, \sigma, a, b)$, which can be regarded as a normal distribution $N(\mu, \sigma^2)$ that lies within the interval $[a, b]$ (please refer to Section 4.6.1 and [26] for details). Figure 4.4 shows the maximum energy reduction (normalized by the energy consumption in the noncooperation case, $W(0,0)$). From Figure 4.4, we conclude that (i) the energy reduction is higher when the variance $\sigma$ is larger (i.e., devices are more heterogeneous), (ii) the energy reduction is* *not affected by the mean $\mu$, and (iii) under a large variance $\sigma$ (e.g., $\sigma = 10$, under which $F(x; \mu, \sigma, a, b)$ approaches to a uniform distribution), doubling the sharing devices fraction (i.e., $r = 2$) leads to a maximum energy reduction of around 20% of the energy consumed in noncooperation.*

**Caching**

The analysis for caching is similar as that for the communication/computation, where the details are given in Appendix 4.8.6. The expected energy reduction of a content $Z(\alpha, Np)$ is as follows:[14]

$$Z(\alpha, Np) = (1 - \frac{M^{ca}}{K})e^{-\alpha Np \frac{M^{ca}}{K}}. \tag{4.32}$$

Under this, the energy reduction due to 3C framework is $\Delta Z(r, \alpha^{1C}, Np) \triangleq Z(\alpha^{1C}, Np) - Z(\alpha^{3C}, Np)$. Then, the maximum energy reduction is given as follows.

**Theorem 4.2** (Maximum Energy Reduction of Caching)**.** *Under a coefficient $r \geq 1$, the maximum energy reduction due to the 3C framework is given by*

$$\max_{\alpha^{1C}, p} \Delta Z(r, \alpha^{1C}, Np) = \left(1 - \frac{M^{ca}}{K}\right)\left(e^{-\frac{\ln r}{(r-1)}} - e^{-\frac{r \ln r}{(r-1)}}\right). \tag{4.33}$$

The proof is given in Appendix 4.8.7. For a better understanding of (4.33), we normalize such a maximum energy reduction with respect to the energy consumption in the noncooperation case (i.e., $Z(0, 0)$). The normalized energy reduction is given by $e^{-\ln r/(r-1)} - e^{-r \ln r/(r-1)}$, which has a similar increasing concave shape as the curves in the two subfigures in Figure 4.3. Based on this, we conclude that (i) the normalized maximum energy reduction is independent of the caching ratio $M^{ca}/K$, as long as $M^{ca}/K > 0$. This means that no matter how many contents that devices have cached, the maximum

[14] Similarly, under the homogeneous distribution settings in Section 4.5.1, each content will have the same expected energy consumption, so we only need to study the expected energy consumption of a content.

normalized energy reduction is fixed; (ii) doubling the sharing devices fraction (i.e., $r = 2$) leads to a maximum energy reduction of around 25% of the energy consumed in noncooperation.

## 4.6 Simulation and Performance

We compare the computation time and energy consumption between optimal and heuristic solutions. And we evaluate the energy reduction due to 3C framework under different D2D transmission energies and different devices' and tasks' heterogeneities. To emphasize, these simulations are based on a more realistic case, where Assumption 4.2 is relaxed.

### 4.6.1 Simulation Setting

We consider a set of $N$ devices, who form pair-wise connections with probability $p = 0.3$. Each device has one task to execute. For each experiment, we perform 100 times and show the average results.

In each time of an experiment, we randomly generate the parameters of the devices' capacities, tasks' demands, and energy consumption coefficients. These parameters follows truncated normal distribution [26], with an identical variance $\sigma$ (which will be evaluated later) and different means. The truncated normal distribution can be regarded as a normal distribution but lies within range $(a, b)$, i.e., for $x \in (a, b)$,

$$f(x; \mu, \sigma, a, b) = \frac{\phi(\frac{x-\mu}{\sigma})}{\sigma\left(\Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right)\right)}, \tag{4.34}$$

where functions $\phi(\xi)$ and $\Phi(\xi)$ are given by

$$\phi(\xi) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\xi^2}, \tag{4.35}$$

$$\Phi(\xi) = \frac{1}{2}\left(1 + \frac{2}{\sqrt{\pi}} \int_0^{\frac{\xi}{\sqrt{2}}} e^{-t^2} dt\right). \tag{4.36}$$

Table 4.2: Parameter settings in experiments.

| Parameter | $(a, b)$ | Parameter | $(a, b)$ | Parameter | $(a, b)$ |
|---|---|---|---|---|---|
| $Q^{down}$ | $(0, 10)$ | $Q^{up}$ | $(0, 4)$ | $Q^{d2d}$ | $(0, 50)$ |
| $Q^{cpu}$ | $(0, 10)$ | $D^{cpu}$ | $(0, 10)$ | $c^{down}$ | $(0, 2.8)$ |
| $c^{cpu}$ | $(0, 1.2)$ | $c^{up}$ | $(0, 2.8)$ | $c^{d2d}$ | $(0, 0.8)$ |

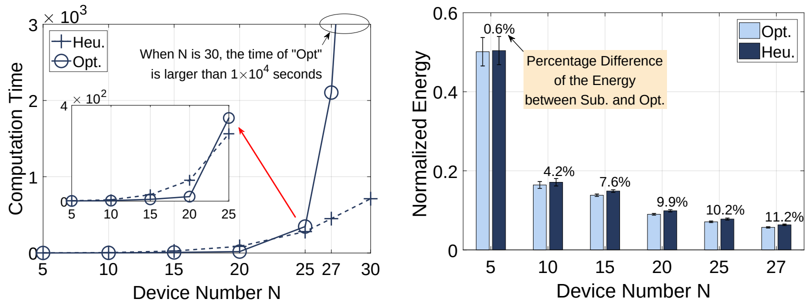
The distribution range $(a, b)$ of the parameters are shown in Table 4.2. The relative values of the maximum downloading, uploading, and D2D transmission capacities are based on references [11] and [8], and the relative values of the maximum energy per time are based on paper [73] and [19]. We pick the same value for the maximum computation capacity and demand, so that performing a computation subtask is one second on average. In addition, for each of these parameters, we set $\mu = (a + b)/2$ and $\sigma = 1.0$ (if not specified), under which the distribution is similar to a uniform distribution within $(a, b)$.

In addition, in each time of an experiment, we randomly generate the parameter of the contents involved. The content sizes are set to be one. Each device caches a random number of contents in its local cache and requests a random number of content input and content output. The uploading contents and caching contents are randomly selected from the content output.

### 4.6.2 Comparison: Optimal and Heuristic Solutions

We show how the device number $N$ affects the computation time and energy consumption of the optimal (named as "Opt.") and the heuristic algorithm (named as "Heu.").

Figure 4.5 (a) shows how the total computation time changes in $N$. For the case of "Opt." (solving Problem (OPT-LINEAR)), the computation time is small when $N$ is small (e.g., $N$ is less than 20). However, as the device number increases, the computation time of "Opt." dramatically increases (i.e., reaches



(a)

(b)

Figure 4.5: Impact of $N$ on (a) computation time and (b) energy consumption.

more than $2,000$ seconds when $N = 27$). In comparison, the computation time of "Heu." increases relatively slowly in $N$, i.e., the computation time of "Heu." is 78.6% smaller than that of "Opt." when $N = 27$.[15]

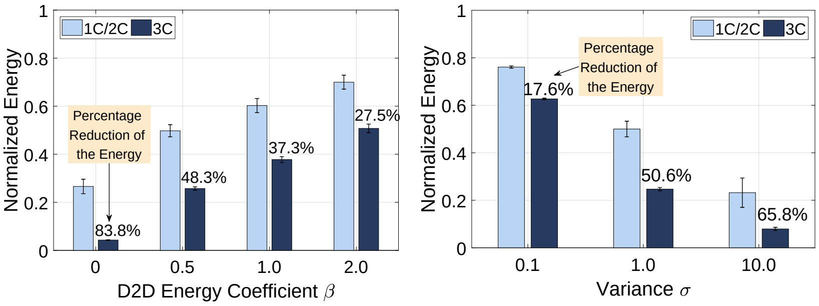
Figure 4.5 (b) shows the energy comparison between "Opt." and "Heu.". The energy is normalized by the energy consumed in the noncooperation case (i.e., each device executes its task by itself).[16] As $N$ increases, the energy gap between "Opt." and "Heu." slightly increases. When $N = 27$, the normalized percentage difference of the energy between "Heu." and "Opt." is only around 11.2%.

### 4.6.3 Comparison: 1C/2C Models and 3C Framework

In this simulation, we let each device randomly selects a task among downloading, content sharing, and distributed data analysis. Then, we perform simulations in two cooperation settings: (i) "1C/2C", where only the devices selecting the same kinds of tasks can cooperate; (ii) "3C", where all the devices can cooperate.

[15] When $N$ is smaller than 20, the computation time of "Opt." is smaller than that of "Heu.". This is because executing "Opt." requires solving one ILP problem, while executing "Heu." requires iteratively solving multiple LP problems. Hence, implementing "Heu." is beneficial only when $N$ is large, under which the NP-complete problem (OPT-LINEAR) (of executing "Opt.") induces extreme large computation time.

[16] We skip the simulation of $N = 30$ due to its huge computation time.



(a) (b)

Figure 4.6: Impact of D2D energy $\beta$ (under $\sigma = 1.0$) and variance $\sigma$ (under $\beta = 0.5$) on the normalized energy consumption.

We compare the energies of the two cooperation settings under different D2D energy coefficients $\beta$ (the ratio of the D2D energy per unit time to the downloading energy per unit time) and variances $\sigma$ (the variance for generating tasks and devices). These energy consumptions are normalized by the energy in the noncooperation case. The percentage reduction in the figure is the energy difference between “1C/2C” and “3C”, normalized by the energy consumed in “1C/2C”.

In Figure 4.6 (a), “3C” can reduce the energy consumption by 83.8% when $\beta = 0$, i.e., no additional energy consumption due to D2D transmissions. Such a energy reduction decreases in $\beta$, but still achieves a value of 27.5% when $\beta = 2.0$, i.e., the D2D energy per unit time is twice as large as the downloading energy per unit time.

In Figure 4.6 (b), as the heterogeneity of devices and tasks (measured by the variance $\sigma$) increases, the energy reduction caused by 3C framework increases, which is consistent with the results in Example 4.1 in Section 4.5. Intuitively, a higher heterogeneity can provide more opportunities for the devices to share resources and help each other. Hence, implementing the 3C framework is more beneficial when devices and tasks are more heterogeneous.

## 4.7 Chapter Summary

In this chapter, we propose a general 3C framework that enables the joint 3C resource sharing among mobile devices, which potentially enhances the QoE of mobile multimedia services. This "resource-centric" framework generalizes existing D2D resource sharing models, and provides a structure for future D2D resource sharing analysis. We theoretically and empirically show that the 3C framework can further exploit resource sharing potentials and improve resource utilization efficiency significantly.

## 4.8 Appendix

### 4.8.1 Proof for Proposition 4.1

To prove Proposition 4.1, we first present a lemma showing that all the extreme points of the feasible region in Problem (OPT-RELAX) are integers. Then, we present a lemma showing that solving Problem (OPT-RELAX) using Simplex method [35] is guaranteed to output an integer solution. Finally, we explain that Algorithm 2 is guaranteed to produce an integer solution that is within the feasible region of Problem (OPT).

**Lemma 4.2** (Integer Extreme Points)**.** *All the extreme points of the feasible region in Problem (OPT-RELAX) are integers.*

*Proof.* To prove this lemma, we first transform Problem (OPT-RELAX) into an equivalent problem. Then, we prove that the equivalent problem has only integer extreme points.

First, we substitute constraints (4.7) $\sim$ (4.9) into the objective function, leading to an equivalent optimization problem in the following form:

$$\begin{aligned}\text{minimize}\quad & \boldsymbol{h}_1^{\mathrm{T}}\boldsymbol{x}+\boldsymbol{h}_2^{\mathrm{T}}\boldsymbol{z}\\ \text{subject to}\quad & \boldsymbol{a}\le[\boldsymbol{A}\ \boldsymbol{0}]\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{z}\end{bmatrix}\le\boldsymbol{b}\\ & \boldsymbol{c}\le\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{z}\end{bmatrix}\le\boldsymbol{d}\\ \text{variables}\quad & \boldsymbol{x},\boldsymbol{z}\in[0,1]\end{aligned}\tag{4.37}$$

The first constraint corresponds to (4.3) $\sim$ (4.6), and the second constraint corresponds to (4.26).

Then, we show that the problem (4.37) has only integral extreme points (so as Problem (OPT-RELAX)). The idea is to show that $\boldsymbol{a},\boldsymbol{b},\boldsymbol{c},\boldsymbol{d}$ contain only integers and $[\boldsymbol{A}\ \boldsymbol{0}]$ is totally unimodular [80]. Through checking the constraints, we directly have $\boldsymbol{a},\boldsymbol{b},\boldsymbol{c},\boldsymbol{d}$ contain only integers. According to [48], matrix $[\boldsymbol{A}\ \boldsymbol{0}]$ is totally unimodular, because (i) all the elements belong to $\{+1,-1,0\}$, (ii) each column of the matrix contains at most two non-zero elements (only $x_{s,k\to n}^{in}$ has two non-zero elements from constraints (4.4) and (4.6), while the others only have at most one), (iii) for the only variables with two non-zero elements $x_{s,k\to n}^{in}$ from constraints (4.4) and (4.6), the two non-zero elements have the same sign and can be separated into two disjoint sets (of rows) by separating constraints (4.4) and (4.6) into two subsets of $\boldsymbol{x}$ by rows. □

**Lemma 4.3** (Integer Output of Problem (OPT-RELAX))**.** *If solving Problem (OPT-RELAX) using Simplex method [35], the output solution is an integer solution.*

*Proof.* Problem (OPT-RELAX) is an LP problem that has bounded variables and is feasible, where the feasibility can be proved by showing that noncooperation is always a feasible point of Problem (OPT-RELAX) (as assumed in

Assumption 4.1). So there always exists a vertex that is the optimal solution of Problem (OPT-RELAX) [80]. One the other hand, Simplex method solves LP by traversing the edges between vertexes on the feasible region, such that the output solution is always a vertex. Hence, the output solution (using Simplex method to solve Problem (OPT-RELAX)) is an integer solution. □

Finally, we explain that Algorithm 2 is guaranteed to produce an integer solution that is within the feasible region of Problem (OPT). There will be two claims: first, Algorithm 2 always has an output; second, if Algorithm 2 has an output, the output is a feasible solution of Problem (OPT). The first claim is directly held under Assumption 4.1. The second claim is true because (i) the output is an integer solution satisfying constraints (4.3) $\sim$ (4.9) (according to Lemma 4.3), and (ii) the output satisfies the delay constraint (4.10).

### 4.8.2 Proof for Proposition 4.2

We prove the two claims in this proposition one by one.

First, the energy consumption of the heuristic algorithm output is no larger than that of the noncooperation case for the following reason. The output of Algorithm 2 is an optimal solution to Problem (OPT-RELAX) under a particular set of control parameters $\tilde{\boldsymbol{N}}^{in}, \tilde{\boldsymbol{N}}^{cpu}, \tilde{\boldsymbol{N}}^{up}$. Under the same set of control parameters, the noncooperation case is also a feasible point of Problem (OPT-RELAX). This implies that the energy consumption of the heuristic algorithm output is no larger than that of the noncooperation case; otherwise, the output cannot be an optimal solution of Problem (OPT-RELAX) under the set of control parameters.

Second, when there is no delay constraint, the heuristic algorithm output is an optimal solution of the original problem (OPT) for the following reason. When there is no delay constraint, Algorithm 2 terminates in the first iteration, so that the output of Algorithm 2 is the optimal solution of Problem (OPT-RELAX) under all the control parameters are equal to ones. We refer to such version of Problem (OPT-RELAX) as the basic version of Problem (OPT-RELAX). On the other hand, when there is no delay constraint, comparing with Problem (OPT), the only modified part of the basic version of Problem (OPT-RELAX) is that it relaxes integer variables $\boldsymbol{x}, \boldsymbol{z} \in \{0, 1\}$ to be continuous ones $\boldsymbol{x}, \boldsymbol{z} \in [0, 1]$. As we proved in Lemma 4.3, if solving Problem (OPT-RELAX) using Simplex method, the optimal solution is an integer solution, i.e., $\boldsymbol{x}, \boldsymbol{z} \in \{0, 1\}$. This means that Problem (OPT) and the basic version of Problem (OPT-RELAX) are equivalent. As a result, when there is no delay constraint, the heuristic algorithm output is an optimal solution of the basic version of of Problem (OPT-RELAX), which is equivalent to the original Problem (OPT).

### 4.8.3 Proof for Proposition 4.3

We show that Algorithm 2 will always terminate within $S \times (N-1)$ iterations, where $S$ is the task number and $N$ is the device number.

We can consider the possible allocation of tasks to devices as a bipartite graph: a set of tasks, a set of devices, and a set of links from the tasks to the devices (a task is connected to a device if the task can be potentially allocated to the device). At the beginning of Algorithm 2, the bipartite graph is complete (i.e., every task is connected to every device), since no allocation is prevented. In each iteration, some links is removed from the bipartite graph by adjusting the control parameters. The maximum iteration happens when (i) each iteration removes one link, (ii) all the links are removed excluding the links from tasks to their task owners, where the maximum iteration time is $S \times (N - 1)$.

### 4.8.4 Proof for Equation (4.29)

The expected energy of a task whose owner has a degree $m$, i.e., $\hat{W}_m(\alpha, Np)$, can be simplified as follows:

$$\begin{aligned}
&\hat{W}_m(\alpha, Np)\\
=&\sum_{\hat{N}=0}^{m} C_m^{\hat{N}} \alpha^{\hat{N}} (1-\alpha)^{m-\hat{N}} \int_{\underline{Q}}^{\overline{Q}} \frac{1}{x} d(F(x))^{\hat{N}+1}\\
&\quad \textit{(Integration by Parts)}\\
=&\sum_{\hat{N}=0}^{m} C_m^{\hat{N}} \alpha^{\hat{N}} (1-\alpha)^{m-\hat{N}} \left( \frac{1}{\overline{Q}} + \int_{\underline{Q}}^{\overline{Q}} \frac{(F(x))^{\hat{N}+1}}{x^2} dx \right)\\
&\quad \textit{(Distributive Property; Binomial Theorem)}\\
=&\frac{1}{\overline{Q}} + \int_{\underline{Q}}^{\overline{Q}} \frac{F(x)}{x^2} (1-\alpha+\alpha F(x))^{\hat{N}} dx.
\end{aligned} \tag{4.38}$$

Taking the expectation of $\hat{W}_m(\alpha, Np)$ over all possible degrees $m = \{0, ..., \infty\}$, the expected energy of each task is given as follows:

$$\begin{aligned}
&W(\alpha, Np)\\
=&\sum_{m=0}^{\infty} P(degree = m) \hat{W}_m(\alpha, Np)\\
=&\sum_{m=0}^{\infty} \frac{(Np)^m e^{-Np} \left( \frac{1}{\overline{Q}} + \int_{\underline{Q}}^{\overline{Q}} \frac{F(x)}{x^2} (1-\alpha+\alpha F(x))^m dx \right)}{m!}\\
&\quad \textit{(Distributive Property; Taylor Series, } \sum_{m=0}^{\infty} \frac{x^m}{m!} = e^x \textit{)}\\
=&\frac{1}{\overline{Q}} + \int_{\underline{Q}}^{\overline{Q}} e^{Np(F(x)-1)\alpha} F(x) x^{-2} dx.
\end{aligned} \tag{4.39}$$

This completes the proof of Equation (4.29).

### 4.8.5 Proof for Theorem 4.1

We first present two lemmas indicating the optimal $\alpha^{1C}$ and $p$ that maximizes $\Delta W(r, \alpha^{1C}, p)$ for any $r \geq 1$, then show the maximum energy reduction, i.e., $\max_{\alpha^{1C}, Np} \Delta W(r, \alpha^{1C}, Np)$.

**Lemma 4.4** (Optimal $\alpha^{1C}$ under a Particular $Np$)**.** *For any ratio $r \geq 1$, connection probability $p$, and distribution $f(x)$, there exists an $\alpha^{1C} = \alpha^*_{Np}$ that maximizes the energy reduction $W(r, \alpha^{1C}, p)$, i.e.,*

$$\alpha^*_{Np} = \begin{cases} 1/r, & p \leq \tilde{p} \\ \tilde{\alpha}_{Np}, & p > \tilde{p} \end{cases}, \tag{4.40}$$

*where $\tilde{\alpha}_{Np}$ satisfies*

$$[W(\tilde{\alpha}_{Np}, Np) - W(r\tilde{\alpha}_{Np}, Np)]_\alpha = 0. \tag{4.41}$$

*and connection probability threshold $\tilde{p}$ satisfies*

$$\int_{\underline{Q}}^{\overline{Q}} (F(x) - 1)F(x) \left( e^{\frac{N\tilde{p}(F(x)-1)}{r}} - re^{N\tilde{p}(F(x)-1)} \right) dx = 0. \tag{4.42}$$

*Proof.* The proof path is as follows. We first prove two claims. *Claim (1):* for any $Np$, there exists a unique $\tilde{\alpha}_{Np}$ that maximizes $W(\alpha, Np) - W(r\alpha, Np)$, and it satisfies

$$[W(\alpha, Np) - W(r\alpha, Np)]_\alpha \begin{cases} > 0, & \alpha < \tilde{\alpha}_{Np} \\ = 0, & \alpha = \tilde{\alpha}_{Np} \\ < 0, & \alpha > \tilde{\alpha}_{Np} \end{cases}. \tag{4.43}$$

*Claim (2):* there exists a unique $\tilde{p}$ that satisfies

$$[W(1/r, N\tilde{p}) - W(1, N\tilde{p})]_\alpha \begin{cases} > 0, & p < \tilde{p} \\ = 0, & p = \tilde{p} \\ < 0, & p > \tilde{p} \end{cases}. \tag{4.44}$$

Then, using these two claims, we then prove Lemma 4.4.

*Claim (1):* We first prove that for any $Np$ there exists an $\tilde{\alpha}_{Np}$ that is a local maximizer of $W(\alpha, Np) - W(r\alpha, Np)$. Then, we show that such $\tilde{\alpha}_{Np}$ is the global maximizer and satisfies (4.43).

First, we prove that there exists an $\tilde{\alpha}_{Np}$ that is a local maximizer of $W(\alpha, Np) - W(r\alpha, Np)$. Taking the first-order derivative of $W(\tilde{\alpha}, Np) - W(r\tilde{\alpha}, Np)$ with respect to $\alpha$,

$$\lim_{\alpha \to 0} [W(\alpha, Np) - W(r\alpha, Np)]_{\alpha} > 0, \tag{4.45}$$

$$\lim_{\alpha \to \infty} [W(\alpha) - W(r\alpha)]_{\alpha} \to 0^{-}. \tag{4.46}$$

According to Intermediate Value Theorem, there exists at least an $\tilde{\alpha}_{Np}$ such that

$$\begin{aligned}[] [W(\tilde{\alpha}_{Np}, Np) - W(r\tilde{\alpha}_{Np}, Np)]_{\alpha} \\ = \int_{\underline{Q}}^{\overline{Q}} \left(e^{Np(F(x)-1)\tilde{\alpha}_{Np}} - re^{Np(F(x)-1)r\tilde{\alpha}_{Np}}\right) \\ \times Np(F(x)-1)F(x)x^{-2}dx = 0, \end{aligned} \tag{4.47}$$

which is the $\tilde{\alpha}_{Np}$ that satisfies (4.41). In addition, there exist $\Delta_1$ and $\Delta_2$ such that

$$[W(\tilde{\alpha}_{Np}, Np) - W(r\tilde{\alpha}_{Np}, Np)]_{\alpha} > 0, \alpha \in (\tilde{\alpha}_{Np} - \Delta_1, \tilde{\alpha}_{Np}), \tag{4.48}$$

$$[W(\tilde{\alpha}_{Np}, Np) - W(r\tilde{\alpha}_{Np}, Np)]_{\alpha} < 0, \alpha \in (\tilde{\alpha}_{Np}, \tilde{\alpha}_{Np} + \Delta_2). \tag{4.49}$$

which implies that the $\tilde{\alpha}_{Np}$ is a local maximizer of $W(\alpha, Np) - W(r\alpha, Np)$.

Then, we prove that the local maximizer $\tilde{\alpha}_{Np}$ is unique, so that it is a global maximizer, and it satisfies (4.43). The second-order derivative of $W(\alpha, Np) -$

$W(r\alpha, Np)$ with respect to $\alpha$ is given by

$$[W(\alpha, Np) - W(r\alpha, Np)]_{\alpha\alpha} = \int_{\underline{Q}}^{\overline{Q}} \left(e^{Np(F(x)-1)\alpha} - r^2 e^{Np(F(x)-1)r\alpha}\right) \times \left(Np(F(x)-1)\right)^2 F(x)x^{-2}dx. \quad (4.50)$$

According to First Mean Value Theorem for Definite Integrals, there exists a $\xi \in [\overline{Q}, \underline{Q}]$ such that

$$[W(\alpha, Np) - W(r\alpha, Np)]_{\alpha\alpha} = Np(F(\xi)-1) \int_{\underline{Q}}^{\overline{Q}} Np(F(x)-1)F(x)x^{-2} \times \left(e^{Np(F(x)-1)\alpha} - r^2 e^{Np(F(x)-1)r\alpha}\right) dx. \quad (4.51)$$

Recall that, at $\tilde{\alpha}_{Np}$, the first-order derivative $[W(\tilde{\alpha}_{Np}, Np) - W(r\tilde{\alpha}_{Np}, Np)]_{\alpha} = 0$. By substituting it, the second order derivative at $\tilde{\alpha}_{Np}$ is given by

$$[W(\tilde{\alpha}_{Np}, Np) - W(r\tilde{\alpha}_{Np}, Np)]_{\alpha\alpha} = Np(F(\xi)-1) \int_{\underline{Q}}^{\overline{Q}} Np(F(x)-1)F(x)x^{-2} \times e^{Np(F(x)-1)r\tilde{\alpha}_{Np}} r(1-r)dx \leq 0. \quad (4.52)$$

As a result of $[W(\tilde{\alpha}, Np) - W(r\tilde{\alpha}, Np)]_{\alpha\alpha} \leq 0$, the $\tilde{\alpha}_{Np}$ has to be unique. If it is not unique, there must exist at least one other $\hat{\alpha} \neq \tilde{\alpha}_{Np}$ satisfying $[W(\hat{\alpha}, Np) - W(r\hat{\alpha}, Np)]_{\alpha} = 0$ such that $[W(\hat{\alpha}, Np) - W(r\hat{\alpha}, Np)]_{\alpha\alpha} > 0$, which contradicts $[W(\hat{\alpha}, Np) - W(r\hat{\alpha}, Np)]_{\alpha\alpha} \leq 0$. Hence, the $\tilde{\alpha}_{Np}$ is unique, and it is the global maximizer of $W(\alpha, Np) - W(r\alpha, Np)$. This always implies that $[W(\alpha, Np) - W(r\alpha, Np)]_{\alpha} > 0$ if $\alpha < \tilde{\alpha}_{Np}$; and $[W(\alpha, Np) - W(r\alpha, Np)]_{\alpha} < 0$ if $\alpha > \tilde{\alpha}_{Np}$.

*Claim (2):* We first prove that there exists a $\tilde{p}$ such that $[W(1/r, N\tilde{p}) - W(1, N\tilde{p})]_{\alpha} = 0$. We then show that the $\tilde{p}$ is unique, and satisfies (4.44).

First, we prove that there exists a $\tilde{p}$ such that $[W(1/r, N\tilde{p}) - W(1, N\tilde{p})]_\alpha = 0$. The proof idea is similar as above. Checking the limitation of $[W(1/r, Np) - W(1, Np)]_\alpha$, we have

$$\lim_{Np \to 0} [W(1/r, Np) - W(1, Np)]_\alpha > 0, \tag{4.53}$$

$$\lim_{Np \to \infty} [W(1/r, Np) - W(1, Np)]_\alpha \to 0^-. \tag{4.54}$$

According to Intermediate Value Theorem, there exists at least a $\tilde{p}$ such that

$$\begin{aligned}[W(1/r, N\tilde{p}) - W(1, N\tilde{p})]_\alpha = \int_{\underline{Q}}^{\overline{Q}} & N\tilde{p}(F(x) - 1)F(x)x^{-2} \\ & \times \left(e^{N\tilde{p}(F(x)-1)/r} - re^{N\tilde{p}(F(x)-1)}\right) dx = 0 \end{aligned} \tag{4.55}$$

According to First Mean Value Theorem for Definite Integrals, this equality can be represented as

$$\int_{\underline{Q}}^{\overline{Q}} (F(x) - 1)F(x) \left(e^{\frac{N\tilde{p}(F(x)-1)}{r}} - re^{N\tilde{p}(F(x)-1)}\right) dx = 0. \tag{4.56}$$

Then, we prove that the $\tilde{p}$ is unique, and satisfies (4.44). Taking the first order derivative of $[W(1/r, Np) - W(1, Np)]_\alpha$ with respect to $Np$, we have

$$\begin{aligned} & [[W(1/r, Np) - W(1, Np)]_\alpha]_{Np} \\ & \quad = \int_{\underline{Q}}^{\overline{Q}} \left(e^{Np(F(x)-1)\frac{1}{r}} \frac{1}{r} - e^{Np(F(x)-1)} r\right) \\ & \qquad \times Np(F(x) - 1)^2 F(x) x^{-2} dx. \end{aligned} \tag{4.57}$$

According to First Mean Value Theorem for Definite Integrals, there exists a $\xi \in [\overline{Q}, \underline{Q}]$ such that

$$\begin{aligned} & [[W(1/r, Np) - W(1, Np)]_\alpha]_{Np} \\ & \quad = Np(F(\xi) - 1) \int_{\underline{Q}}^{\overline{Q}} (F(x) - 1)F(x)x^{-2} \\ & \qquad \times \left(e^{Np(F(x)-1)\frac{1}{r}} \frac{1}{r} - e^{Np(F(x)-1)} r\right) dx. \end{aligned} \tag{4.58}$$

Substituting $W(1/r, N\tilde{p}) - W(1, N\tilde{p})]_\alpha = 0$, we have

$$\begin{aligned}&[[W(1/r, N\tilde{p}) - W(1, N\tilde{p})]_\alpha]_{Np} \\ &\qquad = N\tilde{p}(F(\xi) - 1)\int_{\underline{Q}}^{\overline{Q}} (F(x) - 1)F(x)x^{-2} \\ &\qquad\qquad \times e^{N\tilde{p}(F(x)-1)\frac{1}{r}}(\frac{1}{r} - 1)dx \leq 0. \quad (4.59)\end{aligned}$$

Hence, similarly, as a result of $[[W(1/r, N\tilde{p})-W(1, N\tilde{p})]_\alpha]_{Np} \leq 0$, the $\tilde{p}$ has to be unique. If it is not unique, there must exist at least one other $\hat{p} \neq \tilde{p}$ satisfying $[W(1/r, N\hat{p})-W(1, N\hat{p})]_\alpha$ such that $[[W(1/r, N\hat{p})-W(1, N\hat{p})]_\alpha]_{Np} > 0$, which contradicts $[[W(1/r, N\hat{p})-W(1, N\hat{p})]_\alpha]_{Np} \leq 0$. This shows the uniqueness of $\tilde{p}$, and shows that $\tilde{p}$ satisfies (4.44).

Based on *Claim (1)* and *Claim (2)*, we now prove Lemma 4.4. Considering $\Delta W(r, \alpha^{1C}, p) \triangleq W(\alpha^{1C}, Np) - W(\alpha^{3C}, Np)$ with $\alpha^{3C} = \min\{r\alpha, 1\}$ under the following three cases:

- $p < \tilde{p}$: $[W(1/r, Np) - W(1, Np)]_\alpha > 0$. According to *Claim (1)*, this means that $\Delta W(r, \alpha^{1C}, p)$ is increasing in $\alpha^{1C}$ when $\alpha^{1C} < 1/r$. Note that when $\alpha^{1C} \geq 1/r$, $\Delta W(r, \alpha^{1C}, p)$ is decreasing in $\alpha^{1C}$, as $W(\alpha^{1C}, Np)$ is decreasing in $\alpha^{1C}$, and $W(\alpha^{3C}, Np)$ is fixed. Hence, the optimal $\alpha^{1C}$ achieves at $\alpha^{1C} = \alpha^*_{Np} = 1/r$.
- $p = \tilde{p}$: $[W(1/r, Np) - W(1, Np)]_\alpha = 0$, so $\alpha^*_{Np} = 1/r$.
- $p > \tilde{p}$: $[W(1/r, Np) - W(1, Np)]_\alpha < 0$. According to *Claim (1)*, $\alpha^*_{Np} = \tilde{\alpha}_{Np} \leq 1/r$.

□

**Lemma 4.5** (Optimal $\alpha^{1C}$ and $Np$)**.** *For any ratio $r \geq 1$ and distribution $f(x)$, the optimal $\alpha^{1C}$ and $Np$ that maximizes $\Delta W(r, \alpha^{1C}, p)$ is as follows:*

$$\alpha^{1C} = \alpha^*_{N\tilde{p}} = 1/r, \tag{4.60}$$

*and $\tilde{p}$ satisfies*

$$\int_{\underline{Q}}^{\overline{Q}} (F(x)-1)F(x)\left(e^{\frac{N\tilde{p}(F(x)-1)}{r}} - re^{N\tilde{p}(F(x)-1)}\right)dx = 0. \tag{4.61}$$

*Proof.* We will discuss the two cases $p \le \tilde{p}$ and $p \ge \tilde{p}$ one by one, and show that under each of these two cases, $\Delta W(r, \alpha^*_{Np}, p)$ is maximized at $p = \tilde{p}$.

Under $p \le \tilde{p}$, according to Lemma 4.4, $\alpha^*_{Np} = 1/r$. Then, $\Delta W(r, \alpha^*_{Np}, Np)$ is given by

$$\Delta W(r, \alpha^*_{Np}, Np) = W(1/r, Np) - W(1, Np). \tag{4.62}$$

Taking the first-order derivative of $W(1/r, Np) - W(1, Np)$ with respect to $Np$, we have

$$[W(1/r, Np) - W(1, Np)]_{Np} = \int_{\underline{Q}}^{\overline{Q}} (F(x)-1)F(x)x^{-2} \times \left(e^{Np(F(x)-1)\frac{1}{r}}\frac{1}{r} - e^{Np(F(x)-1)}\right)dx \ge 0. \tag{4.63}$$

This shows that $\Delta W(r, \alpha^*_{Np}, Np)$ is maximized at $p = \tilde{p}$.

Under $p \le \tilde{p}$, $\alpha^*_{Np} = \tilde{\alpha}_{Np} \le 1/r$, which is $\Delta W(r, \alpha^*_{Np}, Np) = W(\tilde{\alpha}_{Np}, Np) - W(r\tilde{\alpha}_{Np}, Np)$. Taking derivative of $\Delta W(r, \alpha^*_{Np}, Np)$ with respect to $Np$, we have

$$[\Delta W(r, \alpha^*_{Np}, Np)]_{Np} = \tilde{\alpha}_{Np} \int_{\underline{Q}}^{\overline{Q}} (F(x)-1)F(x)x^{-2} \times \left(e^{Np(F(x)-1)\tilde{\alpha}_{Np}} - re^{Np(F(x)-1)\tilde{\alpha}_{Np}r}\right)dx. \tag{4.64}$$

On the other hand, since that

$$[\Delta W(r, \alpha^*_{Np}, Np)]_{\alpha} = Np \int_{\underline{Q}}^{\overline{Q}} (F(x)-1)F(x)x^{-2} \times \left(e^{Np(F(x)-1)\tilde{\alpha}_{Np}} - re^{Np(F(x)-1)\tilde{\alpha}_{Np}r}\right)dx = 0. \tag{4.65}$$

Hence, $[\Delta W(r, \alpha^*_{Np}, Np)]_{Np} = 0$, so that $\Delta W(r, \alpha^*_{Np}, Np)$ is fixed as $Np$ changes. We can say that $p = \tilde{p}$ maximizes $\Delta W(r, \alpha^*_{Np}, Np)$.

To sum up, the $\alpha^{1C} = \alpha^*_{N\tilde{p}} = 1/r$ and the $\tilde{p}$ maximizes $\Delta W(r, \alpha^{1C}, p)$. □

Based on Lemma 4.5, the $\alpha^{1C} = 1/r$ and $p = \tilde{p}$ satisfying $\int_{\underline{Q}}^{\overline{Q}}(F(x) - 1)F(x)\left(e^{N\tilde{p}(F(x)-1)/r} - re^{N\tilde{p}(F(x)-1)}\right)dx = 0$ maximizes $\Delta W(r, \alpha^{1C}, p)$. Through plugging in these $\alpha^{1C}$ and the $\tilde{p}$, we obtain the maximum energy reduction as in Theorem 4.1.

### 4.8.6 Proof for Equation (4.32)

The analysis for caching will be similar as it for communication/computation. We will first formulate the expected energy consumption, then introduce the analysis idea. In the random group $G(N, p)$, suppose each device joins the cooperative system with a probability $\alpha \in [0, 1]$. Under these case, retrieving each content will have an expected energy of $Z(\alpha, Np)$.[17] Under the 1C model, let us denoted the probability that each device joins the caching sharing model as $\alpha^{1C} \in [0, 1]$; under the 3C framework, the corresponding probability is $\alpha^{3C} = \min\{r\alpha^{1C}, 1\}$, where $r \geq 1$ is a coefficient reflecting the ratio of the increased cooperation opportunities. We will compute the energy reduction $\Delta Z(r, \alpha^{1C}, Np) \triangleq Z(\alpha^{1C}, Np) - Z(\alpha^{3C}, Np)$ for $r \geq 1$.

In the random graph, for a device with degree $m$, the probability that $\hat{N}$ of his neighbors implement the content sharing is $P(\hat{N}|m) = C_m^{\hat{N}}\alpha^{\hat{N}}(1-\alpha)^{m-\hat{N}}$, and the expected probability that the content has not been cached by these $\hat{N}$ devices and itself is $\left(1 - \frac{M^{ca}}{K}\right)^{\hat{N}+1}$. Taking the expectation over $\hat{N} = \{0, 1, ..., m\}$, the expected probability that the content has to be downloaded is given by

$$\hat{Z}_m(\alpha, Np) = \sum_{\hat{N}=0}^{m} P(\hat{N}|m)\left(1 - \frac{M^{ca}}{K}\right)^{\hat{N}+1}. \tag{4.66}$$

[17] To clarify, if the requested content cannot be found in the cache of the devices in the cooperative system, the content will be downloaded by some devices. So the expected energy here corresponds to the downloading energy. In addition, under the homogeneous distribution settings in Section 4.5.1, retrieving any content will have the same expected energy consumption, so we only need to study the expected energy consumption of a content.

Taking the expectation of $\hat{Z}(\alpha, m)$ over all degrees $m = \{0, 1, ..., \infty\}$, the expected probability that the content has to be downloaded is

$$Z(\alpha, Np) = (1 - \frac{M^{ca}}{K})e^{-\alpha Np \frac{M^{ca}}{K}}, \tag{4.67}$$

which is the Equation (4.32). To clarify, we do not consider the sharing of downloading resources here, so the expected energy consumption of a content is the expected probability that the content has to be downloaded multiplied by an expected downloading energy of a device. Normalizing the expected downloading energy to be one, (4.67) is the expected energy consumption of a content.

### 4.8.7 Proof for Theorem 4.2

We first present two lemmas indicating the optimal $\alpha^{1C}$ and $p$ that maximizes $\Delta Z(r, \alpha^{1C}, p) \triangleq Z(\alpha^{1C}, Np) - Z(\alpha^{3C}, Np)$ for any $r \geq 1$, then show the maximum energy reduction, i.e., $\max_{\alpha^{1C}, Np} \Delta Z(r, \alpha^{1C}, Np)$.

**Lemma 4.6** (Optimal $\alpha^{1C}$ under a Particular $Np$)**.** *For any ratio $r$, connection probability $p$, and distribution $f(x)$, there exists a $\alpha^{1C} = \alpha^*_{Np}$ that maximizes the reduction $\Delta Z(r, \alpha^{1C}, Np)$, i.e.,*

$$\alpha^*_{Np} = \begin{cases} \frac{1}{r}, & Np \leq \frac{r \ln r}{(r-1)\frac{M^{ca}}{K}} \\ \frac{\ln r}{(r-1)Np\frac{M^{ca}}{K}}, & Np > \frac{r \ln r}{(r-1)\frac{M^{ca}}{K}} \end{cases} \tag{4.68}$$

The proof idea is similar as it for Lemma 4.4, and we omit the details. Specifically, we first prove two claims. *Claim (1):* for any $Np$, there exists a unique $\tilde{\alpha}_{Np}$ that maximizes $Z(\alpha, Np) - Z(r\alpha, Np)$, and it satisfies

$$[Z(\alpha, Np) - Z(r\alpha, Np)]_\alpha \begin{cases} > 0, & \alpha < \tilde{\alpha}_{Np} \\ = 0, & \alpha = \tilde{\alpha}_{Np} \\ < 0, & \alpha > \tilde{\alpha}_{Np} \end{cases}, \tag{4.69}$$

where

$$\tilde{\alpha}_{Np} = \frac{\ln r}{(r-1)Np\frac{M^{ca}}{K}}. \tag{4.70}$$

*Claim (2):* there exists a unique $\tilde{p}$ that satisfies

$$[Z(1/r, N\tilde{p}) - Z(1, N\tilde{p})]_{\alpha} \begin{cases} > 0, & p < \tilde{p} \\ = 0, & p = \tilde{p} \\ < 0, & p > \tilde{p} \end{cases}, \tag{4.71}$$

where

$$\tilde{p} = \frac{r \ln r}{(r-1)\frac{M^{ca}}{K}}. \tag{4.72}$$

Then, using these two claims, we can prove Lemma 4.6.

**Lemma 4.7** (Optimal $\alpha^{1C}$ and $\tilde{p}$)**.** *For any ratio $r \geq 1$ and distribution $f(x)$, the optimal $\alpha^{1C}$ and $\tilde{p}$ that maximizes $\Delta Z(r, \alpha^{1C}, p)$ is as follows:*

$$\alpha^{1C} = \alpha^{*}_{N\tilde{p}} = 1/r, \;\; \tilde{p} = \frac{r \ln r}{(r-1)\frac{M^{ca}}{K}}. \tag{4.73}$$

The proof idea is similar as it for Lemma 4.5, and we omit the details. Specifically, we can check each of the case $p \leq \tilde{p}$ and $p \geq \tilde{p}$, and show that $p\tilde{p}$ maximizes $\Delta Z(r, \alpha^{*}_{Np}, p)$.

Based on Lemma 4.7, $\alpha^{1C} = 1/r$ and $p = r \ln r/((r-1)M^{ca}/K)$ maximizes $\Delta Z(r, \alpha^{1C}, p)$. By plugging in the $\alpha^{1C}$ and the $p$, we obtain Theorem 4.2.

**Algorithm 2** Heuristic Algorithm

```
**Initialization:** $\tilde{\boldsymbol{N}}^{in}, \tilde{\boldsymbol{N}}^{cpu}, \tilde{\boldsymbol{N}}^{up} \leftarrow 1$
$\boldsymbol{x}, \boldsymbol{z} \leftarrow$Solve Problem (OPT-RELAX) (e.g., using Simplex method [35])
Calculate $T^{down}_s, T^{cpu}_s, T^{up}_s$ using (4.13), (4.14), (4.15)
**while** delay constraints (4.10) are not fully satisfied **do**
    **for** each task $s \in \S$ **do**
        **if** $T^{down}_s > \bar{T}^{down}_s$ **then**
            $\hat{n} \leftarrow \max_n\{\tau^{down}_n y^{down}_{s\to n}\}$    ▷ select a device
            $\boldsymbol{I} \leftarrow \{s| \sum_{\hat{s}\in\boldsymbol{s}_{\hat{n}}} \sum_{k} x^{in}_{\hat{s},k\to\hat{n}} x^{in}_{s,k\to\hat{n}} \geq 1\}$ ▷ find device $\hat{n}$'s *non-preventable* task set
            $\bar{s} \leftarrow \arg\min_s\{\bar{T}^{down}_s | y^{down}_{s\to\hat{n}} \neq 0, s \notin \boldsymbol{I}\}$  ▷ select a task $\bar{s}$ that is preventable
            $\bar{\mathcal{K}} \leftarrow \{k|x^{in}_{\bar{s},k\to\hat{n}}(1-\boldsymbol{Q}^{ca}_{\hat{n}k}) = 1\}$    ▷ select contents
            $\tilde{N}^{in}_{s,k\to\hat{n}} \leftarrow 0, \forall\{s,k|x^{in}_{s,k\to\hat{n}} = 1, k \in \bar{\mathcal{K}}\}$    ▷ prevent the allocations
        **end if**
        **if** $T^{cpu}_s > \bar{T}^{cpu}_s$ **then**
            $\hat{n} \leftarrow \max_n\{\tau^{cpu}_n y^{cpu}_{s\to n}\}$    ▷ select a device
            $\bar{s} \leftarrow \arg\min_s\{\bar{T}^{cpu}_s | y^{cpu}_{s\to\hat{n}} \neq 0, s \notin \boldsymbol{s}_{\hat{n}}\}$    ▷ select a task
            $\tilde{N}^{cpu}_{\bar{s}\to\hat{n}} \leftarrow 0$    ▷ prevent the allocation
        **end if**
        **if** $T^{up}_s > \bar{T}^{up}_s$ **then**
            $\hat{n} \leftarrow \max_n\{\tau^{up}_n y^{up}_{s\to n}\}$    ▷ select a device
            $\bar{s} \leftarrow \arg\min_s\{\bar{T}^{up}_s | y^{up}_{s\to\hat{n}} \neq 0, s \notin \boldsymbol{s}_{\hat{n}}\}$    ▷ select a task
            $\tilde{N}^{up}_{\bar{s},k\to\hat{n}} \leftarrow 0, \forall k$    ▷ prevent the allocations
        **end if**
    **end for**
    $\boldsymbol{x}, \boldsymbol{z} \leftarrow$Solve Problem (OPT-RELAX) (e.g., using Simplex method [35])
    Calculate $T^{down}_s, T^{cpu}_s, T^{up}_s$ using (4.13), (4.14), (4.15)
**end while**
**return** $\boldsymbol{x}, \boldsymbol{z}$
```

# Part II

# Multimedia Platform Operation

# Chapter 5

# Live Streaming Platform

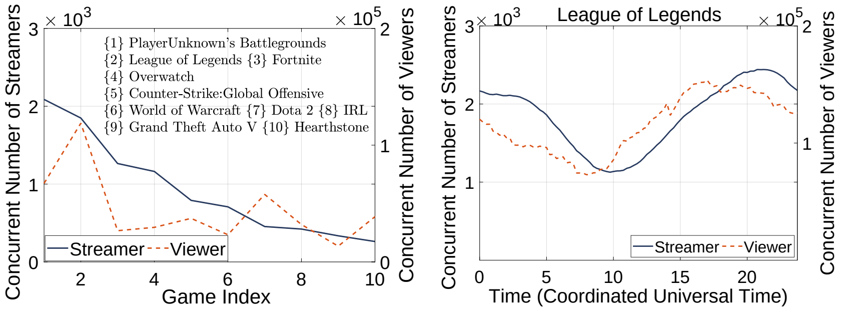
## 5.1 Introduction

### 5.1.1 Background and Motivation

Live streaming platform is an emerging type of multimedia platforms, where individual streamers broadcast live streams for viewers. Twitch[1] is one of the most important commercial examples, which has more than 15 million unique daily visitors and 2 million unique monthly streamers [12].

One important feature of the live streaming platform is that the platform operates based on donation-based markets. Specifically, the streamers provide streaming services without mandatory charges, and the viewers enjoy the services and voluntarily donate to the streamers. The viewers donate mainly due to their desires of being acknowledged on the platforms (to gain community presents) and supporting the streamers for future service provisions [77]. The donations are split between the streamers and the platform with a fixed *donation-split-fraction* (DSF), which corresponds to the fraction of donations kept by the streamers. The total donation volume on similar types of platforms is huge. In 2017, a total of $101 million dollars of donations were

[1] Twitch: `https://www.twitch.tv/`



(a) (b)

Figure 5.1: Mismatch of the concurrent numbers of streamers and viewers in Twitch: (a) game attribute; (b) time attribute.

received by top live streaming platforms including Twitch, YouTube Live, Mixer, Facebook Live, and Periscope [7].

Such donation-based feature bring two unique questions as follows:

First, from the streamers' point of view, *how should they decide their service attributes (e.g., what game to broadcast at what time) given a fixed DSF?* The streamers and viewers may have different preferences over the service attributes, and streamers' choices (which can be different from their own preferences) will affect the competition levels among streamers and the satisfactions of the viewers.

As an example, Figure 5.1 illustrates the mismatch of the numbers of streamers and viewers in Twitch.[2] The subfigures (a) and (b) correspond to game and time attributes, respectively. Specifically, Figure 5.1(a) shows the average concurrent numbers of streamers and viewers (over a two-week data collection period) of different games. Some games with a small number of viewers have a large number of streamers (e.g., {3} Fortnite), which implies that these streamers may improve their payoffs by switching to stream other games with less competitors. Figure 5.1(b) shows the corresponding

[2] Figure 5.1 is based on the stream data that we collected from Twitch. The data is collected every 15 minutes from Nov. 05 to Nov. 20, 2017.

average numbers (over the same two-week period) of the game *League of Legends* at different times. Similarly, some streamers may increase their payoffs by changing their streaming time (e.g., from 3am to 11am in Coordinated Universal Time)

Second, from the platform's point of view, *how should it set the DSF to maximize its payoff?* A higher DSF leads to a smaller per-donation revenue to the platform. On the other hand, it can increase the incentive for the streamers to participate in the platform and better match the viewers' preferences to induce more donations.

Despite the fact that the live streaming platform has been embraced by top companies (e.g., YouTube and Facebook) and attracts millions of streamers and viewers, there does not exist a good understanding regarding the answers of the above two key questions. This motivates the research in this chapter.

### 5.1.2 Solution Approach and Contribution

Although this chapter is motivated by the live streaming platform, the modeling approach and analysis techniques are applicable to donation-based markets in other platforms as well.[3] Hence, for presentation generality, we will use the word "firm" to refer to the streamer, and use the word "customer" to refer to the viewer in the rest of this chapter. The platform first announces the DSF, then each firm decides whether to participate and what service attribute to choose (for example, at what time to stream). We use a two-stage model to capture such a sequential decision process. Such a two-stage game is challenging to solve due to several reasons.

First, consider the Stage II problem where firms make their participation and service attribute selection decisions. This is an extended version of the

[3] Other donation-based market examples are blogging platforms (e.g., WeChat Subscription) and online music platforms (e.g., Songtradr).

Hotelling model [36] with many firms, which is still an open problem [40]. To resolve this issue, we consider a large population approximation where each firm is non-atomic, i.e., a single firm's strategy choice does not affect the entire market. This approximation is reasonable given the large number of firms (and customers) on these platforms in practice. The remaining difficulty is to compute the asymmetric equilibrium, where firms of the same preference may choose different strategies at the equilibrium. This is significant more difficult than focusing on the symmetric equilibrium only as in many previous work [61]. Despite these difficulties, we are able to prove that the Stage II game is a potential game [76], based on which we derive the game equilibria and corresponding equilibrium features.

Next, we consider the Stage II problem where the platform optimizes the value of DSF. The problem is non-convex and hence is challenging to solve. By exploiting the structure of the problem, we derive lower-bound of the optimal solution. To gain further insights, we consider a special case with two attribute values (e.g., streamers decide to broadcast at day or at night), where we show more explicitly how the optimal DSF changes with the system parameters.

Our key contributions are listed as follows:

- *Donation-Based Market Formulation:* To the best of our knowledge, this is the first work that presents a two-stage model of a donation-based market. We characterize how the platform optimizes its DSF, and how the firms decide whether to participate and choose their service attributes.

- *Stage II Equilibrium of Firm Behavior:* For the Stage II problem, we consider a large population approximation with non-atomic firms. We prove that the Stage II problem is a potential game and derive the asymmetric equilibria. We show that a larger DSF leads to more firm participations and a better match to the customers' preferences at the equilibrium.

- *Stage I Equilibrium of Platform Behavior:* For the Stage I non-convex optimization problem, we derive the lower-bound of the optimal solution, which reflects how the optimal DSF changes with system parameters. We further analyze a special case with two attribute values. We show that as the firms' costs increase, the platform may not always choose to share less donations with the firms (hence provide less incentives to firms) to maximize the platform's revenue.
- *A Case Study based on Empirical Twitch Data:* We collect two weeks' data about streamers and viewer behaviors from the Twitch platform. Based on the data, we demonstrate how to compute the platform's optimal DSF without knowing the preferences of the firms and customers. The study suggests that under our data and model settings, Twitch should set the DSF to be 0.38, rather than the 0.71 in reality, to maximize its revenue.

The rest of this chapter is organized as follows. We review the existing works in Section 5.2. We propose the system model in Section 5.3. In Sections 5.4 and 5.5, we analyze the equilibria of Stages II and I, respectively. We perform the case study with Twitch data in Section 5.6, and conclude in Section 5.7.

## 5.2 Literature Review

### 5.2.1 Donation-Based Market

There are only few paper studying the donation-based markets. Most of these prior works studied the customers' donation behaviors in these markets. Hu *et al.* [50] conducted an online survey to study why customers visit live streaming platforms, where some of the reasons (such as cognitive communion and resonant contagion) also explain their donation behaviors. Scheibe *et al.* [77] conducted surveys to study why customers donate, and the major reasons include the customers' desires of being acknowledged on the platforms and supporting the firms for future service provisions. Zhu *et al.* [101] analyzed the data from Douyu (a live streaming platform in China) to investigate the customers' donation behaviors. Tang *et al.* [87] used an all-paid auction framework to understand the customers' donation behaviors.

Through data analysis, some papers identified the importance of motivating firm service selection behaviors. For example, Jia *et al.* [56] discussed the firms' and customers' different preferences on live streams, and mentioned that the platform has to motivate firms to participate and match the customers' preferences to increase the platform's revenue.

However, as far as we know, there is no paper analytically characterizing how the platform should motivate the firms' service selections. As a first step, this chapter studies the platform's optimal DSF decision and analyzes the firms' service selections in these donation-based markets.

### 5.2.2 Hotelling Model

The Stage II of the two-stage game can be regarded as an extended version of the Hotelling model [36, 49]. In the classical Hotelling model setting, customers are distributed along an interval (representing their preferences over an service attribute), and two firms decide their locations over the interval to maximize their own payoffs, respectively. The survey [40] presented various recent extensions of the Hotelling model, such as different service attribute model (e.g., over a line or a circle) and different firm decision process (e.g., simultaneous or sequential).

The Stage II model in this chapter is related to a Hotelling model with a

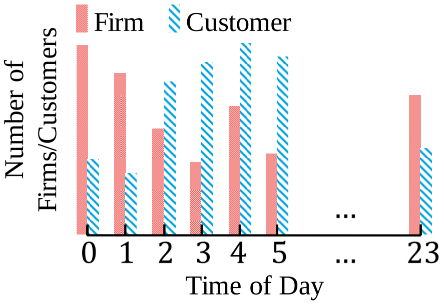


Figure 5.2: System model: an example with time attribute.

larger number of firms. This still remains as an open problem in the literature when the number of firms is arbitrary [40]. Economides [39] studied a multi-firm model on the interval without discussing the firm equilibria. Brenner [24] theoretically studied a three-firm case, and empirically studied four- to nine-firm cases. Behringer *et al.* [22] theoretically analyzed a four-firm case. However, the analysis in [24] and [22] cannot be easily generalized to the case of an arbitrary number of firms.

We circulate the difficulty by approximating the problem with a large number of non-atomic firms, where a single firm's strategy choice does not affect the market. Schmeidler [78] first analyzed a game with non-atomic players, and proved the existence of Nash equilibrium (without deriving the equilibrium). However, we are not aware of papers that explicitly characterizing the equilibrium of a general Hotelling model with non-atomic firms. Based on the reformulated model, we derive the asymmetric equilibria with non-atomic players, which is a challenging problem according to [61].

## 5.3 System Model

In this section, we first introduce the system setting, and then define the two-stage game.

### 5.3.1 System Setting

We first introduce the platform model and the service attribute. Then, we introduce the firm and customer models.

#### Platform

We consider a platform with a large number non-atomic firms and non-atomic customers, where the firm and customer sets are continuum. The large population setting is reasonable in practice, e.g., Twitch often has thousands of streamers and millions of viewers on average (see Figure 5.1).

The firms provide services without mandatory charge, and the customers enjoy the services and voluntarily donate to the firms. The donation will be shared between the firms and the platform with a fixed fraction. As the firms and customers are non-atomic, we will consider the aggregate donations from the customers and the average donations earned by each firm, with details discussed in Section 5.3.1.

#### Service Attribute

For simplicity, we only consider one service attribute in this chapter. For example, on a live game streaming platform, the attribute can be the streaming time or the type of game to be streamed.[4]

Similar as in the classical Hotelling model [36], we represent the service attributes over the unit line segment of $[0, 1]$. We label the possible values of the attributes by the set of $\mathcal{L} = \{0, 1, 2, ..., L\}$, where we call each of these possible values as a "location" for presentation simplicity. A location $s \in \mathcal{L}$ is located at $l_s \in [0, 1]$ along the interval.[5] For the rest of this chapter, we will

[4] Our model can be extended to the case of multi-attribute, under which the potential game in Section 5.4 still applies. For presentation clarify, we focus on the case of one attribute here.

[5] Such a discrete setting is consistent with attributes with discrete values (e.g., the choice of game). For an attribute with continuous values (e.g., streaming time), we can regard the discrete values as a set of

use “location” and “attribute” interchangeably. Figure 5.2 shows an example, where we have location set $\mathcal{L} = \{0, 1, ..., 23\}$, each representing hour of the day. Firms and customers are distributed along the interval based on their preferences, and the firms can decide their stream times over the set $\mathcal{L}$.

### Non-Atomic Firms

A firm has a preferred attribute. Let $N^s$ denote the number of the firms preferring the location $s \in \mathcal{L}$. A firm can choose an attribute that is different from his preference, so as to avoid competitions with other firms or encounter more customers.

### Non-Atomic Customers

A customer has a preferred attribute. Let $M^s$ denote the number of customers preferring the location $s \in \mathcal{L}$. In this chapter, we focus on studying the decision of the firms. For simplicity, we assume that a customer always choose his preferred attribute.[6]

A customer will donate to the firms whose selected attributes are the same as the customer's own preference. Instead of characterizing the donation behavior of each customer, we consider an aggregate donation function depending on the number of firms $N$ (according to the firms' choices) and the number of customers $M$ (according to the customers' preferences) at one particular attribute (location), as captured by a function $D(M, N)$. Such a function can be obtained by data analysis, using similar methods as described in [101]. We assume that $D(M, N)$ satisfies the following assumption.

samples of the continuous values. As the number of the samples increases, this model can approximate an continuous attribute well.

[6] In a more realistic model, the customers may also deviate. However, analyzing the model where both customers and firms deviate is challenging due to the coupling of their choices. We will consider this in future work.

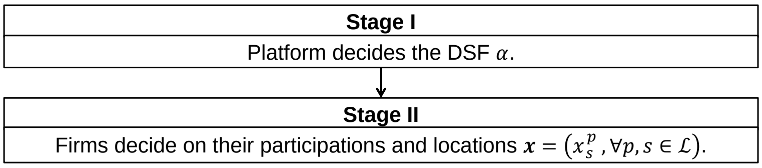

Stage I
Platform decides the DSF $\alpha$.
Stage II
Firms decide on their participations and locations $\boldsymbol{x} = (x_s^p, \forall p, s \in \mathcal{L})$.

Figure 5.3: Two-stage game.

**Assumption 5.1** (Donation Function)**.** *Function $D(M,N)$ (i) is strictly increasing in $M$, (ii) is strictly increasing and concave in $N$, and (iii) has an elasticity that is smaller than one, i.e.,*

$$\eta_M(N) = \frac{[D(M,N)]_N \cdot N}{D(M,N)} \leq 1, \quad \forall M, N \in \mathbb{R}_+, \tag{5.1}$$

*where $[D(M,N)]_N$ denotes the partial derivative of $D(M,N)$ with respect to $N$.*

In Section 5.6.1, we show that the donation function suggested by [101] satisfies Assumption 5.1. Point (i) implies that as the number of customers increases, the total donation strictly increases. Point (ii) implies that as the number of firms increases, the total donation strictly increases but the marginal change decreases. In live streaming platforms, for example, more streamers implies a higher probability that a viewer can find his satisfactory streams so that he will donate more, while the probability of finding a satisfactory stream is concave in the number of streamers.

Point (iii) on elasticity can be written as follows:

$$1 \geq \frac{[D(M,N)]_N \cdot N}{D(M,N)} \approx \frac{\%\Delta D(M,N)}{\%\Delta N}, \quad \forall M, N \in \mathbb{R}_+. \tag{5.2}$$

This implies that a unit percentage increase in the number of firms leads to a percentage donation increase less than one. Because of this, the firms tend to avoid competition (e.g., a streamer would prefer to stream at a time when there are more viewers and less streamers).

### 5.3.2 Two-Stage Game

#### Two-Stage Game

Let us take the live streaming platform as an example. The platform first announces the DSF, then each streamer decides whether to participate and what attribute to choose. We use a two-stage game to capture such a sequential decision process, as shown in Figure 5.3. Next we explain the two stages in more details.

In Stage I, the platform decides the DSF $\alpha \in [0,1]$, i.e., the fraction of donations kept by firms.

In Stage II, firms decide whether to participate and what will be their location choices (if they choose to participate). In this chapter, we use superscripts to denote preferences and subscripts to denote decisions. Let $x_s^p$ denote the number of firms preferring location $p \in \mathcal{L}$ and choosing location $s \in \mathcal{L}$. Note that we allow asymmetric equilibrium, hence $x_s^p$ maybe positive for multiple values of $s$. Due to the assumption of non-atomic firms, the $x_s^p$ can take a non-integer value. The strategies of the firms preferring location $p \in \mathcal{L}$ is characterized by $\boldsymbol{x}^p = (x_s^p, \forall s \in \mathcal{L})$. Let $\boldsymbol{x} = (x_s^p, \forall p, s \in \mathcal{L})$ denote the strategies of all the firms.

Specifically, as we allow the firms to not participate in the system, the sum of the firms choosing all locations could be smaller than the total number of firms, i.e., $\sum_{s\in\mathcal{L}} x_s^p \leq N^p, \forall p \in \mathcal{L}$. Let $\widehat{N}_s(\boldsymbol{x}) \triangleq \sum_{p\in\mathcal{L}} x_s^p$ be the aggregate number of firms choosing location $s$ under strategy $\boldsymbol{x}$.

#### Payoff Functions

Given the platform strategy $\alpha$ and the firm strategy $\boldsymbol{x}$, we define their payoffs as follows:

**Platform's Payoff**: The platform's payoff equals $1-\alpha$ fraction of the total donations from customers at all locations:

$$G(\alpha, \boldsymbol{x}) = (1-\alpha)\sum_{s\in\mathcal{L}} D(M^s, \widehat{N}_s(\boldsymbol{x})). \tag{5.3}$$

**A Firm's Payoff**: If a firm does not participate in the platform, it gains a zero payoff.[7]

If a firm preferring location $p \in \mathcal{L}$ participates and chooses a location $s \in \mathcal{L}$, its payoff equals the difference between the donation gain and its cost, i.e.,

$$F_s^p(\alpha, \boldsymbol{x}) = \alpha \times U(\widehat{N}_s(\boldsymbol{x})) - C_s^p(V, W),\ \forall p, s \in \mathcal{L}. \tag{5.4}$$

Specifically, the donation gain is the DSF $\alpha$ multiplied by the average donation[8] that a firm can gain at the location $s$, where the average donation is defined as

$$U(\widehat{N}_s(\boldsymbol{x})) = \frac{D(M^s, \widehat{N}_s(\boldsymbol{x}))}{\widehat{N}_s(\boldsymbol{x})}. \tag{5.5}$$

The cost[9] contains a fixed opportunity cost $V$ and a distance-associated deviation cost $W \times (l_p - l_s)^2$. Formally,

$$C_s^p(V, W) = V + W \times (l_p - l_s)^2. \tag{5.6}$$

The quadratic form of the deviation cost is used to characterize the firms' increasing marginal costs on the deviation, similar as in the original Hotelling model [36].

Table 5.1 summarizes the key notations of this chapter. We solve the two-stage game using backward induction. Next, we analyze the Stage II equilibrium in Section 5.4 and the Stage I equilibrium in Section 5.5.

[7] If the non-participation induces a positive payoff, we can normalize it to zero by adjusting the value of the opportunity cost $V$ defined in (5.6).

[8] We assume that the firms choosing the same location will equally share the donations from customers at this location. We will study the non-equal sharing case in the future.

[9] Our model can be extended to the case of heterogeneous opportunity and deviation costs, under which the potential game in Section 5.4 still applies. For presentation clarify, we focus on the case of homogeneous costs here.

Table 5.1: Key notations.

| Parameters | |
|---|---|
| $N^s$ | The number of firms preferring location $s \in \mathcal{L}$ |
| $M^s$ | The number of customers preferring location $s \in \mathcal{L}$ |
| $V$ | A firm's opportunity cost for participation |
| $W$ | A firm's deviation cost per unit distance |
| Decisions & Decision-Related Notations | |
| $\alpha$ | Platform's DSF decision |
| $x_s^p$ | The number of firms preferring $p \in \mathcal{L}$ and choosing $s \in \mathcal{L}$ |
| $\boldsymbol{x}^p$ | $\boldsymbol{x}^p = (x_s^p, \forall s \in \mathcal{L})$, the strategies of firms preferring $p \in \mathcal{L}$ |
| $\boldsymbol{x}$ | $\boldsymbol{x} = (x_s^p, \forall p, s \in \mathcal{L})$, the strategies of all firms |
| $\widehat{N}_s(\boldsymbol{x})$ | $\widehat{N}_s(\boldsymbol{x}) = \sum_{p \in \mathcal{L}} x_s^p$, the total number of firms choosing location $s \in \mathcal{L}$ under strategy $\boldsymbol{x}$ |

## 5.4 Stage II: Firm Location Equilibrium

In Stage II, given any DSF $\alpha$, we analyze the firm location game as follows.

**Definition 5.1** (Stage II Firm Location Game)**.**

- *Players: all the firms;*
- *Strategies: each firm preferring a location $p \in \mathcal{L}$ selects a location $s \in \mathcal{L}$, and the aggregate strategy is represented by $\boldsymbol{x} = (x_s^p, \forall p, s \in \mathcal{L})$;*
- *Payoffs: $F_s^p(\alpha, \boldsymbol{x})$ for each firm preferring a location $p \in \mathcal{L}$ and choosing a location $s \in \mathcal{L}$.*

Next, we first define the firm location equilibrium, and then derive the equilibrium and its corresponding features.

### 5.4.1 Equilibrium Definition

We first define the support correspondence. Then, we define a firm's best response and the firm location equilibrium based on such a correspondence.

#### Support Correspondence

We define a correspondence that outputs a vector's positive elements.

**Definition 5.2** (Support Correspondence)**.** *For a vector $\boldsymbol{z} \in \mathbb{R}_+^{1\times L}$, the support correspondence $S(\boldsymbol{z}) = \{s \in \mathcal{L} : z_s > 0\}$ is the set of indexes corresponding to positive elements in $\boldsymbol{z}$.*

For example, if $\boldsymbol{z} = \{3, 1, 0, 0, 2, 0\}$, then $S(\boldsymbol{z}) = \{1, 2, 5\}$. Consequently, for any strategy $\boldsymbol{x}^p$, the correspondence $S(\boldsymbol{x}^p) = \{s \in \mathcal{L} : x_s^p > 0\}$ indicates the set of locations which are chosen by the firms preferring location $p$ under the strategy $\boldsymbol{x}^p$.

#### Best Response

We now define a firm's best response.

For a firm preferring location $p \in \mathcal{L}$, its best response location choice is the set of locations that induce the maximum firm payoff, i.e.,

$$BR^p(\alpha, \boldsymbol{x}) = \arg\max_{s\in\mathcal{L}} F_s^p(\alpha, \boldsymbol{x}). \tag{5.7}$$

Normally, the best response is defined as a correspondence of all other firms' strategies excluding the firm its own's. However, due to the non-atomic firm assumption, the change of one firm's strategy does not affect the aggregate strategies of all the firms [17]. This allows us to directly write the best response as a correspondence of $\boldsymbol{x}$.

Based on a single firm's best response, we define a correspondence representing the aggregate best response of all the firms preferring location $p \in \mathcal{L}$:

$$ABR^p(\alpha, \boldsymbol{x}) = \{\boldsymbol{z} \in \mathbb{R}_+^{1\times L} : S(\boldsymbol{z}) \subset BR^p(\alpha, \boldsymbol{x}), \textstyle\sum_{s\in\mathcal{L}} z_s \le N^p\}. \tag{5.8}$$

Specifically, the aggregate best response for the firms preferring a location $p$ is any vector $\boldsymbol{z}$ where all the elements (locations) with positive firm numbers belong to $BR^p(\alpha, \boldsymbol{x})$. Notice that we allow firms with the same preference to choose different locations in their best responses.

#### Definition of Firm Location Equilibrium

Firm location equilibrium is defined as the fixed point of the best responses.

**Definition 5.3** (Firm Location Equilibrium)**.** *Given any $\alpha$, firm location strategy $\boldsymbol{x}$ is an equilibrium if and only if the aggregate strategy of each firm population preferring the same location $p$ belongs to their aggregate best response under $\boldsymbol{x}$, i,e., $\boldsymbol{x}^p \in ABR^p(\alpha, \boldsymbol{x}), \forall p \in \mathcal{L}$.*

An interpretation of this equilibrium is that an $\boldsymbol{x}$ is an equilibrium if and only if the firms' aggregate best response (i.e., the updated firm strategies under their best responses) can recover this strategy distribution $\boldsymbol{x}$.

### 5.4.2 Deriving the Firm Location Equilibrium

Directly computing the equilibrium based on the best response is challenging, due to the challenge of computing the fixed point of the multi-dimensional best response mapping of an $L \times L$-dimensional vector $\boldsymbol{x} = \{x_s^p, \forall p, s \in \mathcal{L}\}$. On the other hand, using a distributed best response update to find the equilibrium is also infeasible under the non-atomic firm setting, because the equilibrium involves the firms' aggregated behavior distribution instead of their individual behaviors.

Instead, to derive the equilibrium distribution, we first prove that the Stage II game is a potential game. Under this, all the firms' payoffs can be related to the same function, i.e., the potential function, which allows us to characterize the equilibrium by solving an optimization problem. Then, we derive the firm location equilibrium.

The key proof of a potential game is to identify a potential function. However, there does not exist a general methodology for doing this, and we have to identify the potential function by exploiting the specific structure of the problem.

**Lemma 5.1** (Stage II Game as Potential Game)**.** *Given any $\alpha$, the Stage II game is a potential game with non-atomic players*[10]*, which has a potential function*

$$f(\alpha,\boldsymbol{x}) = \alpha \times \sum_{s\in\mathcal{L}} \int_0^{\widehat{N}_s(\boldsymbol{x})} \frac{D(M^s,z)}{z} dz \\ - V \times \sum_{s\in\mathcal{L}}\sum_{p\in\mathcal{L}} x_s^p - W \times \sum_{s\in\mathcal{L}}\sum_{p\in\mathcal{L}} x_s^p (l_p - l_s)^2. \quad (5.9)$$

*Proof.* According to [76], the Stage II game is a potential game if there is a potential function $f(\alpha,\boldsymbol{x})$ such that the following equality always holds:

$$\frac{\partial f(\alpha,\boldsymbol{x})}{\partial x_s^p} = F_s^p(\alpha,\boldsymbol{x}),\ \forall s,p\in\mathcal{L}. \quad (5.10)$$

By checking the first-order partial derivative of (5.9), we can show that the Stage II game is a potential game, and $f(\alpha,\boldsymbol{x})$ is the potential function. □

Showing that the game is a potential game allows us to characterize the Stage II firm location equilibrium by solving an optimization problem, which is easier than finding the fixed point of firms' best responses. Formally,

**Theorem 5.1** (Firm Location Equilibrium)**.** *The set of firm location equilibria of the Stage II game is the set of global optimal solutions to the following optimization problem:*

$$\boldsymbol{x}^*(\alpha) \triangleq \mathop{\textit{arg maximize}}_{\boldsymbol{x}\geq 0} \quad f(\alpha,\boldsymbol{x}) \quad (5.11a)$$

$$\textit{subject to} \quad \textstyle\sum_{s\in\mathcal{L}} x_s^p \leq N^p,\ \forall p\in\mathcal{L}. \quad (5.11b)$$

*(STAGE II-NE)*

[10] A potential game with non-atomic players is different from that with atomic players. For detailed discussions, see [76].

*More specifically, a vector* $\boldsymbol{x}$ *is an equilibrium, i.e.,* $\boldsymbol{x} \in \boldsymbol{x}^*(\alpha)$*, if and only if there exists a pair of* $\boldsymbol{\mu} \in \mathbb{R}^{1\times L}$ *and* $\boldsymbol{\lambda} \in \mathbb{R}^{L\times L}$ *such that the following constraints are satisfied:*

$$F_s^p(\alpha, \boldsymbol{x}) = \mu^p - \lambda_s^p, \qquad \forall p, s, \qquad (5.12a)$$

$$\lambda_s^p x_s^p = 0,\ \lambda_s^p \geq 0,\ x_s^p \geq 0, \qquad \forall p, s, \qquad (5.12b)$$

$$(\textstyle\sum_{s\in\mathcal{L}} x_s^p - N^p)\mu^p = 0,\ \mu^p \geq 0, \qquad \forall p, \qquad (5.12c)$$

$$\textstyle\sum_{s\in\mathcal{L}} x_s^p \leq N^p, \qquad \forall p. \qquad (5.12d)$$

*(STAGE II-NE-CONDITION)*

*Proof.* We have proved in Lemma 5.1 that this game is a potential game, so its equilibria are the solutions to Problem (STAGE II-NE) [76]. On the other hand, the conditions (STAGE II-NE-CONDITION) are essentially the KKT conditions of Problem (STAGE II-NE). To show that (STAGE II-NE-CONDITION) are the conditions for equilibria, we have to show that the KKT conditions of Problem (STAGE II-NE) is necessary and sufficient conditions to its global optimal solutions. This is true because the $f(\alpha, \boldsymbol{x})$ in (5.11a) is concave (by checking its Hessian Matrix), and the constraint (5.11b) fulfills the Slater's condition. □

Theorem 5.1 implies that the firm location equilibrium may not be unique under a given $\alpha$. However, we can show that any of the equilibria leads to the same sets of firms' payoffs and the same platform's payoff,

**Corollary 5.1** (Unique Equilibrium Payoffs). *Under any given* $\alpha$*, any equilibrium of the Stage II game induces the same set of firms' payoffs, i.e.,* $F_s^p(\alpha, \boldsymbol{x}) = F_s^p(\alpha), \forall p, s \in \boldsymbol{\mathcal{L}}, \boldsymbol{x} \in \boldsymbol{x}^*(\alpha)$*, and the same platform's payoff, i.e.,* $G(\alpha, \boldsymbol{x}) = G(\alpha), \forall \boldsymbol{x} \in \boldsymbol{x}^*(\alpha)$*.*

*Proof.* The dual variables of Problem (STAGE II-NE), i.e., $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$, are unique, because the constraints (5.11b) are linearly independent [63]. Hence, according to (5.12a), under a fixed $\alpha$, any of the equilibria induces the same set of firms' payoffs, i.e., $F_s^p(\alpha, \boldsymbol{x}) = F_s^p(\alpha), \forall p, s \in \mathcal{L}, \boldsymbol{x} \in \boldsymbol{x}^*(\alpha)$. Based on this, we can show that under a fixed $\alpha$, any of the equilibria induces the same set of aggregate number of firms at all locations, i.e., $\widehat{N}_s(\boldsymbol{x}) = \widetilde{N}_s(\alpha), \forall s \in \mathcal{L}, \boldsymbol{x} \in \boldsymbol{x}^*(\alpha)$. This is because the mapping from the firms' payoffs (defined in (5.4)) to aggregate number of firms is a one-to-one correspondence, due to the strictly increasing donation function in the number of firms as in Assumption 5.1. Hence, from the platform's point of view, given any $\alpha$, it achieves the same payoff under any of the firm location equilibria in Stage II, as its payoff (defined in (5.3)) only depends on the aggregate number of firms $\widehat{N}_s(\boldsymbol{x})$. □

In the rest of this chapter, let $G(\alpha)$ denote the platform's payoff under the firm location equilibrium given an $\alpha$.

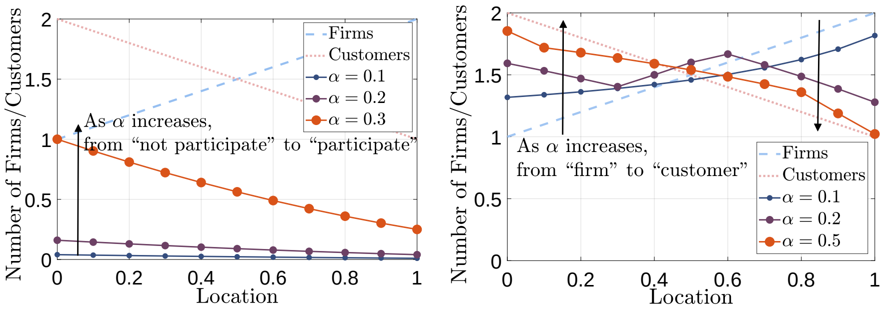
### 5.4.3 Impact of $\alpha$ on Firm Location Equilibrium

Based on the conditions in Theorem 5.1, we show how the firm location equilibrium changes with the DSF $\alpha$. A key insight is that a larger $\alpha$ leads to more firm participations and a better match to the customers' preferences.

Given any opportunity cost $V$, deviation cost $W$, firm preference $N^s, \forall s \in \mathcal{L}$, and customer preference $M^s, \forall s \in \mathcal{L}$, the firm location equilibrium changes with $\alpha$ as follows.

**Proposition 5.1** (Participation and Preference Matching)**.** *The ratios $W/\alpha$ and $V/\alpha$ determine the Stage II equilibria, i.e.,*

- *$V/\alpha \to 0$: full participation, $\sum_{s\in\mathcal{L}} x_s^p = N^p, \forall p \in \mathcal{L}$;*
- *$V/\alpha \to \infty$: no participation, $\sum_{s\in\mathcal{L}} x_s^p = 0, \forall p \in \mathcal{L}$;*



(a) (b)

Figure 5.4: The impact of $\alpha$ on firm equilibrium: (a) $W = 0$, $V = 1$; (b) $W = 1$, $V = 0$.

- $W/\alpha \to 0$: *full preference matching*[11],

$$U(\widehat{N}_s(\boldsymbol{x})) = \frac{D(M^s, \widehat{N}_s(\boldsymbol{x}))}{\widehat{N}_s(\boldsymbol{x})} = \widetilde{U}, \forall s \in \mathcal{L}, \tag{5.13}$$

*where* $\widetilde{U}$ *is a positive value;*

- $W/\alpha \to \infty$: *no active matching,* $\sum_{s\in\mathcal{L}/p} x_s^p = 0, \forall p \in \mathcal{L}$.

Specifically, the potential function (5.9) is a weighted sum of three functions with the corresponding weights as $\alpha$, $V$, and $W$, respectively. As these weights change, the firm location equilibrium (the optimal solution of Problem (STAGE II-NE)) changes accordingly as in Proposition 5.1.

Figure 5.4 shows the impact of $\alpha$ under two choices of $(W, V)$. The donation function is $D(M, N) = M\sqrt{N}$. The x-axis represents the location, and the y-axis corresponds to the number of firms or customers. The "Firm" and "Customer" curves correspond to the firms' and customers' location preferences, respectively. The curves labeled with $\alpha = 0.1$, $0.2$, and $0.3$ are the firms' location equilibrium under the corresponding values of $\alpha$.

Figure 5.4 (a) shows the results with $W = 0$ and $V = 1$, under which firms always fully match the customers' preferences due to the zero deviation

[11] We refer this case as "full preference matching", because the firms gain the same average donations, i.e., $U(\widehat{N}_s(\boldsymbol{x}))$, at all locations (under the deviation), so that further deviation cannot increase their payoffs.

cost $W$. In this case, under any $\alpha$, firms are distributed in a shape that is similar as the customers' preferences do. As $\alpha$ increases, the firm participation increases, i.e., the total number of participating firms increases. Figure 5.4 (b) shows the result with $W = 1$ and $V = 0$, under which firms always fully participate due to the zero opportunity cost $V$. In this case, as $\alpha$ increases, the firm matching increases, i.e., firms' location choices deviate from firms' preferences (i.e., the blue dash line) to match customers' preferences (i.e., the red dot line). To sum up, a larger $\alpha$ leads to more firm participation and a better match to the customers' preferences.

Next, we analyze the platform's optimal DSF in Stage I.

## 5.5 Stage I: Platform DSF Decision

In Stage I, the platform chooses the DSF $\alpha$ to maximize its payoff. We first present the platform's payoff optimization problem. As the problem is non-convex and cannot be solved in closed-form, we derive the lower-bound of the optimal solution. Finally, we study a special case to demonstrate how system parameters affect the optimal DSF.

### 5.5.1 Platform Profit Maximization Problem

In Stage I, the platform selects the optimal fraction $\alpha^*$ that maximizes its payoff. Formally,

$$\alpha^* \triangleq \arg \max_{\alpha \in [0,1]} \quad G(\alpha) \tag{STAGE I-NE}$$

Here $G(\alpha)$ is the platform's payoff under the firm location equilibrium given an $\alpha$, as in Corollary 5.1 in Section 5.4.2.

Problem (STAGE I-NE) is a non-convex optimization problem due to the non-convex objective function $G(\alpha)$. Specifically, the objective function $G(\alpha)$ is a piece-wise function that is not always differentiable. In addition, this piece-wise function may not be a quasi-concave function, so we cannot use an effective bisection algorithm [25] to solve the problem. Hence, it is difficult to derive the closed-form optimal solution to Problem (STAGE I-NE).

### 5.5.2 Optimal Solution Bounds and Approximate Solution

Despite the non-convexity of Problem (STAGE I-NE), we can characterize the lower-bound of the optimal solution. This can reduce the complexity for us to search for the optimal $\alpha^*$.

#### Lower-Bound of $\alpha^*$

The lower-bound of $\alpha^*$ are as follows:

**Proposition 5.2** (Lower-Bound of $\alpha^*$). *The optimal $\alpha^*$ is lower-bounded by $\underline{\alpha}$ as follows:*

$$\underline{\alpha} = \min\left\{\frac{V}{\max\{\frac{D(M^p,N^p)}{N^p}, \forall p \in \mathcal{L}\}}, \min\{\eta_{M^p}(\tilde{x}^p), \forall p \in \mathcal{L}\}\right\}, \tag{5.14}$$

*where $[D(M^p, \tilde{x}^p)]_x = V$ and $\eta_M(N)$ is defined in* (5.1).

The proof is presented in Appendix 5.8.1. Specifically, the first part characterizes how hard it is to motivate participation: if either the opportunity cost $V$ is larger or the $\max\{D(M^p, N^p)/N^p, \forall p \in \mathcal{L}\}$ is smaller,[12] the lower bound is larger, which implies a larger incentive is needed to motivate firm participation. The second part characterizes how much benefit the platform can gain by motivating firms to participate. This is captured by the elasticity: when the elasticity is larger, the customers' donations are more sensitive to the number of firms, so the platform should increase $\alpha^*$ to incentivize firms to participate and satisfy customers' preferences.

[12] Intuitively, a smaller $\max\{D(M^p, N^p)/N^p, \forall p \in \mathcal{L}\}$ implies a smaller average donation after full participation, under which it is harder to motivate the full participation.

**Searching Method**

The last step of computing the optimal DSF is to search in the interval of $[\underline{\alpha}, 1]$. Specifically, we divide the internal into $K$ segments, and the approximate optimal solution is $\alpha_K^* = \arg\max\{G(\alpha)|\alpha \in \{\underline{\alpha} + (1-\underline{\alpha})k/(K-1), k = 0, 1, ..., K-1\}\}$. Let $\alpha^* = \arg\max_{\alpha\in[0,1]} G(\alpha)$ be the actual optimal solution. The gap between $\alpha_K^*$ and $\alpha^*$ is bounded:

**Lemma 5.2** (Optimal Solution Approximation)**.** *Given any $\epsilon$, there always exists a threshold $\underline{K}$ such that $|G(\alpha_K^*) - G(\alpha^*)| \leq \epsilon$ for any $K \geq \underline{K}$.*

The proof of Lemma 5.2 relies on applying the Maximum Theorem [29] to show the continuity of function $G(\alpha)$. The proof is in Appendix 5.8.2.

### 5.5.3 A Special Case: How $\alpha^*$ Changes with System Parameters

To better understand the optimal DSF $\alpha^*$, we consider a special case and study how the optimal $\alpha^*$ changes with system parameters. This special case shows that as the firms' costs increase, the platform may not always choose to share less donations with the firms (hence provide less incentives to firms) in order to maximize the platform's revenue.

We first introduce the setting of the special case. (i) There are only two available locations, i.e., $\mathcal{L} = \{1, 2\}$. For example, streamers can choose to stream at day or night. (ii) The customers are uniformly distributed over the locations based on their preferences, i.e., $M_1 = M_2 = 1$. The firm population, however, can have asymmetric preferences of the two locations. Without the loss of generality, we assume that $N^2 \geq N^1$. (iii) The donation function $D(M, N) = MN^{1/r}$ with $r \geq 2$, which satisfies Assumption 5.1 for any $M$ and $N$. By choosing a proper value of $r$, this donation function (under $M = 1$) is the same as the one that obtained from empirical data in [101].

Under this special case, we study how the optimal DSF $\alpha^*$ changes with

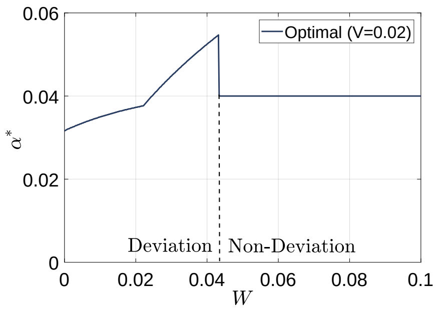


Figure 5.5: Impact of $W$.

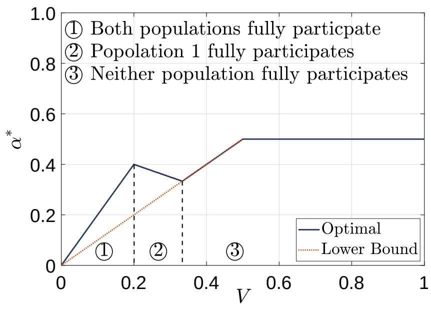


Figure 5.6: Impact of $V$.

the deviation cost $W$ and the opportunity cost $V$. Specifically, we can obtain the explicit formulation of $G(\alpha)$, based on which we can derive the following results. For presentation simplicity, we refer to firms preferring location 1 and 2 as populations 1 and 2, respectively.

#### Impact of the Deviation Cost $W$

As $W$ increases, there always exists a threshold such that the platform changes from motivating deviation (i.e., preference matching) to not, where there will be a drop in $\alpha^*$.

**Proposition 5.3** (Impact of $W$). *There exists a threshold $W^\circ$ such that the following are true:*

- *$W \le W^\circ$: $\alpha^*$ changes with $W$, i.e., $\alpha^* = A^{\mathrm{D}}(W)$, under which firms deviate, i.e., $\sum_{p\in\mathcal{L}}\sum_{s\in\mathcal{L}/p} x_s^p > 0$.*
- *$W > W^\circ$: $\alpha^*$ is a constant, i.e., $\alpha^* = A^{\mathrm{ND}}$, under which firms do not deviate, i.e., $\sum_{p\in\mathcal{L}}\sum_{s\in\mathcal{L}/p} x_s^p = 0$.*

*In addition, $A^{\mathrm{D}}(W^\circ) \ge A^{\mathrm{ND}}$.*

Proposition 5.3 shows that as $W$ increases from 0, the platform first aims to motivate the preference matching. As $W$ becomes larger than the threshold $W^\circ$, the platform changes its strategy and chooses not to motivate the firms to match the customers' preferences. In addition, there is a sudden decrease of $\alpha^*$ at which the platform gives up motivating the preference matching (at $W^\circ$). Figure 5.5 shows an example with $V = 0.02$, with x-axis being the deviation cost $W$ and y-axis being the optimal $\alpha^*$. In this case, $W^o = 0.044$.

### Impact of the Opportunity Cost $V$

As $V$ increases, the platform changes from motivating full participation to not, while the optimal $\alpha^*$ may not be monotonically increasing.

We first show an overview of how the $\alpha^*$ changes with $V$.

**Proposition 5.4** (Impact of $V$)**.** *There exist thresholds $V_1^\circ$ and $V_2^\circ$ such that the following are true:*

- $V \le V_1^\circ$*:* $\alpha^*$ *is a constant, i.e.,* $\alpha^* = A^{\text{FP}}$*, under which firms fully participate, i.e.,* $\sum_{s\in\mathcal{L}} x_s^p = N^p, \forall p \in \mathcal{L}$*.*
- $V_1^\circ \le V \le V_2^\circ$*:* $\alpha^*$ *changes with* $V$*, i.e.,* $\alpha^* = A^{\text{PP}}(V)$*, which is bounded by*

$$\max\left\{A^{\text{FP}}, \min\left\{V \times (N^1)^{\frac{r-1}{r}}, \frac{1}{r}\right\}\right\} \le A^{\text{PP}}(V) \le 1/r. \tag{5.15}$$

- $V \ge V_2^\circ$*:* $\alpha^*$ *is a constant, i.e.,* $\alpha^* = A^{\text{NP}} = 1/r$*, under which firms never fully participate, i.e.,* $\sum_{s\in\mathcal{L}} x_s^p < N^p, \forall p \in \mathcal{L}$*.*

*In addition,* $A^{\text{FP}} \le A^{\text{PP}}(V) \le A^{\text{NP}}$ *for all* $V \in [V_1^\circ, V_2^\circ]$*.*

Specifically, as $V$ increases, the platform first motivates fully participation, then gradually reduces its effort on providing incentives. After $V \ge V_2^\circ$, the platform no longer motivates fully participation, because the cost for providing the incentive is too high due to the large opportunity cost.

According to Proposition 5.4, an intuitive guess is that as $V$ increases, the platform keeps increasing the optimal $\alpha^*$ during $V_1^\circ \le V \le V_2^\circ$ until $\alpha^* = A^{\text{NP}}$.

However, this is not always true. There are two scenarios that may induce the decrease of $\alpha^*$ in $V$: (i) when $\alpha^*$ leads to an firm location equilibrium that population 1 fully participates[13] while population 2 does not; (ii) when the platform changes from motivating preference matching to not. Due to the space limit, we only illustrate scenario (i) with the following example.

**Example 5.1** (Decreasing $\alpha^*$ with $V$)**.** *Let $N^1 = 1$, $N^2 = 4$, $W = 0.5$, and $r = 2$, under which $A^{\text{FP}} = 0$ and $A^{\text{NP}} = 1/2$. The impact of $V$ on the optimal $\alpha^*$ is shown in Figure 5.6. Within the internal ②, the optimal $\alpha^*$ leads to an equilibrium that population 1 fully participates while population 2 does not, under which the $\alpha^*$ decreases with $V$. Intuitively, as $V$ increases, increasing $\alpha^*$ only increases the participation of population 2 but not that of population 1, and the loss of per-donation revenue for the platform by doing so cannot be recovered by the increasing number of firm participation. As a result, it is optimal for the platform to decrease $\alpha^*$ within the internal ②.*

## 5.6 Case Study with Data Collected from Twitch

We collect real-world data from the Twitch platform and conduct analysis based on our model accordingly. The data is collected from Twitch every 15 minutes from Nov. 05 to Nov. 20, 2017. The information contains user_id, game_id, streamer_type, viewer_count, started_at, and language.

In this case study, we demonstrate how to compute the platform's optimal DSF with only the firms' and customers' actual behaviors data (instead of their preferences, which are usually private information). The study suggests that under the collected data and our model settings, Twitch should significantly reduce the value of DSF (comparing with its current practice) to maximize its payoff.

[13] A population $p = \{1, 2\}$ fully participate if $\sum_{s \in \mathcal{L}} x_s^p = N^p$.

We focus on the game *League of Legends*, and consider the streaming time as the service attribute. To better capture the periodic feature of the time attribute and the streamers' multiple time slots streams, we propose a new attribute model, i.e., a circular model with multiple location coverage. Although this model is different (and more complicated) from the unit length interval model discussed in Section 5.3, our modeling and analysis are still applicable (detailed discussions in 5.6.1).

We first discuss the system setting, then explain how to map from the firms choices of streaming time (observable from the collected data) to their time preferences (not directly observable). Finally, we derive the optimal DSF.

### 5.6.1 System Setting

#### New Attribute Model

The model here is different from the previous model (in Section 5.3) in two aspects. (i) Circular model: the attributes are distributed over a circle (with no extreme values) instead of over a line interval (with two extreme values). (ii) Multiple location coverage: once a firm selects a location, its service can cover several locations (starting from the selected one). Let $\mathcal{L}_s \subset \mathcal{L}$ denote the set of locations that a firm can cover if it selects a location $s \in \mathcal{L}$. Figure 5.7 illustrates such a model, where the circle represents 24 hours in a day. More specifically, the circle contains 96 locations, so the distance between each pair of adjacent locations corresponds to 15 minutes (which corresponds to the time interval between two consecutive data collections in our dataset). We assume that each streamer broadcasts for an consecutive period of 2 hours, which corresponds to the average broadcasting time of streamers in *League of Legends*. For example, when a streamer selects 1am (location 4), he will continue to serve until 3am (location 12), represented by the shaded area in Figure 5.7, i.e., $\mathcal{L}_4 = \{4, 5, ..., 12\}$.

Under this new attribute model, if a firm preferring location $p \in \mathcal{L}$ participates and chooses a location $s \in \mathcal{L}$, its payoff is the difference between the donation gain over all the locations in set $\mathcal{L}_s$ and its cost in the circular model, i.e.,

$$\widetilde{F}_s^p(\alpha, \boldsymbol{x}) = \alpha \times \textstyle\sum_{l \in \mathcal{L}_s} U(\widehat{N}_l(\boldsymbol{x})) - \widetilde{C}_s^p(V, W), \tag{5.16}$$

where the cost in the circular model $\widetilde{C}_s^p(V, W)$ is the sum of the opportunity cost and the deviation cost that is associated the shortest path along the circle, i.e.,

$$\widetilde{C}_s^p(V, W) = V + W \times \left(\min\{|l_p - l_s|, 1 - |l_p - l_s|\}\right)^2. \tag{5.17}$$

The firms' non-participation payoffs and the platform's payoff are the same as those of the line model in Section 5.3.

Under this new model, we can still prove that the Stage II game is a potential game, just with a different and more complicated potential function $\widetilde{f}(\alpha, \boldsymbol{x})$.

**Lemma 5.3** (Firm Location Equilibrium Under the New Attribute Model)**.** *Given any $\alpha$, the Stage II game is a potential game with non-atomic players, with a potential function*

$$\widetilde{f}(\alpha, \boldsymbol{x}) = \alpha \times \left( \sum_{s \in \mathcal{L}} \sum_{l \in \mathcal{L}_s} \int_0^{\sum_{h \in L(l)} \sum_{p \in \mathcal{L}} x_h^p} \frac{D(M^l, z)}{z \times |L(l)|} dz \right) - \sum_{s \in \mathcal{L}} \sum_{p \in \mathcal{L}} x_s^p \widetilde{C}_s^p(V, W), \tag{5.18}$$

*where $L(l) \triangleq \{h | l \in \mathcal{L}_h\}$ is the location strategy set that can cover location $l$, and $|L(l)|$ is the size of the set $L(l)$.*

*The set of firm location equilibria of the Stage II game is the set of global optimal solutions of the following problem:*

$$\boldsymbol{x}_{cir}^*(\alpha) \triangleq \mathop{arg\ maximize}_{\boldsymbol{x} \geq 0} \quad \widetilde{f}(\alpha, \boldsymbol{x}), \tag{5.19a}$$

$$\textit{subject to} \quad \textstyle\sum_{s \in \mathcal{L}} x_s^p \leq N^p, \ \forall p \in \mathcal{L}. \tag{5.19b}$$

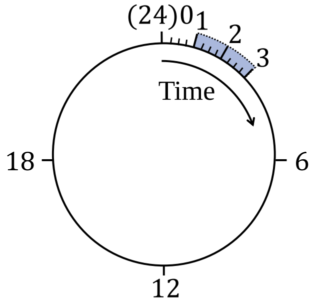


Figure 5.7: Firm model.

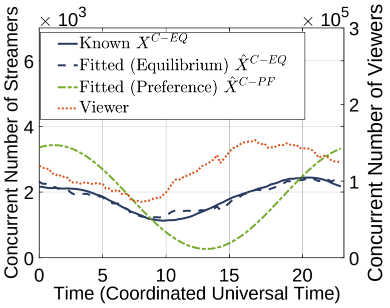


Figure 5.8: A fitting example.

Accordingly, we can obtain a similar result as it in Corollary 5.1. That is, under given any fixed $\alpha$, all Stage II equilibria lead to the same sets of firms' payoffs and the same platform's payoff. Hence, in Stage I, we can compute the optimal DSF by searching over an interval of $\alpha$ as in Section 5.5.2.

**Donation Function**

As Twitch does not provide donation information through API, we choose the donation function according to paper [101], which analyzes the donation from Douyu[14]. In [101], the donation to a live stream increases with the number of viewers in the following manner:

$$[\text{received donation per firm}] = e^{b_0}([\text{viewers per firm}])^{b_1}, \tag{5.20}$$

where $b_0 = -1.17$ and $b_1 = 0.6$ based on empirical data. Hence, we use the following donation function:

$$D(M, N) = e^{b_0}\left(M/N\right)^{b_1} N, \tag{5.21}$$

which is the per-firm donation $e^{b_0}(M/N)^{b_1}$ multiplied by the number of firms. This $D(M, N)$ satisfies Assumption 5.1.

[14] Douyu is one of the most popular live streaming platforms in China, and it has a similar business model as Twitch.

**Current Donation-Split-Fraction**

On Twitch, viewers purchase 100 bits (i.e., a virtual currency on Twitch) with \$1.4, while streamers can exchange 100 bits with \$1. Hence, $\alpha = 1/1.4 \approx 0.71$.

### 5.6.2 Mapping from Firm Distribution to Firm Preference

Before deriving the firms' and platform's equilibrium strategies, we need to first estimate the firms' preference locations based on the firms' and customers' actual locations.

Specially, based on the known cumulative firm distribution $\boldsymbol{X}^{\text{C-EQ}}$ (i.e., how many firms serving customers at each location in the dataset, assuming that these firms behave according to the equilibrium in Lemma 5.3), we aim to estimate the unknown actual firm preference distribution $\boldsymbol{X}^{\text{PF}}$ (i.e., how many firms prefer to start at each location). The consideration of "cumulative" is due to fact that a streamer will cover multiple locations. The estimated firm preference is denoted by $\hat{\boldsymbol{X}}^{\text{PF}}$, based on which we can obtain the cumulative firm (preference) distribution $\hat{\boldsymbol{X}}^{\text{C-PF}}$ (i.e., how many firms serving customers at each location if all the firms start at their preferring locations), and the cumulative firm (equilibrium) distribution $\hat{\boldsymbol{X}}^{\text{C-EQ}}$ (which will be different from $\boldsymbol{X}^{\text{C-EQ}}$ due to the errors introduced in the estimation process). Each of the vectors defined above has 96 elements, where the $s^{th}$ element corresponds to the number of firms at location $s$.

We assume that the firm actual preference $\boldsymbol{X}^{\text{PF}}$ follows a sine function, i.e., the number of firms at a location $l \in \mathcal{L}$ is $S(l) = c_1 \times sin(2\pi l + c_2) + c_3$. This is because the lag plot of $\boldsymbol{X}^{\text{C-EQ}}$ follows a circular shape, which suggests that the $\boldsymbol{X}^{\text{C-EQ}}$ is in a sine function [5]. Moreover, we can verify through simulation that a sine function preference $\boldsymbol{X}^{\text{PF}}$ is likely to output a sine function equilibrium $\boldsymbol{X}^{\text{C-EQ}}$.

To estimate $\boldsymbol{X}^{\text{PF}}$, the key is to estimate the set of parameters $(c_1, c_2, c_3)$.

Table 5.2: Optimal DSF.

| $V\backslash W$ | 2 | 10 | 20 | 100 | 200 | 1000 |
|---|---|---|---|---|---|---|
| 0.2 | 0.08 | 0.09 | 0.08 | 0.07 | 0.07 | 0.07 |
| 1 | 0.32 | 0.31 | 0.30 | 0.28 | 0.28 | 0.28 |
| 2 | 0.39 | **0.38** | 0.38 | 0.40 | 0.40 | 0.40 |
| 4 | 0.40 | 0.40 | 0.40 | 0.40 | 0.40 | 0.40 |

Table 5.3: Platform's payoff increase under the optimal DSF in Table 5.2, comparing with that under 0.71 set by Twitch.

| $V\backslash W$ | 2 | 10 | 20 | 100 | 200 | 1000 |
|---|---|---|---|---|---|---|
| 0.2 | 215% | 205% | 206% | 217% | 217% | 218% |
| 1 | 132% | 120% | 120% | 132% | 132% | 132% |
| 2 | 55% | **54%** | 53% | 43% | 43% | 43% |
| 4 | 41% | 41% | 41% | 41% | 41% | 41% |

We choose the parameters that minimize the root-mean-square-error (RMSE) between the known cumulative distribution $\boldsymbol{X}^{\text{C-EQ}}$ and fitted cumulative distribution $\hat{\boldsymbol{X}}^{\text{C-EQ}}$:

$$\text{RMSE} = \sqrt{\sum_{l \in \mathcal{L}} \left( X_l^{\text{C-EQ}} - \hat{X}_l^{\text{C-EQ}} \right)^2 / L}. \tag{5.22}$$

Figure 5.8 shows the fitting result (under the parameters $(c_1, c_2, c_3)$ that lead to the minimum RMSE) under $b_1 = 0.6$.[15] The "Fitted (Equilibrium) $\hat{X}^{\text{C-EQ}}$" is the fitted cumulative firm distribution in equilibrium, and the "Fitted (Preference) $\hat{X}^{\text{C-PF}}$" is the fitted cumulative firm distribution when all the firms choose their preferring attributes. We can see that the streamers deviate from "Fitted (Preference) $\hat{\boldsymbol{X}}^{\text{C-PF}}$" to "Fitted (Equilibrium) $\hat{X}^{\text{C-EQ}}$" in order to better match the viewers' preferences distribution.

[15] The probability density function of the fitting residuals roughly follows a normal distribution with zero mean. This implies that the residuals are random, which suggests that our fitting model works well [5].

### 5.6.3 Deriving Optimal Donation-Split-Fraction

Based on the estimated $\hat{\boldsymbol{X}}^{\mathrm{PF}}$, we derive the platform's optimal DSF based on Lemma 5.3 and the numerical search mentioned in Section 5.5.2. Table 5.2 shows the optimal DSF values under different possible values of $V$ and $W$ (as we do not know the actual values). Table 5.3 shows how much the platform's payoff increases under the optimal DSF given in Table 5.2, comparing with that under 0.71 currently implemented by Twitch.

In both tables, we use bold fonts to represent the values corresponding to $V = 2$ and $W = 10$, because this combination of parameters leads to the minimum RMSE (defined in (5.22)) over all the possible values (hence is most likely to be the one in reality).

**Remark 5.1** (Optimal $\alpha^*$)**.** *In Table 5.2, the bold text suggests that the optimal DSF should be* $\alpha^* = 0.38$*. Furthermore, all the optimal DSF values under various* $V$ *and* $W$ *values are significantly smaller than* 0.71 *chosen by Twitch.*

**Remark 5.2** (Platform's Payoff Increase)**.** *In Table 5.3, the bold text suggests that the platform's payoff can increase by* 54% *under the optimal DSF choice of* $\alpha^* = 0.38$*, comparing with that under* 0.71 *chosen by Twitch. Furthermore, the platform's payoff increase could be up to* 218% *based on our recommendation, if* $V = 0.2$ *and* $W = 1000$ *in reality.*

## 5.7 Chapter Summary

This chapter studies the donation-based market in a live streaming platform, understanding the platform's optimal donation-split-fraction (DSF) choice and the firms' equilibrium service attribute selections. Our analysis shows that, regarding the firm service attribute selection, a larger DSF leads to more firm participations and a better match to the customers' preferences.

Regarding the platform's optimal DSF, we derive the lower-bound of the optimal DSF, and show that as the firms' costs increase, the platform may not always choose to share less donations with the firms (hence provide less incentives to firms) to maximize the platform's revenue. In addition, we perform a case study based on the dataset from Twitch. Our analysis and simulation results suggest that there exists a significant potential for Twitch to improve its revenue, by setting the DSF to 0.38, instead of 0.71 as in Twitch's current practice.

## 5.8 Appendix

### 5.8.1 Proof for Proposition 5.2

We prove this lower bound by showing that for any $\alpha \leq \underline{\alpha}$, the platform's payoff increases with $\alpha$, so the optimal DSF $\alpha^*$ cannot be smaller than $\underline{\alpha}$.

Before proving the lower bound, we first prove several lemmas that reveal the features of the firm location equilibrium.

**Lemma 5.4.** *In an equilibrium, firms deviate to a location $s \in \mathcal{L}$ only if the firms preferring location $s$ fully participate, i.e., if $\sum_{l\in\mathcal{L}} x_l^s < N^s$, there cannot be any $p \neq s$ and $p \in \mathcal{L}$ such that $x_s^p > 0$.*

*Proof.* Suppose $\sum_{l\in\mathcal{L}} x_l^s < N^s$, according to (STAGE II-NE-CONDITION), $F_l^s(\alpha, \boldsymbol{x}) = \mu^s = 0, \forall l \in \mathcal{L}$. As a result,

$$F_s^s(\alpha, \boldsymbol{x}) = \frac{D(M^s, \widehat{N}_s(\boldsymbol{x}))}{\widehat{N}_s(\boldsymbol{x})} - V \leq \mu^s = 0. \tag{5.23}$$

Hence, there cannot be any firm deviate to location $s$, because for all $p \neq s$,

$$F_s^p(\alpha, \boldsymbol{x}) = \alpha \times \frac{D(M^s, \widehat{N}_s(\boldsymbol{x}))}{\widehat{N}_s(\boldsymbol{x})} - V - W \times (l_p - l_s)^2 < 0, \tag{5.24}$$

which implies $\lambda_s^p > 0$ (according to (5.12a)), so that $x_s^p = 0$. □

**Lemma 5.5.** *In an equilibrium, if there exists a set of locations $\widehat{\mathcal{L}} \subset \mathcal{L}$ such that all the firms preferring any location $p \in \widehat{\mathcal{L}}$ participate, i.e., $\sum_{s\in\mathcal{L}} x_s^p = N^p$, there exists a $q \in \widehat{\mathcal{L}}$ such that that all the firms preferring location $q$ participate and choose location $q$, i.e., $\sum_{s\in\mathcal{L}} x_s^q = N^q$ and $x_s^q = 0, \forall s \neq q$.*

*Proof.* If the set $\widehat{\mathcal{L}}$ contains one element, this only element $q \in \widehat{\mathcal{L}}$ satisfies $\sum_{s\in\mathcal{L}} x_s^q = N^q$ and $x_s^q = 0, \forall s \neq q$. This is obtained according to Lemma 5.4.

If the set $\widehat{\mathcal{L}}$ contains more than one elements, the deviation destination can only be among the set $\widehat{\mathcal{L}}$, according to Lemma 5.4. Hence, there exists a location $q \in \widehat{\mathcal{L}}$ such that it has the maximum average donations among the set $\widehat{\mathcal{L}}$, i.e.,

$$q = \arg\max_{p\in\widehat{\mathcal{L}}} \frac{D(M^p, \sum_{i\in\mathcal{L}} x_p^i)}{\sum_{i\in\mathcal{L}} x_p^i}, \tag{5.25}$$

so that the firms preferring location $p$ cannot gain more payoff through deviation. As a result, $\sum_{s\in\mathcal{L}} x_s^q = N^q$ and $x_s^q = 0, \forall s \neq q$. □

**Lemma 5.6.** *If the following holds*

$$\alpha < \min\{\frac{V}{D(M^p, N^p)/N^p}, \forall p \in \mathcal{L}\} = \frac{V}{\max\{D(M^p, N^p)/N^p, \forall p \in \mathcal{L}\}}, \tag{5.26}$$

*all the firm populations do not fully participate, i.e., $\sum_{s\in\mathcal{L}} x_s^p < N^p, \forall p \in \mathcal{L}$.*

*Proof.* We prove this lemma by contradiction. According to Lemma 5.5, if there exist a $p \in \mathcal{L}$ such that $\sum_{s\in\mathcal{L}} x_s^p = N^p$, there exists a $q \in \mathcal{L}$ such that that all the firms preferring location $q$ participate and choose location $q$, i.e., $\sum_{s\in\mathcal{L}} x_s^q = N^q$ and $x_s^q = 0, \forall s \neq q$. This implies that the following inequality should hold:

$$F_q^q(\alpha, \boldsymbol{x}) = \alpha \times \frac{D(M^q, N^q)}{N^q} - V \geq 0, \tag{5.27}$$

and this is contradict to (5.26). □

We now prove Proposition 5.2. According to Lemma 5.6, within $\alpha \in \Big[0, V/\max\{\frac{D(M^p,N^p)}{N^p}, \forall p \in \mathcal{L}\}\Big)$, all the firm populations do not fully participate in the firm location equilibrium. Based on such a type of firm location equilibrium, we can write the specific formulation of the platform's payoff function (within this range of $\alpha$) as follows:

$$G(\alpha) = (1-\alpha)\sum_{s\in\mathcal{P}} D(M^s, x^s(\alpha)),\ \alpha \in \left[0, \frac{V}{\max\{\frac{D(M^p,N^p)}{N^p}, \forall p \in \mathcal{L}\}}\right), \quad (5.28)$$

where $x^s(\alpha) = \{x|D(M^s,x)/x = V/\alpha\}$. This function (5.28) is concave, because it is a summation of concave functions. According to first-order condition, function (5.28) is maximized at

$$\underline{\alpha} = \min\left\{\frac{V}{\max\{\frac{D(M^p,N^p)}{N^p}, \forall p \in \mathcal{L}\}}, \min\{\eta_{M^p}(\tilde{x}^p), \forall p \in \mathcal{L}\}\right\}, \quad (5.29)$$

where $[D(M^p,\tilde{x}^p)]_x = V$ and $\eta_M(N)$ is defined in equation (5.1). Hence, for any for any $\alpha \leq \underline{\alpha}$, the platform's payoff increases with $\alpha$.

### 5.8.2 Proof for Lemma 5.2

The platform's payoff $G(\alpha)$ can be written as follows:

$$G(\alpha) = \underset{\alpha\in[0,1]}{\text{maximize}} \quad (1-\alpha)\sum_{s\in\mathcal{L}} D(M^s, \widehat{N}_s(\boldsymbol{x}^*)) \quad (5.30a)$$

$$\text{subject to} \quad \boldsymbol{x}^* \in \arg\max_{\boldsymbol{x}\geq 0} f(\alpha, \boldsymbol{x}). \quad (5.30b)$$

According to Maximum Theory, $G(\alpha)$ is continuous in $\alpha$. Hence, due to the continuity, for any $\epsilon > 0$, there always exists a $\delta_\epsilon > 0$ such that $|G(\alpha) - G(\alpha^*)| \leq \epsilon$ for any $|\alpha - \alpha^*| \leq \delta_\epsilon$. We define the $\underline{K} = \lceil 1/\delta_\epsilon \rceil + 1$. For any $K \geq \underline{K}$, there always exists a sampled $\hat{\alpha}$ (which may be $\hat{\alpha} \neq \alpha_K$) such that $|\hat{\alpha} - \alpha^*| \leq \delta_\epsilon$, so that $|G(\hat{\alpha}) - G(\alpha^*)| \leq \epsilon$. According to the definition of $\alpha_K$, $|G(\alpha_K^*) - G(\alpha^*)| \leq |G(\hat{\alpha}) - G(\alpha^*)| \leq \epsilon$.

# Chapter 6

# Conclusion and Future Work

In this thesis, we studied the optimizations and economics of multimedia from two aspects: multimedia service provision and multimedia platform operation. Our study on the service provision significantly enhances the QoE of mobile multimedia services, and our study on the platform operation provides useful insights for the marketing of the live streaming platform.

First, for multimedia service provision, we enhanced the QoE of mobile multimedia services by user cooperation enabling and efficient 3C resource scheduling. We started with communication resource sharing in a video streaming application scenario, and proposed a crowdsourced mobile video streaming (CMS) model that enables mobile users to share their downloading resources through D2D connections for cooperative video streaming. Under CMS model, we designed effectively online scheduling algorithm that approaches to the theoretical performance bound of the CMS model. We also proposed incentive mechanisms for the CMS model to motivate user cooperation, where the mechanisms achieve truthful user information revelation and efficient resource allocation. We then proposed a general 3C resource sharing framework for mobile multimedia services. This framework generalizes many of existing mobile user sharing models, and provides additional network design and optimization flexibilities.

Then, for multimedia platform operation, we focused on live streaming platforms and its donation-based markets, analyzing the platform's optimal donation-split-fraction (DSF) selection and the streamers' equilibrium service attribute selections. Our analysis shows that, regarding the streamers' service attribute selections, a larger DSF leads to more streamer participations and a better match to the viewers preferences. Regarding the platform's optimal DSF, we derive the lower-bound of the optimal DSF, and show that as the streamers' costs increase, the platform may not always choose to share less donations with the streamers (hence provide less incentives to streamers) to maximize the platform's revenue. In addition, we performed a case study based on the dataset collected from Twitch, a top live streaming platform. Our analysis and simulation results suggest that there exists a significant potential for Twitch to improve its revenue by decreasing its DSF.

Next, we discuss several potential future research directions.

## 6.1 Extensions on Communication Sharing

In Chapters 2 and 3, we proposed a CMS model for the communication sharing in video streaming services, and studied the corresponding optimization problem and incentive mechanism design, respectively. There are several interesting future research directions in this area.

**Human mobility pattern:** In reality, instead of being static (as we assumed in the current work), mobile users are always moving from one location to another, which makes user cooperation groups time-varying and makes it difficult to schedule resources effectively. Frequent disconnections of D2D connections among users can increase the signaling overhead of the cooperation and lead to downloading resource waste (e.g., a helper may find a receiver disconnected after downloading the requested segment). Thus, it is important to design effective and robust algorithms by taking the users' random mobility into consideration.

**Security and privacy:** The security and privacy issues are always crucial in wireless networks, especially when mobile users share local network resources and individual information frequently. It is important to design proper authentication and monitoring schemes to support real-time video streaming services together with distributed and massive D2D connections.

**Interventions of content providers and network operators:** The current work mainly focuses on the cooperation among video users, without considering the potential involvement of network operators and video content providers. In practice, network operators may be reluctant to support the network resource sharing scheme among users, and some network operators (e.g., AT&T in the United States) have started to charge additional fees for "tethering" among users. Considering such an intervention, Meng et al. [96] provide an initial study on deriving the optimal data and tethering price for the crowdsourced networking scheme. Moreover, video content providers may not be willing to support user cooperation, for example, when a user with a certain monthly subscription for an unlimited video plan downloads video for another user with a usage-based video subscription plan. Hence, to implement the CMS model in reality, we need to further consider incentives for the content providers and network operators.

## 6.2 Extensions on Communication, Computation, and Caching Sharing

In Chapter 4, we proposed the joint 3C resource sharing model for mobile multimedia services, and studied the corresponding offline resource scheduling problem. There are several interesting directions for future research.

**Online scheduling algorithm:** In the current work, we studied the offline resource allocation problem, which is the first-step for studying the joint 3C framework. In reality, however, when scheduling resources, many information (e.g., future network channel conditions) are unknown to both the users and the system operator, so it is important to design an online resource scheduling algorithm. In the joint 3C framework, nevertheless, designing an online scheduling algorithm is challenging due to the joint resource optimization. Specifically, each service may require a sequence of heterogeneous operations, e.g., downloading, computing, and then uploading, and the allocation of any operation will impact on the optimal allocation of the other operations. Besides the joint resource optimization, such an algorithm should also address limited system and user information and have a low computational complexity to support real-time service provision.

**Incentive mechanism design:** Due to the cost of resource sharing, mobile users may not be willing to share their resources unless there is a proper incentive mechanism. Although we have proposed incentive mechanisms for the communication sharing in Chapter 3, the mechanisms cannot be implemented in the joint 3C framework, because of the sharing of three different resources. Specifically, the incentive mechanism should address the different features among the three different resources. For example, when sharing computation and communication resources, the shared resources are occupied and cannot be utilized by other users; however, when sharing cached content, a copy of the shared content is delivered to the requesting user, while the original content file is still stored and can be shared to other users. Facing such heterogeneous features, it is challenging to evaluate users' contribution in the joint 3C sharing model and determine the proper incentive levels.

## 6.3 Extensions on Live Streaming Platform

In Chapter 5, we studied the donation-based markets in live streaming platforms, understanding the platform's and the streamers' optimal decisions. There are several interesting directions to extend this work.

**Heterogeneous streamer quality:** In the current model, we assumed that the streamers selecting a same attribute value will equally share the donations from viewers. It is also worthy to study the model that the streamers are heterogeneous in their streaming qualities, so that the donations are not equally shared among them. Such an extended model will help to understand how the streamers should behave differently given their various qualities, and how the platform should behave to encourage the participation of the streamers with particular qualities.

**Strategic viewer service selection:** Not only the streamers but also the viewers will decide on their service attribute values by considering the streamers' and viewers' preferences. For example, for a viewer who likes game *Hearthstone*, he may eventually choose to watch the game *PlayerUnknown's Battlegrounds* due to the limited number of streamers streaming *Hearthstone*. The viewer may also choose to watch the games with a large number of viewers to be better involved in the live streaming community. It is challenging to study such a model with both strategic streamers and viewers. This is because the users in each of the two groups (streamers and viewers) have to consider the strategic decisions and preferences of the users in not only the same group but the other group, while the two groups of users have quite different settings and objectives.

# Bibliography

[1] BesTV. http://www.bestv.com.cn/.

[2] Http dynamic streaming. http://www.adobe.com/products/hds-dynamic-streaming.html.

[3] The ilesansfil/wifidog dataset. http://crawdad.cs.dartmouth.edu/ilesansfil/wifidog/.

[4] Nextwifi. http://www.nextwifi.cn/wifind/.

[5] Nist/sematech e-handbook of statistical methods. http://www.itl.nist.gov/div898/handbook/.

[6] Smooth streaming. http://www.iis.net/downloads/microsoft/smooth-streaming.

[7] Streamlabs livestreaming q4 report: Tipping reaches $100m for the year; youtube dominates in streamer growth, increasing by 343% as twitch rises 197% in 2017. https://blog.streamlabs.com/streamlabs-livestreaming-q4-report-tipping-reaches-100m-for-the-year-youtube-dominates-in-4bf450fae536.

[8] Wifi direct data performance tests. https://msdn.microsoft.com/en-us/library/windows/hardware/dn247504(v=vs.85).aspx.

[9] OIPF release 2 specification, volume 2a - http adaptive streaming. Technical report, OIPF, 2014.

[10] Cisco visual networking index: Global mobile data traffic forecast update, 2015-2020. Technical report, Cisco, 2016.

[11] United states speedtest market report. Technical report, Speedtest, 2016.

[12] Twitch year in review. Technical report, Twitch, 2018.

[13] Abdallah, M. , Cavagna, R. , and Laval, D. . Incentive-based on-demand video streaming using a dual spatially-organized peer-to-peer network. In *IEEE Consumer Communications and Networking Conference*, pages 689–695, 2015.

[14] Akhshabi, S. , Begen, A. C. , and Dovrolis, C. . An experimental evaluation of rate-adaptation algorithms in adaptive streaming over http. In *ACM Conference on Multimedia systems*, pages 157–168, 2011.

[15] Arnold, B. C. , Balakrishnan, N. , and Nagaraja, H. N. . *A first course in order statistics*, volume 54. Siam, 1992.

[16] Asker, J. and Cantillon, E. . Properties of scoring auctions. *The RAND Journal of Economics*, 39(1):69–85, 2008.

[17] Aumann, R. J. and Shapley, L. S. . *Values of non-atomic games*. Princeton University Press, 2015.

[18] Balasubramanian, N. , Balasubramanian, A. , and Venkataramani, A. . Energy consumption in mobile phones: a measurement study and implications for network applications. In *ACM SIGCOMM Conference on Internet Measurement*, pages 280–293, 2009.

[19] Balasubramanian, N. , Balasubramanian, A. , and Venkataramani, A. . Energy consumption in mobile phones: a measurement study and implications for network applications. In *ACM SIGCOMM Conference on Internet Measurement*, pages 280–293, 2009.

[20] Barabási, A.-L. . *Network science*. Cambridge university press, 2016.

[21] Bastug, E. , Bennis, M. , and others, . Living on the edge: The role of proactive caching in 5g wireless networks. *IEEE Communications Magazine*, 8(52):82–89, 2014.

[22] Behringer, S. and Filistrucchi, L. . Hotelling competition and political differentiation with more than two newspapers. *Information Economics and Policy*, 30:36–49, 2015.

[23] Bichler, M. and Werthner, H. . A classification framework of multidimensional, multi-unit procurement negotiations. In *IEEE International Workshop on Database and Expert Systems Applications*, pages 1003–1009, 2000.

[24] Brenner, S. . Hotelling games with three, four, and more players. *Journal of Regional Science*, 45(4):851–864, 2005.

[25] Burden, R. L. and Faires, J. D. . Numerical analysis. 2001. *Brooks/Cole, USA*, 2001.

[26] Burkardt, J. . The truncated normal distribution. *Department of Scientific Computing Website, Florida State University*, 2014.

[27] Camp, T. , Boleng, J. , and Davies, V. . A survey of mobility models for ad hoc network research. *Wireless communications and mobile computing*, 2(5):483–502, 2002.

[28] Cao, Y. , Chen, X. , Jiang, T. , and Zhang, J. . Socast: Social ties based cooperative video multicast. In *IEEE International Conference on Computer Communications*, pages 415–423, 2014.

[29] Čech, E. , Frolík, Z. , and Katětov, M. . Topological spaces. 1966.

[30] Chao, H.-P. and Wilson, R. . Multi-dimensional procurement auctions for power reserves: Robust incentive-compatible scoring and settlement rules. *Journal of Regulatory Economics*, 22(2):161–183, 2002.

[31] Che, Y.-K. . Design competition through multidimensional auctions. *The RAND Journal of Economics*, pages 668–680, 1993.

[32] Chen, M. , Hao, Y. , Li, Y. , Lai, C.-F. , and Wu, D. . On the computation offloading at ad hoc cloudlet: architecture and service modes. *IEEE Communications Magazine*, 53(6):18–24, 2015.

[33] Chen, Z. , Liu, Y. , Zhou, B. , and Tao, M. . Caching incentive design in wireless d2d networks: A stackelberg game approach. In *IEEE International Conference on Communications*, pages 1–6, 2016.

[34] Chi, F. , Wang, X. , Cai, W. , and Leung, V. . Ad-hoc cloudlet based cooperative cloud gaming. *IEEE Transactions on Cloud Computing*, 2015.

[35] Dantzig, G. B. . *Origins of the simplex method.* ACM, 1990.

[36] D'Aspremont, C. , Gabszewicz, J. J. , and Thisse, J.-F. . On hotelling's" stability in competition". *Econometrica*, pages 1145–1150, 1979.

[37] David, E. , Azoulay-Schwartz, R. , and Kraus, S. . Bidding in sealed-bid and english multi-attribute auctions. *Decision Support Systems*, 42(2):527–556, 2006.

[38] Destounis, A. , Paschos, G. S. , and Koutsopoulos, I. . Streaming big data meets backpressure in distributed network computation. In *IEEE International Conference on Computer Communications*, pages 1–9, 2016.

[39] Economides, N. . Minimal and maximal product differentiation in hotelling's duopoly. *Economics Letters*, 21(1):67–71, 1986.

[40] Eiselt, H. A. and Marianov, V. . *Foundations of location analysis*, volume 155. Springer Science & Business Media, 2011.

[41] El Essaili, A. , Schroeder, D. , Steinbach, E. , Staehle, D. , and Shehada, M. . Qoe-based traffic and resource management for adaptive http video delivery in lte. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(6):988–1001, 2015.

[42] Gao, L. , Tang, M. , Pang, H. , Huang, J. , and Sun, L. . Performance bound analysis for crowdsourced mobile video streaming. pages 366–371, 2016.

[43] Garcia, M.-N. , De Simone, F. , Tavakoli, S. , Staelens, N. , Egger, S. , Brunnström, K. , and Raake, A. . Quality of experience and http adaptive streaming: A review of subjective studies. In *IEEE International Workshop on Quality of Multimedia Experience*, pages 141–146, 2014.

[44] Georgopoulos, P. , Elkhatib, Y. , Broadbent, M. , Mu, M. , and Race, N. . Towards network-wide qoe fairness using openflow-assisted adaptive video streaming. In *ACM SIGCOMM workshop on Future human-centric multimedia networking*, pages 15–20, 2013.

[45] Gimpel, H. and Mäkiö, J. . Towards multi-attribute double auctions for financial markets. *Electronic Markets*, 16(2):130–139, 2006.

[46] Hao, J. , Zimmermann, R. , and Ma, H. . Gtube: geo-predictive video streaming over http in mobile environments. In *ACM Multimedia Systems Conference*, pages 259–270, 2014.

[47] He, Z. , Cheng, W. , and Chen, X. . Energy minimization of portable video communication devices based on power-rate-distortion optimization. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(5):596–608, 2008.

[48] Heller, I. and Tompkins, C. . An extension of a theorem of dantzig's. *Linear inequalities and related systems*, 38:247–254, 1956.

[49] Hotelling, H. . Stability in competition. In *The Collected Economics Articles of Harold Hotelling*, pages 50–63. Springer, 1990.

[50] Hu, M. , Zhang, M. , and Wang, Y. . Why do audiences choose to keep watching on live video streaming platforms? an explanation of dual identification framework. *Computers in Human Behavior*, 75:594–606, 2017.

[51] Huang, T.-Y. , Johari, R. , McKeown, N. , Trunnell, M. , and Watson, M. . A buffer-based approach to rate adaptation: Evidence from a large video streaming service. *ACM SIGCOMM Computer Communication Review*, 44(4):187–198, 2015.

[52] Iosifidis, G. , Gao, L. , Huang, J. , and Tassiulas, L. . Enabling crowd-sourced mobile internet access. In *IEEE International Conference on Computer Communications*, pages 451–459, 2014.

[53] Iosifidis, G. , Gao, L. , Huang, J. , and Tassiulas, L. . Incentive mechanisms for user-provided networks. *IEEE Communications Magazine*, 52(9):20–27, 2014.

[54] Iosifidis, G. , Gao, L. , Huang, J. , and Tassiulas, L. . Efficient and fair collaborative mobile internet access. *IEEE/ACM Transactions on Networking*, 25(3):1386–1400, 2017.

[55] Iosifidis, G. , Gao, L. , Huang, J. , and Tassiulas, L. . Efficient and fair collaborative mobile internet access. *IEEE/ACM Transactions on Networking*, 25(3):1386–1400, 2017.

[56] Jia, A. L. , Shen, S. , Epema, D. H. , and Iosup, A. . When game becomes life: The creators and spectators of online game replays and live streaming. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 12(4):47, 2016.

[57] Jiang, J. , Zhang, S. , Li, B. , and Li, B. . Maximized cellular traffic offloading via device-to-device content sharing. *IEEE Journal on Selected Areas in Communications*, 34(1):82–91, 2016.

[58] Joseph, V. and Veciana, G. de . Nova: Qoe-driven optimization of dash-based video delivery in networks. In *IEEE International Conference on Computer Communications*, pages 82–90, 2014.

[59] Kang, X. and Wu, Y. . Incentive mechanism design for heterogeneous peer-to-peer networks: A stackelberg game approach. *IEEE Transactions on Mobile Computing*, 14(5):1018–1030, 2015.

[60] Keller, L. , Le, A. , Cici, B. , Seferoglu, H. , Fragouli, C. , and Markopoulou, A. . Microcast: Cooperative video streaming on smartphones. In *ACM International Conference on Mobile Systems, Applications, and Services*, pages 57–70, 2012.

[61] Khan, M. A. and Sun, Y. . Non-cooperative games with many players. *Handbook of game theory with economic applications*, 3:1761–1808, 2002.

[62] Klusch, M. , Kapahnke, P. , Cao, X. , Rainer, B. , Timmerer, C. , and Mangold, S. . Mymedia: mobile semantic peer-to-peer video search and live streaming. In *International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, pages 277–286, 2014.

[63] Kyparisis, J. . On uniqueness of kuhn-tucker multipliers in nonlinear programming. *Mathematical Programming*, 32(2):242–246, 1985.

[64] Li, J. , Bhattacharyya, R. , Paul, S. , Shakkottai, S. , and Subramanian, V. . Incentivizing sharing in realtime d2d streaming networks: A mean field game perspective. *IEEE/ACM Transactions on Networking*, 25(1):3–17, 2017.

[65] Li, Z. , Zhu, X. , Gahm, J. , Pan, R. , Hu, H. , Begen, A. C. , and Oran, D. . Probe and adapt: Rate adaptation for http video streaming at scale. *IEEE Journal on Selected Areas in Communications*, 32(4):719–733, 2014.

[66] Masry, M. , Hemami, S. S. , and Sermadevi, Y. . A scalable wavelet-based video distortion metric and applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(2):260–273, 2006.

[67] Meeker, M. . Internet trends report 2018. `https://www.kleinerperkins.com/perspectives/internet-trends-report-2018`, 2018.

[68] Militano, L. , Orsino, A. , Araniti, G. , Molinaro, A. , and Iera, A. . A constrained coalition formation game for multihop d2d content uploading. *IEEE Transactions on Wireless Communications*, 15(3):2012–2024, 2016.

[69] Nash Jr, J. F. . The bargaining problem. *Econometrica: Journal of the Econometric Society*, pages 155–162, 1950.

[70] Neely, M. J. . Stochastic network optimization with application to communication and queueing systems. *Synthesis Lectures on Communication Networks*, 3(1):1–211, 2010.

[71] Olson, D. L. . *Decision aids for selection problems.* Springer Science & Business Media, 1996.

[72] Pantos, R. and May, W. . Http live streaming: draft-pantos-http-live-streaming-06. *The Internet Engineering Task Force*, 24, 2011.

[73] Perrucci, G. P. , Fitzek, F. H. , and Widmer, J. . Survey on energy consumption entities on the smartphone platform. In *IEEE Vehicular Technology Conference*, pages 1–6, 2011.

[74] Pu, W. , Zou, Z. , and Chen, C. W. . Video adaptation proxy for wireless dynamic adaptive streaming over http. In *IEEE Packet Video Workshop*, pages 65–70, 2012.

[75] Rainer, B. , Timmerer, C. , Kapahnke, P. , and Klusch, M. . Real-time multimedia streaming in unstructured peer-to-peer networks. In *IEEE Consumer Communications and Networking Conference*, pages 1136–1137, 2014.

[76] Sandholm, W. H. . Potential games with continuous player sets. *Journal of Economic theory*, 97(1):81–108, 2001.

[77] Scheibe, K. , Fietkiewicz, K. J. , and Stock, W. G. . Information behavior on social live streaming services. *Journal of Information Science Theory and Practice*, 4(2):6–20, 2016.

[78] Schmeidler, D. . Equilibrium points of nonatomic games. *Journal of statistical Physics*, 7(4):295–300, 1973.

[79] Schrijver, A. . *Theory of linear and integer programming.* John Wiley & Sons, 1998.

[80] Schrijver, A. . *Theory of linear and integer programming.* John Wiley & Sons, 1998.

[81] Seenivasan, T. V. and Claypool, M. . Cstream: neighborhood bandwidth aggregation for better video streaming. *Multimedia Tools and Applications*, 70(1):379–408, 2014.

[82] Spiteri, K. , Urgaonkar, R. , and Sitaraman, R. K. . Bola: Near-optimal bitrate adaptation for online videos. In *IEEE International Conference on Computer Communications*, pages 1–9, 2016.

[83] Stockhammer, T. . Dynamic adaptive streaming over http–: standards and design principles. In *ACM Conference on Multimedia systems*, pages 133–144, 2011.

[84] Stojmenovic, I. and Wen, S. . The fog computing paradigm: Scenarios and security issues. In *IEEE Federated Conference on Computer Science and Information Systems*, pages 1–8, 2014.

[85] Syrivelis, D. , Iosifidis, G. , Delimpasis, D. , Chounos, K. , Korakis, T. , and Tassiulas, L. . Bits and coins: Supporting collaborative consumption of mobile internet. In *IEEE Conference on Computer Communications*, pages 2146–2154, 2015.

[86] Tang, M. , Gao, L. , Pang, H. , Huang, J. , and Sun, L. . Optimizations and economics of crowdsourced mobile streaming. *IEEE Communications Magazine*, 55(4):21–27, 2017.

[87] Tang, P. , Zeng, Y. , and Zuo, S. . Fans economy and all-pay auctions with proportional allocations. In *Association for the Advancement of Artificial Intelligence*, pages 713–719, 2017.

[88] Tian, G. and Liu, Y. . Towards agile and smooth video adaptation in dynamic http streaming. In *ACM Emerging networking experiments and technologies*, pages 109–120, 2012.

[89] Varsa, V. and Curcio, I. . Transparent end-to-end packet switched streaming service (pss); rtp usage model (release 5). *3GPP TR 26.937 V1. 4.0*, 2003.

[90] Vickrey, W. . Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance*, 16(1):8–37, 1961.

[91] Xing, M. , Xiang, S. , and Cai, L. . Rate adaptation strategy for video streaming over multiple wireless access networks. In *IEEE Global Communications Conference*, pages 5745–5750, 2012.

[92] Xu, C. , Zhao, F. , Guan, J. , Zhang, H. , and Muntean, G.-M. . Qoe-driven user-centric vod services in urban multihomed p2p-based vehicular networks. *IEEE Transactions on Vehicular Technology*, 62(5):2273–2289, 2013.

[93] Yin, X. , Jindal, A. , Sekar, V. , and Sinopoli, B. . A control-theoretic approach for dynamic adaptive video streaming over http. In *ACM SIGCOMM Computer Communication Review*, volume 45, pages 325–338, 2015.

[94] Yin, X. , Sekar, V. , and Sinopoli, B. . Toward a principled framework to design dynamic adaptive streaming algorithms over http. In *ACM Workshop on Hot Topics in Networks*, page 9, 2014.

[95] Yuan, P. and Ma, H. . Opportunistic forwarding with hotspot entropy. In *IEEE World of Wireless, Mobile and Multimedia Networks*, pages 1–9, 2013.

[96] Zhang, M. , Gao, L. , Huang, J. , and Honig, M. . Cooperative and competitive operator pricing for mobile crowdsourced internet access. In *IEEE Conference on Computer Communications, IEEE*, pages 1–9, 2017.

[97] Zhang, W. , Wen, Y. , Chen, Z. , and Khisti, A. . Qoe-driven cache management for http adaptive bit rate streaming over wireless networks. *IEEE Transactions on Multimedia*, 15(6):1431–1445, 2013.

[98] Zhang, Y. , Li, C. , and Sun, L. . Decomod: collaborative dash with download enhancing based on multiple mobile devices cooperation. In *ACM Multimedia Systems Conference*, pages 160–163, 2014.

[99] Zhong, M. , Hu, P. , Indulska, J. , and Kumar, M. J. . Colstream: collaborative streaming of on-demand videos for mobile devices. In *IEEE International Symposium on World of Wireless, Mobile and Multimedia Networks*, pages 1–7, 2014.

[100] Zhou, C. , Lin, C.-W. , and Guo, Z. . mdash: A markov decision-based rate adaptation approach for dynamic http streaming. *IEEE Transactions on Multimedia*, 18(4):738–751, 2016.

[101] Zhu, Z. , Yang, Z. , and Dai, Y. . Understanding the gift-sending interaction on live-streaming video websites. In *International Conference on Social Computing and Social Media*, pages 274–285. Springer, 2017.