Multi-User Cooperative Mobile Video Streaming: Performance Analysis and Online Mechanism Design

Lin Gao, *Senior Member, IEEE*, Ming Tang, *Student Member, IEEE*, Haitian Pang [ID], *Student Member, IEEE*, Jianwei Huang [ID], *Fellow, IEEE*, and Lifeng Sun [ID], *Member, IEEE*

Abstract—Adaptive bitrate streaming enables video users to *adapt* their playing bitrates to the real-time network conditions, hence achieving the desirable quality-of-experience (QoE). In a multi-user wireless scenario, however, existing single-user based bitrate adaptation methods may fail to provide the desirable QoE, due to lack of consideration of multi-user interactions (such as the multi-user interferences and network congestion). In this work, we propose a novel user cooperation framework based on *user-provided networking* for multi-user mobile video streaming over wireless cellular networks. The framework enables nearby mobile video users to crowdsource their cellular links and resources for cooperative video streaming. We first analyze the social welfare performance bound of the proposed cooperative streaming system by introducing a virtual time-slotted system. Then, we design a low complexity Lyapunov-based online algorithm, which can be implemented in an online and distributed manner without the complete future and global network information. Numerical results show that the proposed online algorithm achieves an average 97 percent of the theoretical maximum social welfare. We further conduct experiments with real data traces, to compare our proposed online algorithm with the existing online algorithms in the literature. Experiment results show that our algorithm outperforms the existing algorithms in terms of both the achievable bitrate (with an average gain of $20 \sim 30$ percent) and social welfare (with an average gain of $10 \sim 50$ percent).

Index Terms—Mobile video streaming, adaptive bitrate, mobile crowdsourcing, online algorithm

◆

1 INTRODUCTION

1.1 Background and Motivations

GLOBAL mobile data traffic is growing at an unprecedented rate, where mobile video streaming contributes most of the data growth. According to Cisco [1], mobile video streaming traffic has accounted for 60 percent of the global mobile data traffic in 2016, and the percentage is expected to increase to 78 percent by 2021. *Adaptive BitRate (ABR)* streaming [2] is a promising technology for video streaming over large distributed HTTP networks (e.g., Internet) and has been adopted by many popular online video streaming systems (e.g., HTTP dynamic streaming of Adobe [3], HTTP live streaming of Apple [4], and smooth streaming of Microsoft [5]). The key idea of ABR is to enable video players to *adapt* the playing bitrate (corresponding to the quality

- L. Gao is with the School of Electronic and Information Engineering, Harbin Institute of Technology, Shenzhen, China, 518055.
  E-mail: gaol@hit.edu.cn.
- M. Tang and J. Huang are with the Network Communications and Economics Lab (NCEL), Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong, 999077.
  E-mail: {mtang, jwhuang}@ie.cuhk.edu.hk.
- H. Pang and L. Sun are with the Department of Computer Science and Technology, Tsinghua University, Beijing, China, 100084.
  E-mail: pht14@mails.tsinghua.edu.cn, sunlf@tsinghua.edu.cn.

of video, e.g., resolution) to the real-time network conditions to ensure the desirable quality-of-experience (QoE).

While most of the existing works focused on the bitrate adaptation methods of a *single user* (e.g., [6], [7]), in this work we consider a more general scenario of *multi-user* video streaming over wireless cellular networks. In the multi-user wireless scenario, the QoE of each mobile user is affected not only by the stochastic changing of his own network condition (e.g., channel fading), but also by the potential resource competition and interference of other users [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19]. Without proper coordination or cooperation among users, such competition and interference may degrade the network performance greatly (e.g., leading to congestion), hence increase the video streaming cost and harm the user QoE. Thus, the existing single-user based bitrate adaptation methods often fail to provide desirable QoE for video users in the multi-user scenario, due to lack of consideration of multi-user competition and interference.

To this end, in this work we will study the multi-user *cooperative* video streaming, where (nearby) mobile video users cooperate with each other in both bitrate adapting and video downloading. Namely, each user can download video data for other users using his own cellular link or download his video data through others' links. In this sense, users *aggregate* their cellular links and resources for the cooperative video streaming. Fig. 1 illustrates such a cooperative streaming model with three users {1, 2, 3}, where user 1 downloads for all three users and user 2 downloads for himself and user 3. Note that user 3 does not have the available cellular link.

Fig. 1. Cooperative video streaming model.

There are several real-world scenarios where the multi-user cooperative streaming is useful and helpful. First, the most relevant scenario is *User-Provided Networking* (UPN) [20], [21], where mobile devices (e.g., 4G smartphones) with abundant cellular link capacities operate as mobile hotspots and provide Internet access to other devices. UPN has been widely studied and implemented today, and some IT and Telecom companies (such as OpenGarden [22], Karma [23], and AT&T [24]) have provided commercial UPN services. The cooperative streaming proposed in this work can enhance the capability of UPN on providing the video streaming service. Another more concrete scenario is *Mobile Live Streaming* (MLS) [25], [26], with which people can watch live activities of their friends or share their own activities to their friends on their smartphones. MLS becomes popular in recent years with the proliferation of smartphones and 4G cellular networks. Nowadays, many social network companies have provided MLS services, such as IngKee [27], Youtube Live [28], and Facebook Livestream [29]. The cooperative streaming proposed in this work can improve the live streaming quality in MLS.

The key motivation for considering such a cooperative streaming system is the *heterogeneity* of mobile devices.[1] Note that cooperative streaming can be easily implemented in a practical scenario (such as UPN and MLS) by installing some customized apps (e.g., OpenGarden [22]) on smartphones, and the related optimization and incentive issues have been studied in the recent literature (e.g., [20], [21]). However, the existing techniques in [20], [21] cannot be directly applied to the cooperative video streaming model, due to the asynchronous operations of video streaming and the unique QoE requirements of video applications. This motivates us to study the multi-user cooperative streaming in this work.

1.2 Solution and Contributions

In this work, we propose a general *multi-user cooperative video streaming* framework based on UPN [20], [21]. The key idea is to enable nearby mobile users to form a cooperative group (via WiFi or Bluetooth) and aggregate their cellular links and resources for the cooperative video downloading and bitrate adapting. We focus on studying the users' streaming

behaviours (i.e., *download scheduling* and *bitrate adaptation*) in the proposed cooperative framework. Namely, for each video user, when and from whom he is going to download each video segment, at which bitrate? Our goal is to understand the performance bound of the system and design an online scheduling method to approach such a bound.

First, *we formally define the users' operations in the cooperative streaming system, and formulate the corresponding social welfare optimization problem (Section 4).* The optimal solution of this problem provides the theoretical performance bound of the proposed cooperative streaming system. A comprehensive analysis for such a performance bound is the foundation of the future study on privacy, security, and incentive mechanism design.[2]

Second, *we analyze the social welfare performance bound of the proposed cooperative streaming system (Section 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 solution of the virtual time-slotted system can provide an effective upper-bound and lower-bound for the optimal solution (performance bound) of the original system, which forms the feasible performance region of the proposed cooperative streaming system.

Finally, *we design a Lyapunov-based online streaming algorithm for the practical implementation of the proposed cooperative streaming system (Section 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 more clarity, we summarize the logical relationship among the above three parts as follows: (i) *the social welfare optimization problem in Section 4 defines the theoretical performance bound of the proposed cooperative streaming system (but it is challenging to solve);* (ii) *the virtual time-slotted system in Section 5 helps characterize the region (i.e., upper-bound and lower-bound) of the above theoretical performance bound;* (iii) *the online algorithm in Section 6 converges to the above theoretical performance bound asymptotically in the realistic scenario without complete future network information.* More specifically, the key contributions of this work are summarized as follows.

---

1. According to [1], smartphones only account for 38 percent of the total mobile devices, and a large amount of non-smartphone mobile devices (e.g., tablets and laptops) still lack stable and always-on Internet connections, especially in the outdoor environment. Our proposed system can help these devices connect to the Internet via the links of nearby smartphones. Furthermore, even for the smartphone devices with the same or similar Internet capability, they may have different cost evaluations for energy consumption (depending on, for example, their battery status), resulting in certain "heterogeneity" among devices.

2. In practice, there have been various existing approaches to address the privacy/security issue in the similar systems. For example, Opengarden [22] solves the privacy/security issue by using advanced data encryption technique through built-in softwares, while Karma [23] solves it by using hardware-enabled data encryption technique through dedicated devices. These existing technical ways can help us quickly build the privacy and security system.

- *Novel Model:* 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 radio connections and resources for cooperative video streaming, and can effectively improve the QoE of video users. Moreover, we provide both theoretical performance analysis and practical algorithm design for such a cooperative streaming system.
- *Performance Bound Analysis:* We analyze the theoretical performance bound of the proposed cooperative streaming system, overcoming the challenging issue of asynchronous operations by using a virtual time-slotted system. Such a performance bound analysis is fundamental for the design, evaluation, and implementation of practical algorithms in such a cooperative streaming system.
- *Online Algorithm Design:* 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.
- *Experiment and Demo:* We conduct extensive experiments with real data traces, which show that our proposed cooperative streaming system, together with the online streaming algorithm, outperforms the existing systems and algorithms in terms of both achieved bitrate (with an average gain of $20 \sim 30$ percent) and social welfare (with an average gain of $10 \sim 50$ percent). We also construct a real demo system to implement and evaluate the proposed system and algorithm.

The rest of the paper is organized as follows. In Section 2, we review the related work. In Section 3, we present the system model. In Section 4, we provide the problem formulation. In Section 5, we propose the virtual time-slotted system and the performance bound analysis. In Section 6, we propose the Lyapunov-based online streaming algorithm. We provide simulation results in Section 7 and conclude in Section 8.

2 LITERATURE REVIEW

Prior works on ABR video streaming mainly focused on the bitrate adaptation of a *single user* using either buffer-based method [6] or channel prediction-based method [7]. Recently, there is a growing interest in exploiting the *multi-user* cooperative video streaming. From the modeling perspective, the existing cooperative streaming models can be classified into four categories (see [8] for more details): *Bandwidth Aggregation (BA)* model [9], *Device-to-Device (D2D)* model [10], [11], [12], *Crowdsourced Mobile Streaming (CMS)* model [13], [14], and *Mobile Peer-to-Peer (MP2P)* model [17], [18], [19].

1) BA Model [9]: The key idea is to aggregate the bandwidth of nearby users to help a particular mobile video user's streaming. The BA model mainly focused on the simple *one-to-many* cooperation between a single video user and multiple helpers [9]. We consider a more general *many-to-many* cooperation framework with multiple video users and multiple helpers, where each user acts as both the video user and the helper.

2) D2D Model [10], [11], [12]: The key idea is to enable nearby video users to share their downloaded video segments with each other through D2D links. In [10], Golrezaei et al. studied the cache-based D2D cooperation, where mobile video users cache popular video contents and deliver to other users via D2D links in the future. Our model differs from that of [10] in the following aspects. First, we consider the real-time cooperation of nearby users, while they considered the future opportunistic cooperation. Second, we study the jointly video streaming of multiple users, while they studied the video streaming of different users separately. In [11], [12], researchers studied the real-time D2D based cooperation, where multiple nearby users watch the *same* video and share video contents cooperatively via D2D links. Our model is similar but more general than those in [11], [12], as we allow different users to watch different videos. This introduces an additional dimension (i.e., video index) when making the scheduling decision, hence involves additional challenges.

3) CMS Model [13], [14], [15], [16]: The key idea is to enable nearby mobile video users pool their network resources together to satisfy all users' video streaming requirements jointly. Note that our proposed cooperative streaming model falls into this category. In [13], Pu et al. proposed a rate adaptation algorithm for optimizing the adaptive streaming across multiple mobile users (possibly watching different videos), but they didn't consider the individual characteristics of different users. In [14], [15], Tang et al. focused on the incentive design in the multi-user CMS model and proposed a multi-dimensional auction-based mechanism to incentivize video users to collaborate with each other under information asymmetry. However, they neither performed the performance bound analysis, nor designed the online algorithm. In [16], Gao et al. analyzed the performance bound for multi-user CMS models, but didn't consider the online algorithm design. In this work, we will study both the theoretical performance bound and the practical online algorithm systematically.

4) MP2P Model [17], [18], [19]: The key idea is to enable video users act as virtual video servers and send the downloaded segments to other users via *Internet*. Thus, in the MP2P model, a video user can potentially help other users that are not physically close-by. The key difference between our model and the MP2P model is as follows. In the MP2P model, each video segment has multiple copies residing on both the video server and the user devices (peers), and video users can download a video segment from either the server or a user peer, via his own wireless cellular link. Hence, the key design purpose of MP2P model is to *reduce the load of the video server*. In our cooperative 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 our model is to *reduce the uncertainty* or *improve the efficiency of user's wireless cellular link*.

3 SYSTEM MODEL

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.

There are many real-world application scenarios for such a cooperative video streaming. Consider, for example, that a group of friends who want to watch a live soccer match together on their phones at a remote location (e.g., a camping or skiing site), or a family who wants to watch one or multiple movies on their phones in the train or in the car, or a group of students who want to watch different online lectures using WiFi at a busy hotspot (e.g., a classroom). In all these cases, some or all of the users may have poor or intermittent cellular connectivity, depending on the coverage of their service providers. Thus, aggregating the resources of nearby users for the cooperative video streaming may significantly improve the overall user satisfactions.

1) User-Provided Network (UPN): UPN enables nearby mobile users to form a cooperative group (via WiFi) and aggregate their radio connections and resources for cooperative data downloading. We consider a general *multi-user cooperative streaming* scheme based on UPN. Namely, in a cooperative group, each user can download video data for other users using his own cellular link (and resources) and download his video data through other users' links (and resources). As mentioned previously, we assume that some well-designed incentive mechanisms (e.g., auction [30], [31], [32], [33], contract [34], [35], [36], [37], or others trust mechanisms [38], [39]) have been adopted, such that users are willing to participate in the cooperative streaming system to help others.

Fig. 1 illustrates such a cooperative streaming model with three users {1, 2, 3}, where user 1 downloads one segment for himself, one segment for user 2, and two segments for user 3, while user 2 downloads two segments for himself and one segment for user 3. Note that user 3 does not download any video content due to the temporary interruption of his cellular link.

2) Mobility Model: The cooperation gain of such a cooperative streaming highly depends on the number of cooperative users and the duration of cooperation, both closely related to the users' mobility patterns. We adopt a *hotspot-based* mobility model [40], where the whole area is divided into a set of small hotspots and the non-hotspot area,[3] 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).* Fig. 2 illustrates such a mobility model,
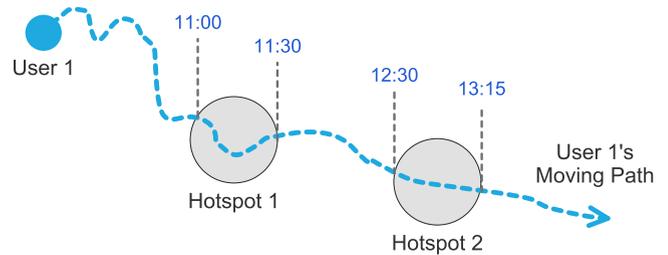
Fig. 2. Hotspot-based mobility model.

where user 1 stays at hotspot 1 for 30 minutes (11:00 $\sim$ 11:30), and then takes 1 hour to move to hotspot 2 and stays at hotspot 2 for 45 minutes (12:30 $\sim$ 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 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 [41]).

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 $a_n(t) = a_m(t) \in \mathcal{A}$.

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 $e_{n,m}(t)$ can be derived from the location information of users n and m.

3.2 Video Streaming Model

We consider a typical ABR streaming model [2], 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). 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).

---

3. 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 for a reasonable amount of time.

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

- $\beta_n > 0$: segment length (in seconds) of user n's video;
- $\mathcal{R}_n \triangleq \{R_n^1, R_n^2, \ldots, R_n^Z\}$ (with $0 < R_n^1 < R_n^2 < \ldots < R_n^Z$): 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.
- $Q_n > 0$: maximum buffer size (in seconds) of user n.

4 PROBLEM FORMULATION

In this section, we first characterize the users' behaviours in the cooperative streaming model, and then 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*.

4.1 Downloading Sequence

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

$$S_n \triangleq \{s_{n[1]}, \; s_{n[2]}, \ldots, s_{n[k]}, \ldots\}, \tag{1}$$

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

$$s_{n[k]} = (u, \; z, \; r, \; [t^s, t^e]).$$

To distinguish the information of different segments, we will also write the information of segment $s_{n[k]}$ as $(u_{n[k]}, z_{n[k]}, r_{n[k]}, t^s_{n[k]}, t^e_{n[k]})$ whenever needed.

It is easy to see that our cooperative 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 S_n of user n.

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

$$\text{C.1}: \quad t^e_{n[k]} \leq t^s_{n[k+1]}, \quad \forall k = 1, \ldots, |S_n|.$$

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

(ii) *Capacity Constraint:* Each segment $s_{n[k]} = (u, z, r, [t^s, t^e])$ consists of $r \cdot \beta_u$ Mbits of video data, and is downloaded by user n within time interval $[t^s_{n[k]}, t^e_{n[k]}]$. Hence, we have the following cellular link capacity constraint:

$$\text{C.2}: \quad r \cdot \beta_u \leq \int_{t^s_{n[k]}}^{t^e_{n[k]}} h_n(t)\mathrm{d}t, \quad \forall k = 1, \ldots, |S_n|,$$

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 $s_{n[k]} = (u, z, r, [t^s, t^e])$, $n \neq u$ is feasible only if users n and u are encountered during $[t^s_{n[k]}, \; t^e_{n[k]}]$, i.e.,

$$\text{C.3}: \quad e_{n,u}(t) = 1, \; t \in \left[t^s_{n[k]}, \; t^e_{n[k]}\right], \quad \forall k = 1, \ldots, |S_n|.$$

4.2 Receiving Sequence

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

$$\widehat{S}_m = \bigcup_{n \in \mathcal{N}, k \in \{1, \ldots, |S_n|\} : u_{n[k]} = m} \left\{s_{n[k]}\right\}. \tag{2}$$

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

$$\widehat{S}_m \triangleq \{\hat{s}_{m[1]}, \; \hat{s}_{m[2]}, \; \ldots, \; \hat{s}_{m[k]}, \; \ldots\}, \tag{3}$$

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

It is easy to see that $\hat{u}_{m[k]} = m$ for all $\hat{s}_{m[k]} \in \widehat{S}_m$. To facilitate the later analysis, we further assume that $\hat{t}^e_{m[k]} \leq \hat{t}^e_{m[k+1]}$, $\forall k = 1, \ldots, |\widehat{S}_m|$, that is, user m receives the segments in $\widehat{S}_m$ sequentially. Note that this can always be achieved by a proper schedule of downloading sequences with the full network information. For example, if $\hat{t}^e_{m[k]} > \hat{t}^e_{m[k+1]}$, i.e., the $k+1$th segment is received before the kth segment, we can simply change their downloading orders.

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 $q_{m[k]}$ denote the buffer level (in seconds) of user m *when receiving the kth segment*, i.e., at the time $\hat{t}^e_{m[k]}$. Then, we have the following *buffer update rule* for user m:

$$q_{m[k]} = \left[q_{m[k-1]} - \left(\hat{t}^e_{m[k]} - \hat{t}^e_{m[k-1]}\right)\right]^+ + \beta_m, \tag{4}$$

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

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

$$\text{C.4}: \quad 0 \leq q_{m[k]} \leq Q_m, \quad \forall k = 1, \ldots, |\widehat{S}_m|.$$

---

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

5. We assume the WiFi transmission time is zero. One motivation for such an assumption is that the recent IEEE 802.11 standard family (e.g., 802.11n, ac, and ad) has become increasingly powerful, and can support a data rate up to Gbit/s, which is much higher than those of the current cellular systems (such as 3G and 4G).

4.3 User Welfare

The welfare 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 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 $g_n(r)$ denote the value that user n achieves from bitrate r during one unit of playback time. Then, the total value that user n achieves from all received segments $\widehat{S}_n$ (each with a playback time of $\beta_{\hat{u}_{n[k]}} = \beta_n$) is:

$$V_n(\widehat{S}_n) \triangleq \sum_{k=1}^{|\widehat{S}_n|} g_n(\hat{r}_{n[k]}) \cdot \beta_n. \tag{5}$$

Obviously, $g_n(\cdot)$ is an increasing function (as video quality monotonically increases with bitrate). In our simulations, we adopt the following value function: $g_n(r) = \log(1 + \theta_n \cdot r)$, 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 work, 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. Let $\phi_n^{\text{QDEG}} > 0$ 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[6]

$$L_n^{\text{QDEG}}(\widehat{S}_n) \triangleq \sum_{k=2}^{|\widehat{S}_n|} \phi_n^{\text{QDEG}} \cdot \left[\hat{r}_{n[k-1]} - \hat{r}_{n[k]}\right]^+, \tag{6}$$

where $[x]^+ = \max\{0, x\}$. Here $\hat{r}_{n[k-1]} > \hat{r}_{n[k]}$ indicates that a quality degradation occurs between $\hat{s}_n[k-1]$ and $\hat{s}_{n[k]}$, 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 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 Eq. (4), a rebuffering occurs when

$$q_{n[k-1]} < \hat{t}_{n[k]}^{\text{e}} - \hat{t}_{n[k-1]}^{\text{e}},$$

with a detailed rebuffering time $\hat{t}_{n[k]}^{\text{e}} - \hat{t}_{n[k-1]}^{\text{e}} - q_{n[k-1]}$. Let $\phi_n^{\text{REBUF}} > 0$ 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

$$L_n^{\text{REBUF}}(\widehat{S}_n) \triangleq \sum_{k=2}^{|\widehat{S}_n|} \phi_n^{\text{REBUF}} \cdot \left[\hat{t}_{n[k]}^{\text{e}} - \hat{t}_{n[k-1]}^{\text{e}} - q_{n[k-1]}\right]^+. \tag{7}$$

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

$$U_n(\widehat{S}_n) \triangleq V_n(\widehat{S}_n) - L_n^{\text{QDEG}}(\widehat{S}_n) - L_n^{\text{REBUF}}(\widehat{S}_n). \tag{8}$$

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 Cellular 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 [42]. Let $c_n^{\text{TIME}} \geq 0$ denote the time-related energy consumption factor of user n (i.e., for each unit of downloading time), and $c_n^{\text{DATA}} \geq 0$ 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 [42]:

$$E_n^{\text{CELL}}(S_n) \triangleq \sum_{k=1}^{|S_n|} \left(c_n^{\text{TIME}} \cdot (t_{n[k]}^{\text{e}} - t_{n[k]}^{\text{s}}) + c_n^{\text{DATA}} \cdot r_{n[k]} \cdot \beta_{u_{n[k]}}\right). \tag{9}$$

(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 [42]. Let $w_n^{\text{TIME}} \geq 0$ and $w_n^{\text{DATA}} \geq 0$ 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 [42]:

$$E_n^{\text{WIFI}}(S_n) \triangleq \sum_{k=1}^{|S_n|} \left(w_n^{\text{TIME}} \cdot 0 + w_n^{\text{DATA}} \cdot r_{n[k]} \cdot \beta_{u_{n[k]}}\right) \\ \cdot \mathbf{1}(u_{n[k]} \neq n), \tag{10}$$

where $\mathbf{1}(u_{n[k]} \neq n) = 1$ if $u_{n[k]} \neq n$ (i.e., the segment $s_{n[k]}$ 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 S_n and receiving sequence $\widehat{S}_n$ as follows:

$$C_n(S_n, \widehat{S}_n) \triangleq E_n^{\text{CELL}}(S_n) + E_n^{\text{WIFI}}(S_n). \tag{11}$$

Note that our proposed system can work with other energy models (e.g., those in [43]). In fact, the energy consumption modeling and energy saving are not the key objective of the proposed system. Instead, our key objective is to improve the QoE of users.

3) Welfare: The welfare 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.,

$$P_n(S_n, \widehat{S}_n) \triangleq U_n(\widehat{S}_n) - C_n(S_n, \widehat{S}_n). \tag{12}$$

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

$$W(S_1, \ldots, S_N) \triangleq \sum_{n=1}^{N} P_n(S_n, \widehat{S}_n), \tag{13}$$

---

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

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

4.4 Problem Formulation

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

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

$$\max_{\{S_n, n \in \mathcal{N}\}} \quad W(S_1, \ldots, S_N), \tag{14}$$
$$\text{s.t.} \quad \text{C.1} \sim \text{C.4.}$$

To solve this *offline* optimization problem, we need to know the complete network information. The solution of Eq. (14), denoted by W^*, provides the theoretical performance bound (in term of social welfare) of the proposed cooperative streaming system. Note that Eq. (14) is an MILP (mixed integer linear programming) and challenging to solve.[7] Hence, we will derive a feasible upper-bound and a feasible lower-bound of W^* in Section 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 6.

5 PERFORMANCE BOUND ANALYSIS

In this section, we study the theoretical social welfare performance bound of the proposed cooperative system (i.e., the solution of the offline social welfare maximization problem Eq. (14)), which serves as a benchmark for the online scheduling solutions in Section 6.

However, directly solving Eq. (14) 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, Eq. (14) involves both discrete variables (e.g., u and z) and continuous variables (e.g., t^s and t^e), hence is a complicated mixed-integral optimization problem. Third, Eq. (14) involves the integral operation (C.2), which is even more challenging. Hence, we will focus on finding upper-bound and lower-bound for the desired performance bound of the cooperative streaming 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 Eq. (14) under the segmented operation scheme (i.e., the theoretical performance bound of the proposed cooperative streaming system) is bounded by the solutions under this virtual time-slotted system. *It is important to note that this time-slotted operation*
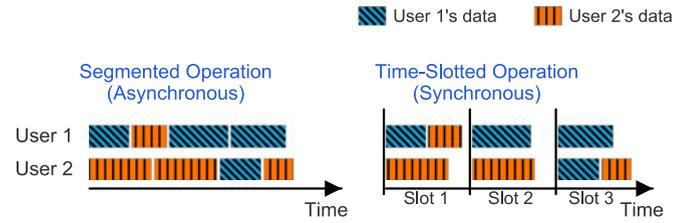
Fig. 3. Segmented versus time-slotted operation.

scheme is only used for characterizing the theoretical performance bound, but not for the practical implementation.

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 $\mathcal{T} = \{1, 2, \ldots, T\}$, with the τth 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.

For clarity, we illustrate the difference (in download scheduling) between the segmented operation and the time-slotted operation in Fig. 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:

$$K_n \triangleq \left\{ \kappa_{n,m}^z(\tau), \quad \forall \tau \in \mathcal{T}, m \in \mathcal{N}, z \in \{1, \ldots, Z\} \right\}, \tag{15}$$

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 τ.

Given the downloading vector K_n, we can derive the total data that user n downloads in each time slot τ:

$$x_n^{\mathrm{DL}}(\tau) = \sum_{m=1}^{N} x_{n,m}(\tau) = \sum_{m=1}^{N} \sum_{z=1}^{Z} \kappa_{n,m}^z(\tau) \cdot \beta_m \cdot R_m^z, \tag{16}$$

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 K_n:

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

where $H_n(\tau) = \int_{\tau-1}^{\tau} h_n(t)dt$ is the total cellular link capacity of user n in time slot τ. 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., $K_n, \forall n \in N$, we can derive the total playback time that user m receives in each time slot τ:

$$y_m^{\text{RE}}(\tau) = \sum_{n=1}^{N} y_{n,m}(\tau) = \sum_{n=1}^{N} \sum_{z=1}^{Z} \kappa_{n,m}^z(\tau) \cdot \beta_m, \qquad (17)$$

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 τ.

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

$$q_m(\tau) = [q_m(\tau-1) - 1]^+ + y_m^{\text{RE}}(\tau), \qquad (18)$$

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

Similarly, we have the following buffer constraint:

$$\widetilde{\text{C}}.4: \quad 0 \leq q_m(\tau) \leq Q_m, \quad \forall \tau = 1, \ldots, T.$$

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

(i) *Video Quality:* Similar as Eq. (5), the value that user n achieves from all received segments is:

$$\widetilde{V}_n \triangleq \sum_{\tau=1}^{T} \sum_{m=1}^{N} \sum_{z=1}^{Z} \kappa_{m,n}^z(\tau) \cdot \beta_n \cdot g_n(R_n^z). \qquad (19)$$

(ii) *Quality Fluctuation:* Without loss of generality, we assume that all the received segments of each user n in each time slot τ are sorted in ascending order of bitrate.[8] Hence, quality degradation only occurs between two successive time slots, while never occurs within a time slot. Let $r_n^{\text{H}}(\tau)$ and $r_n^{\text{L}}(\tau)$ denote the highest bitrate and lowest bitrate that user n receives in slot τ. Then, similar as Eq. (6), the value loss of user n due to quality degradation is:

$$\widetilde{L}_n^{\text{QDEG}} \triangleq \sum_{\tau=2}^{T} \phi_n^{\text{QDEG}} \cdot \left[r_n^{\text{H}}(\tau-1) - r_n^{\text{L}}(\tau) \right]^+. \qquad (20)$$

(iii) *Rebuffering:* By the buffer update rule in Eq. (18), a rebuffering occurs in time slot τ when

$$q_m(\tau-1) < 1,$$

with a rebuffering time $1 - q_m(\tau-1)$. Then, similar as Eq. (7), the value loss of user n induced by rebuffering is

$$\widetilde{L}_n^{\text{REBUF}} \triangleq \sum_{\tau=2}^{T} \phi_n^{\text{REBUF}} \cdot [1 - q_m(\tau-1)]^+. \qquad (21)$$

(iv) *Energy Consumption for Video Downloading (via Cellular and Interent):* Similar as Eq. (9), the energy consumption of user n for downloading video is

$$\widetilde{E}_n^{\text{CELL}} \triangleq \sum_{\tau=1}^{T} \left(c_n^{\text{TIME}} \cdot \frac{x_n^{\text{DL}}(\tau)}{H_n(\tau)} + c_n^{\text{DATA}} \cdot x_n^{\text{DL}}(\tau) \right), \qquad (22)$$

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

(v) *Energy Consumption for Video Exchanging (via WiFi):* Similar as Eq. (10), the energy consumption of user n for exchanging video on local WiFi links is

$$\widetilde{E}_n^{\text{WIFI}} \triangleq \sum_{\tau=1}^{T} \sum_{m=1, m \neq n}^{N} \left(w_n^{\text{TIME}} \cdot 0 + w_n^{\text{DATA}} \cdot x_{n,m}(\tau) \right). \qquad (23)$$

Based on the above, the welfare of each user n is

$$\widetilde{P}_n(K_1, \ldots, K_N) \triangleq \widetilde{V}_n - \widetilde{L}_n^{\text{QDEG}} - \widetilde{L}_n^{\text{REBUF}} - \widetilde{E}_n^{\text{CELL}} - \widetilde{E}_n^{\text{WIFI}}. \qquad (24)$$

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

$$\max_{\{K_n, n \in \mathcal{N}\}} \quad \widetilde{W} \triangleq \sum_{n=1}^{N} \widetilde{P}_n(K_1, \ldots, K_N), \\ \text{s.t.} \quad \widetilde{\text{C}}.2 \sim \widetilde{\text{C}}.4. \qquad (25)$$

Similar to Eq. (14), this is an *offline* optimization problem and requires the complete network information. Moreover, Eq. (25) 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 Eq. (25) by $\widetilde{W}^*$.

5.2 Performance Bound

Now we characterize the theoretical performance bound W^* of the proposed cooperative streaming system (under the segmented operation) by using the solution $\widetilde{W}^*$ of Eq. (25) under the virtual time-slotted operation.

For convenience, we denote $\beta \triangleq (\beta_1, \ldots, \beta_N)$ as the vector consisting of all users' segment lengths, and denote $W^*_{(\beta)}$ and $\widetilde{W}^*_{(\beta)}$ as the solutions of Eqs. (14) and (25) under β, respectively. We refer to a vector β as an *integer multiple* of another vector β', if each element β_n in β is an integer multiple of the corresponding element β'_n in β'. For example, $\beta = (1, \ldots, N)$ is an integer multiple of $\beta' = (0.5, \ldots, N/2)$.

Proposition 1. *If β is an integer multiple of β', then*

$$W^*_{(\beta)} \leq W^*_{(\beta')}, \text{ and } \widetilde{W}^*_{(\beta)} \leq \widetilde{W}^*_{(\beta')}.$$

This proposition can be proved by showing that in both schemes, any downloading operation under β can be equivalently achieved under β'.

Proposition 2. *If $\beta \rightarrow 0$ (i.e., $\beta_n \rightarrow 0, \forall n \in \mathcal{N}$), then*

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

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

---

8. If not, we can simply change the orders of related segments.

Proposition 3. *If $\boldsymbol{\beta} \succeq \mathbf{0}$ is a finite vector (i.e., each element $\beta_n \geq 0$ is a finite number), then*

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

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 1. *Given a segment length $\boldsymbol{\beta}$, the theoretical performance upperbound $W^*_{(\boldsymbol{\beta})}$ is bounded by:*

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

Intuitively, this theorem states that with any $\boldsymbol{\beta}$, the theoretical performance bound $W^*_{(\boldsymbol{\beta})}$ of our proposed cooperative streaming 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 cooperative streaming system.

6   ONLINE SCHEDULING ALGORITHMS

In the previous section, we have analyzed the theoretical performance bound of the cooperative streaming 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 [44], which relies only on the current local network information and the scheduling history, while not on any future or global network information.

6.1   Online versus 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 Eq. (14), 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 cooperative streaming 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 [6]), the channel bandwidth or throughput predictions (e.g., in [7]), and other specific objective functions (e.g., Lyapunov drift-plus-penalty described below).

6.2   Lyapunov-Based Online Scheduling

Lyapunov optimization [44] 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.[9] 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 $z \in \{1, \ldots, Z\}$ 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 $q_m(t)$ 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 $R^z_u \cdot \beta_u$ (Mbit), and the *estimated* downloading time is $\gamma_{u,z} \triangleq \frac{R^z_u \cdot \beta_u}{h_n(t)}$.[10] The total energy consumption of user n (*for this particular downloading operation*) and user u (for playing the downloaded segment) is:

$$C_n(u, z) = E^{\text{CELL}}_n + E^{\text{WIFI}}_n.$$

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

$$U_u(u, z) = V_u - L^{\text{QDEG}}_u - L^{\text{REBUF}}_u.$$

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

$$U_m(u, z) = -L^{\text{REBUF}}_m = -\phi^{\text{REBUF}}_m \cdot \left[\gamma_{u,z} - q_m(t)\right]^+,$$

which only includes the potential rebuffering loss.

Therefore, the total welfare generated under (u, z) is

$$P(u, z) \triangleq \sum^N_{m=1} U_m(u, z) - C_n(u, z). \tag{26}$$

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

$$J(t) \triangleq \tfrac{1}{2} \sum^N_{m=1} [Q_m - q_m(t)]^2. \tag{27}$$

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

---

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

10. 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 stochastics.

$$\Delta(t) \triangleq J(t + \gamma_{u,z}) - J(t),$$
$$= \tfrac{1}{2} \sum_{m=1}^{N} \Big(\big[Q_m - q_m(t + \gamma_{u,z}) \big]^2 - \big[Q_m - q_m(t) \big]^2 \Big), \quad (28)$$

where $q_m(t + \gamma_{u,z})$ 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:

$$q_u(t + \gamma_{u,z}) = \min\{Q_u, \ [q_u(t) - \gamma_{u,z}]^+ + \beta_u\}.$$

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

$$q_m(t + \gamma_{u,z}) = [q_m(t) - \gamma_{u,z}]^+.$$

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*:

$$\Phi(t) \triangleq \Delta(t) - \lambda \cdot P(u, z), \quad (29)$$

where the negative welfare $(-P(u, z))$ 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 Eq. (7), which is the actually realized loss when a rebuffering event actually happens.

Algorithm 1. Lyapunov-Based Online Scheduling

---

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

Based on the above analysis, we now design an on-line algorithm that aims at minimizing the drift-plus-penalty Eq. (29) in each decision-making time. We present the detailed algorithm in Algorithm 1. 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*.

Note that the online scheduling in Algorithm 1 works in a distributed manner, as each user makes the decision independently. To coordinate the downloading decisions of different users and to avoid the redundant downloading of the same segment, nearby users need to exchange the context information (e.g., buffer length, segment size, encode bitrate, and url). To illustrate this, we construct a real demo system on Raspberry PI. Please refer to our online technical report [49] for more details.

4) Performance Analysis: Now we analyze the performance of Algorithm 1. Let $t_{[k]}$ denote the kth decision-making time

(counting all users), and let $P_{[k]}$ denote the associated welfare achieved in the kth download operation. Then, the social welfare generated by Algorithm 1 during the whole time $[0, T]$ can be computed by:

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

By the Lyapunov optimization theorem (Theorem 4.2 in [44]), we obtain the following gap for $W'_{(\boldsymbol{\beta})}$ and $W^*_{(\boldsymbol{\beta})}$, i.e., the theoretical performance upperbound.

Theorem 2.

$$\lim_{T \to \infty} \ \mathrm{E}\Big[W'_{(\boldsymbol{\beta})}\Big] \geq \mathrm{E}\Big[W^*_{(\boldsymbol{\beta})}\Big] - \frac{B}{\lambda},$$

where $\mathrm{E}[.]$ is expectation, and B is a positive constant.

Theorem 2 shows that Algorithm 1 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.,

$$\Big| W^*_{(\boldsymbol{\beta})} - W'_{(\boldsymbol{\beta})} \Big| \leq \Big| \widetilde{W}^*_{(\boldsymbol{\beta}' \to 0)} - W'_{(\boldsymbol{\beta})} \Big|.$$

Note that $\widetilde{W}^*_{(\boldsymbol{\beta}' \to 0)}$ is the solution of the integer programming problem Eq. (25), 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 $\widetilde{W}^*_{(\boldsymbol{\beta}' \to 0)}$ is smaller than 3 percent.

7 EXPERIMENTS AND PERFORMANCE

7.1 Experiment Setting

1) Datasets: To evaluate the realistic performance of the proposed cooperative streaming system, we conduct experiments based on real data traces from two datasets: ISF [46] and NWF [47].[11] Both datasets record 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), representing two different (hotspot-based) mobility scenarios: users encounter more frequently in NWF, while the duration of each encounter is larger in ISF.

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 [48], one of the largest OTT (Over The Top) video service providers in China. 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.

---

11. ISF is provided by a non-profit organization "Île Sans Fil" in Canada, and is open source (available at CRAWDAD [46]). NWF is obtained from a wireless service provider "NextWiFi" in China [47].
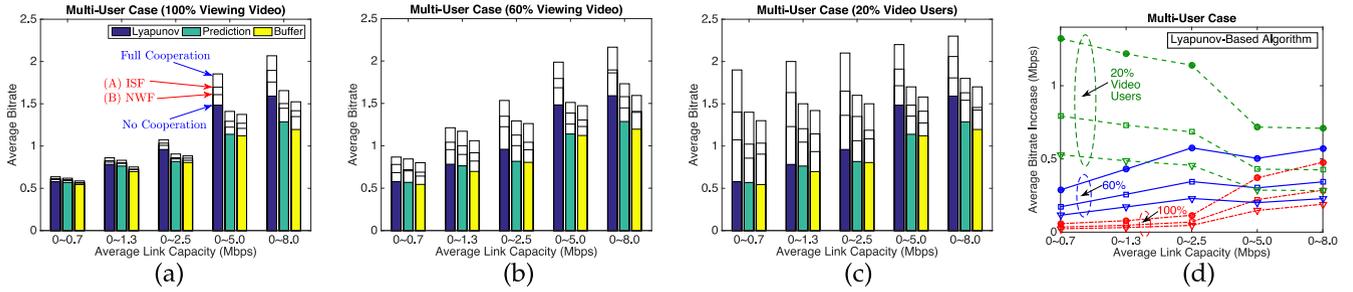
Fig. 4. (a) Average Bitrate with 100 percent Video Users. (b) Average Bitrate with 60 percent Video Users. (c) Average Bitrate with 20 percent Video Users. (d) Average Bitrate Increase under the Lyapunonv-based Algorithm.

Moreover, the energy consumption factors are chosen according to the real measurement given in [45].

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 [6] and Channel Prediction-based algorithm [7]. Specifically, buffer-based algorithm [6] 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 [7] 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.*[12]

7.2 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.[13] For simplicity, we assume that all video users play the high-resolution videos (bitrate 2.3 Mbps). The total video length is 500 seconds, the segment length is 2 seconds, and the maximum buffer length at the user's device is 40 seconds. We use these multi-user experiments to illustrate both the cooperation gain of the proposed cooperative streaming 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

---

12. Note that both algorithms in [6] and [7] 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 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) $q_n \geq \delta_{TH} \cdot Q_n$, (ii) $q_n - q_u \geq \Delta_{TH}$, and (iii) $q_u = \min_{m \in \mathcal{N}} q_m$.* 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 δ_{TH} and meanwhile is higher than the lowest buffer level by a threshold Δ_{TH}. In our experiments, we will try different values of δ_{TH} and Δ_{TH}, and choose the best ones.

13. We also construct experiments for the single-user scenario (i.e., non-cooperative scenario) to illustrate the performance gap of our proposed Lyapunov-based online algorithm to the theoretical performance bound as well as to compare the bitrate adaptation performance of our proposed algorithm with the existing online algorithms. The detailed results are provided in our online technical report [49].

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 0.7 Mbps.

1) Average Bitrate: Fig. 4 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 percent 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) using our proposed Lyapunov-based algorithm, comparing with the achieved bitrate under the non-cooperative scenario. The dash, solid, and dash-dot lines denote the results with 20, 60, and 100 percent video users, respectively. The marks "circle", "square", and "triangle" denote full cooperation, 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 percent), the increase of bitrate is very small under a low link capacity range (e.g., lower than 2.5 Mbps), 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 percent or 20 percent), 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 percent 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.
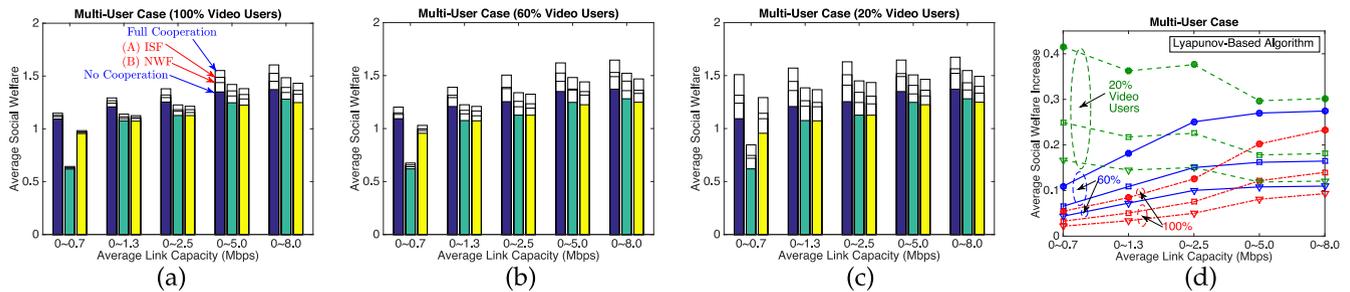
Fig. 5. (a) Social Welfare with 100 percent Video Users. (b) Social Welfare with 60 percent Video Users. (c) Social Welfare with 20 percent Video Users. (d) Average Social Welfare Increase under the Lyapunonv-based Algorithm.

With 20 percent 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 percent video users, the increase of bitrate first increases with the link capacity (due to a similar reason in the 100 percent case), and then decreases with the link capacity (due to a similar reason in the 20 percent case). The maximum bitrate increase ratio under the full cooperation scenario can be up to $50 \sim 230$ percent with 20 percent video users, $35 \sim 60$ percent with 60 percent video users, and $4 \sim 40$ percent with 100 percent video users. Moreover, the bitrate increase under the real data traces is bounded by the above maximum ratio, and actually depends on the encountering probability. In our experiments, the bitrate increases under ISF and NWF can reach around 60 and 40 percent of the maximum bitrate increase, respectively.

2) Social Welfare: Fig. 5 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 Fig. 4, 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 $20 \sim 40$ percent with 20 percent video users, $10 \sim 20$ percent with 60 percent video users, and $5 \sim 15$ percent with 100 percent video users. The social welfare increase under ISF and NWF can reach 60 and 40 percent of the maximum welfare increase, respectively.

3) Algorithm Comparison: From Figs. 4a, 4b, 4c and 5a, 5b, 5c, we can also evaluate the performance difference between our proposed algorithm and the algorithms in [6] and [7] in the multi-user scenario. By comparing the difference between solid bars (for the non-cooperative 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 [6], [7]) become more significant in the cooperative scenario, especially when the video user percentage is small. Such a performance difference is mainly due to the non-optimal segment owner selection in [6], [7]. In our algorithm, however, the segment owner selection and the bitrate adaptation are optimised jointly.

8 CONCLUSION

In this work, we proposed a multi-user cooperative video streaming framework for video streaming over wireless networks. We analyzed the theoretical performance bound of the proposed cooperative streaming system, and designed the online streaming algorithm for the practical implementation. We conducted extensive experiments with real data traces, and illustrated both the cooperation gain of the cooperative streaming system and the performance gain of the proposed online streaming 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 paper developed a unified cooperative framework, for both theoretical analysis and practical implementation. There are several interesting future research directions in this area. An important one is to consider the users' strategic behaviours and the associated incentive issues in the cooperative streaming.

REFERENCES

[1] Cisco, "Cisco VNI: Global Mobile Data Traffic Forecast Update, 2016–2021," *Cisco White Paper*, 2012.
[2] S. Akhshabi, Ali C. Begen, and C. Dovroli, "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP," in *Proc. 2nd Annu. ACM Conf. Multimedia Syst.*, 2011, pp. 157–168.
[3] Adobe Systems, "HTTP Dynamic Streaming," 2017.
[4] R. P. Pantos, "HTTP Live Streaming draft-pantos-http-live-streaming-13," *Network Working Group*, 2014. [Online]. Available: http://tools.ietf.org/html/draft-pantos-http-live-streaming-13
[5] Microsoft, "Smooth Streaming," 2017.
[6] T. Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A buffer-based approach to rate adaptation: Evidence from a large video streaming service," *Proc. ACM Conf. SIGCOMM*, 2014, pp. 187–198.
[7] Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A. C. Begen, D. Oran, "Probe and adapt: Rate adaptation for http video streaming at scale," *IEEE J. Select. Areas Commun.*, vol. 32, 4, pp. 719–733, Apr. 2014.

[8] M. Tang, L. Gao, H. Pang, J. Huang, and L. Sun, "Optimizations and economics of crowdsourced mobile streaming," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 21–27, Apr. 2017.

[9] Y. Zhang, C. Li, and L. Sun, "DECOMOD: Collaborative DASH with download enhancing based on multiple mobile devices cooperation," in *Proc. 5th ACM Multimedia Syst. Conf.*, 2014, pp. 160–163.

[10] N. Golrezaei, P. Mansourifard, A. F. Molisch, and A. G. Dimakis, "Base-Station Assisted Device-to-Device Communications for High-Throughput Wireless Video Networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 7, pp. 3665–3676, Jul. 2014.

[11] L. Keller, A. Le, B. Cici, H. Seferoglu, C. Fragouli, and A. Markopoulou "MicroCast: Cooperative video streaming on smartphones," in *Proc. 10th Int. Conf. Mobile Syst. Appl. Serv.*, 2012, pp. 57–70.

[12] Y. Cao, X. Chen, T. Jiang, and J. Zhang, "SoCast: Social ties based cooperative video multicast," *Proc. IEEE INFOCOM*, 2014, pp. 415–423.

[13] W. Pu, Z. Zou, and C. W. Chen, "Video adaptation proxy for wireless dynamic adaptive streaming over HTTP," in *Proc. IEEE Workshop Packet Video*, 2012, pp. 65–70.

[14] M. Tang, S. Wang, L. Gao, J. Huang, and L. Sun, "MOMD: A multi-object multi-dimensional auction for crowdsourced mobile video streaming," in *Proc. IEEE INFOCOM*, 2017, pp. 1–9.

[15] M. Tang, L. Gao, H. Pang, J. Huang, and L. Sun, "Multi-dimensional auction mechanism for mobile crowdsourced video streaming," in *Proc. 14th Int. Symp. Model. Optimization Mobile Ad Hoc Wireless Netw.*, 2016, pp. 1–8.

[16] L. Gao, M. Tang, H. Pang, J. Huang, and L. Sun, "Performance Bound Analysis for Crowdsourced Mobile Video Streaming," in *Proc. Ann. Conf. Information Science and Systems*, 2016, pp. 366–371.

[17] X. Kang and Y. Wu, "Incentive Mechanism Design for Heterogeneous Peer-to-Peer Networks: A Stackelberg Game Approach," *IEEE Trans. Mobile Comput.*, vol. 14, no. 5, pp. 1018–1030, May 2015.

[18] M. Klusch, P. Kapahnke, X. Cao, B. Rainer, C. Timmerer, and S. Mangold, "MyMedia: Mobile semantic peer-to-peer video search and live streaming," in *Proc. 11th Int. Conf. Mobile Ubiquitous Syst.: Comput. Netw. Serv.*, 2014, pp. 277–286.

[19] B. Rainer, C. Timmerer, P. Kapahnke, and M. Klusch, "Real-time multimedia streaming in unstructured peer-to-peer networks," *Proc. IEEE 11th Consum. Commun. Netw. Conf.*, 2014, pp. 1136–1137.

[20] G. Iosifidis, L. Gao, J. Huang, and L. Tassiulas, "Incentive mechanisms for user-provided networks," *IEEE Commun. Mag.*, vol. 52, no. 9, pp. 20–27, Sep. 2014.

[21] G. Iosifidis, L. Gao, J. Huang, and L. Tassiulas, "Enabling crowdsourced mobile internet access," in *Proc. IEEE INFOCOM*, 2014, pp. 451–459.

[22] Open Garden, (2017). [Online]. Available: http://opengarden.com/

[23] Karma, (2017). [Online]. Available: https://yourkarma.com/

[24] Tethering of AT&T, (2017). [Online]. Available: www.att.com/shop/wireless/tethering.html

[25] K. Mori, S. Hatakeyama, and H. Shigeno, "DCLA: Distributed chunk loss avoidance method for cooperative mobile live streaming," *Proc. IEEE 29th Int. Conf. Adv. Inf. Netw. Appl.*, 2015, pp. 837–843.

[26] T. Wu, W. Dou, Q. Ni, S. Yu, and G. Chen, "Mobile live video streaming optimization via crowdsourcing brokerage," *IEEE Trans. Multimedia*, vol. 19, no. 10, pp. 2267–2281, Oct. 2017.

[27] IngKee, (2017). [Online]. Available: www.ingkee.com

[28] Youtube Live, (2017). [Online]. Available: www.youtube.com/live dashboard splash

[29] Facebook Livestream, (2017). [Online]. Available: www.facebook.com/livestream

[30] G. Iosifidis, L. Gao, J. Huang, and L. Tassiulas, "A double auction mechanism for mobile data offloading markets," *IEEE/ACM Trans. Netw.*, vol. 23, no. 5, pp. 1634–1647, Oct. 2015.

[31] T. Luo, S. S. Kanhere, J. Huang, S. K. Das, and F. Wu, "Sustainable incentives for mobile crowdsensing: Auctions, lotteries, and trust and reputation systems," *IEEE Commun. Mag.*, vol. 55, no. 3, pp. 68–74, Mar. 2017.

[32] L. Gao, Y. Xu, and X. Wang, "MAP: Multi-auctioneer progressive auction for dynamic spectrum access," *IEEE Trans. Mobile Comput.*, vol. 10, no. 8, pp. 1144–1161, Aug. 2011.

[33] C. Jiang, L. Gao, L. Duan, and J. Huang, "Data-centric mobile crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 17, no. 6, pp. 1275–1288, Jun. 2018.

[34] Q. Ma, L. Gao, Y.F. Liu, and J. Huang, "Incentivizing wi-fi network crowdsourcing: A contract theoretic approach," *IEEE/ACM Trans. Netw.*, to be published, doi: 10.1109/TNET.2018.2812785.

[35] L. Gao, J. Huang, Y. Chen, and B. Shou, "An integrated contract and auction design for secondary spectrum trading," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 3, pp. 581–592, Mar. 2013.

[36] L. Duan, L. Gao, and J. Huang, "Cooperative spectrum sharing: A contract-based approach," *IEEE Trans. Mobile Comput.*, vol. 13, no. 1, pp. 174–187, Jan. 2014.

[37] L. Gao, X. Wang, Y. Xu, and Q. Zhang, "Spectrum trading in cognitive radio networks: A contract-theoretic modeling approach," *IEEE J. Select. Areas Commun.*, vol. 29, no. 4, pp. 843–855, Apr. 2011.

[38] X. Zhang, Z. Yang, W. Sun, Y. Liu, S. Tang, K. Xing, and X. Mao, "Incentives for mobile crowd sensing: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 54–67, Jan.-Mar. 2016.

[39] L. Gao, G. Iosifidis, J. Huang, L. Tassiulas, and D. Li, "Bargaining-based Mobile Data Offloading," *IEEE J. Select. Areas Commun.*, vol. 32, no. 6, pp. 1114–1125, Jun. 2014.

[40] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Commun. Mobile Comput.*, vol. 2, no. 5, pp. 483–502, 2002.

[41] P. Yuan and H.-D. Ma, "Opportunistic Forwarding with Hotspot Entropy," in *Proc. IEEE 14th Int. Symp. Workshops World Wireless Mobile Multimedia Netw.*, 2013, 1–9.

[42] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, "Energy consumption in mobile phones: a measurement study and implications for network applications," in *Proc. 9th ACM SIGCOMM Conf. Internet Meas.*, 2009, pp. 280–293.

[43] M. A. Hoque, M. Siekkinen, and J. K. Nurminen, "Energy efficient multimedia streaming to mobile devicesa survey," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 579–597, Jan.-Mar. 2014.

[44] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems.* Morgan & Claypool, San Rafael, CA, USA, 2010.

[45] G. P. Perrucci, F. H. Fitzek, and J. Widmer, "Survey on energy consumption entities on the smartphone platform, in *Proc. IEEE 73rd Veh. Technol. Conf.*, 2011, pp. 1–6.

[46] The ilesansfil/wifidog dataset, (2015). [Online]. Available: http://crawdad.cs.dartmouth.edu/ilesansfil/wifidog/

[47] NextWiFi, (2016). [Online]. Available: http://www.nextwifi.cn/wifind/

[48] BestTV, (2016). [Online]. Available: http://www.bestv.com.cn/

[49] Technical Report at arXiv, (2018). [Online]. Available: http://arxiv.org/abs/1805.08008

Lin Gao (S'08-M'10-SM'16) received the PhD degree in electronic engineering from Shanghai Jiao Tong University, in 2010. He is an associate professor with the School of Electronic and Information Engineering, Harbin Institute of Technology, Shenzhen, China. His main research interests include network economics and games, with applications in wireless communications and networking. He received the IEEE ComSoc Asia-Pacific Outstanding Young Researcher Award in 2016. He is a senior member of the IEEE.

Ming Tang (S'16) is currently working toward the PhD degree in the Department of Information Engineering, The Chinese University of Hong Kong (CUHK). Her research interests include wireless communications and network economics, with particular emphasis on user-provided networks, mobile video streaming, and fog computing. She is a student member of the IEEE.

Haitian Pang (S'16) received the BE degree from the Department of Automation, Tsinghua University, Beijing, China, in 2014. He is currently working toward the PhD degree in computer science at Tsinghua University. His research interests include network game modeling, cellular-WiFi networking, video streaming system design, and mobile networking optimizations. He is a student member of the IEEE.

Jianwei Huang (F'16) is a professor with the Department of Information Engineering, The Chinese University of Hong Kong. He is the co-author of nine Best Paper Awards, including the IEEE Marconi Prize Paper Award in Wireless Communications 2011. He has co-authored six books, including the textbook *Wireless Network Pricing*. He has served as the chair of the *IEEE Communications Society Cognitive Networks Technical Committee* and *Multimedia Communications Technical Committee*. He is an IEEE ComSoc distinguished lecturer and a Thomson Reuters Highly cited researcher. He is a fellow of the IEEE.

Lifeng Sun received the PhD degree in system engineer from the National University of Defense Technology, Changsha, in 2000. Currently, he is a professor with Tsinghua University. His research interests include video streaming, video coding, video analysis, and multimedia cloud computing. He received the Best Paper Award in the *IEEE Transactions on Circuits and Systems for Video Technology,* in 2010, Best Paper Award at ACM Multimedia 2012, and Best Student Paper Award at MMM 2015 and IEEE BigMM 2017. He is member of the IEEE and ACM.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.