

Performance Bound Analysis for Crowdsourced Mobile Video Streaming

Lin Gao, Ming Tang, Haitian Pang, Jianwei Huang, and Lifeng Sun

Abstract—Adaptive bitrate (ABR) streaming enables video users to *adapt* the playing bitrate to the real-time network conditions to achieve the desirable quality of experience (QoE). In this work, we propose a novel *crowdsourced streaming framework* for multi-user ABR video streaming over wireless networks. This framework enables the nearby mobile video users to crowdsource their radio links and resources for cooperative video streaming. We focus on analyzing the social welfare performance *bound* of the proposed crowdsourced streaming system. Directly solving this bound is challenging due to the asynchronous operations of users. To this end, we introduce a *virtual time-slotted system* with the synchronized operations, and formulate the associated social welfare optimization problem as a linear programming. We show that the optimal social welfare performance of the virtual system provides effective upper-bound and lower-bound for the optimal performance (bound) of the original asynchronous system, hence characterizes the feasible performance *region* of the proposed crowdsourced streaming system. The performance bounds derived in this work can serve as a benchmark for the future online algorithm design and incentive mechanism design.

I. INTRODUCTION

A. Background and Motivations

Adaptive BitRate (ABR) streaming [1] 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 playing bitrate (which corresponds to the video quality such as resolution) to the real-time network conditions. To achieve the flexible bitrate adaptation, a source video is first partitioned into a sequence of short multi-second *segments*, each 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 quality of experience (QoE) and the streaming cost (e.g., energy consumption).

While most of the existing work on ABR streaming focused on the bitrate adaptation of a *single user* [2]–[9], in this work we consider a more general scenario of *multi-user* video streaming over wireless cellular networks. Note that in a multi-user scenario, the QoE of each video user is affected

This work is supported by the General Research Fund (Project Number CUHK 14202814) established under the University Grant Committee of the Hong Kong Special Administrative Region, China, and the National Natural Science Foundation of China (NSFC under No. 61472204).

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

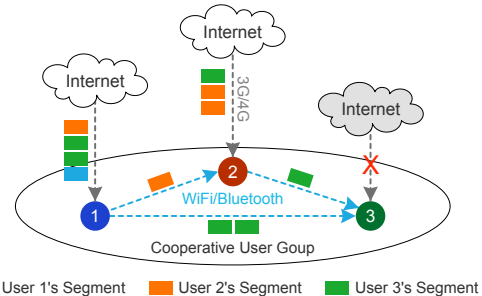


Fig. 1. Crowdsourced Video Streaming Model.

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 [10]. Without proper coordination or cooperation among users, such resource competition and multi-user 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 degrade the QoE of video users. However, traditional single-user based bitrate adaptation methods in [2]–[9] often fail to provide a desirable QoE for video users in the multi-user scenario, due to the lack of considerations of the potential cooperation among video users.

In this work, we propose a novel user cooperation framework, called *crowdsourced (video) streaming*, for multi-user video streaming over wireless cellular networks, based on the *user-provided networking* (UPN) technology [11]–[13]. 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.¹ Namely, in a cooperative group, each user can download video segments for other users using his cellular link and resources and download his own video segments through other users' links and resources. Figure 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 (who has no available cellular link), and user 2 downloads one segment for user 3.

B. Solution and Contributions

Specifically, we focus on studying the users' cooperative streaming operations (including *download scheduling* and *bitrate adaptation*) and analyzing the theoretical social welfare performance *bound* of such a crowdsourced streaming system.

¹The idea of crowdsourcing has also been applied in other applications such as wireless community networking [14] and mobile crowdsensing [15].

Namely, for each video user, when and for whom he is going to download the video segments at which bitrates, for the purpose of maximizing the social welfare performance?

We first formulate the users' cooperative streaming operations in the crowdsourced system and the associated social welfare optimization problem. The solution of such a problem can provide the theoretical social welfare performance bound of the crowdsourced system. However, directly solving this bound 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 associated 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, which leads to the feasible performance region of the proposed crowdsourced streaming system. In summary, we list the key contributions of this work 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. This framework enables mobile video users to crowdsource their radio connections and resources for cooperative video streaming, hence can increase the users' QoE.
- *Performance Bound Analysis*: We analyze the theoretical social welfare performance bound of the proposed crowdsourced system comprehensively, overcoming the challenging issue of asynchronous operations by introducing a virtual time-slotted system.
- *Practical Insights*: The performance bound analysis in this work is an important first-step towards the online algorithm design and incentive mechanism design for a crowdsourced streaming system, where the performance bounds in this work can serve as a benchmark.

The rest of the paper is organized as follows. In Section II, we present the system model. In Section III, we provide the problem formulation. In Section IV, we propose the virtual time-slotted system and the performance bound analysis. In Section V, we conclude this work and discuss the future work.

II. SYSTEM MODEL

A. Network Model

We consider a set $\mathcal{N} \triangleq \{1, 2, \dots, N\}$ of mobile video users, and each user desires to watch a video (on his smartphone) via wireless cellular network. Users move randomly in a certain area, and nearby users can form a cooperative group (via WiFi) and crowdsource their radio connections and resources for cooperative video streaming.² We refer to such a multi-user cooperative video streaming scheme as *crowdsourced (video)*

²We assume that some well-designed incentive mechanisms (e.g., Nash bargaining [17]) have been adopted such that all users are willing to participate in such a crowdsourced system to help others.

streaming. Figure 1 illustrates such a crowdsourced streaming model with a cooperative user group $\{1, 2, 3\}$.

We consider the operation in a period of continuous 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 the cellular link capacity of user n at time t , and $e_{n,m}(t) \in \{0, 1\}$ denote whether users n and m are close enough at time t such that they can connect with each other via WiFi (hence help each other). We refer to $\{(h_n(t), e_{n,m}(t)), \forall m, n \in \mathcal{N}, t \in \mathcal{T}\}$ as the *network information*, which varies randomly over time.

B. Video Streaming Model

We consider a typical ABR streaming standard [1], where a single source video file is partitioned into multiple segments and delivered to a video user using HTTP. The key features of this ABR streaming model are summarized below.

(i) *Video Segmenting*: A source video file is divided into a sequence of small 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. A user downloads the video segment by segment.

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

(iii) *Data Buffering*: Each downloaded segment is first saved in a buffer (e.g., 20–40 seconds) at the user's device, and then fetched to the video player sequentially for playback.

We denote $\beta_n > 0$ as the segment length (in seconds) of user n 's video, $Q_n > 0$ as the maximum buffer size (in seconds) of user n 's device, and $\mathcal{R}_n \triangleq \{R_n^1, \dots, R_n^Z\}$ as 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 and video player.

III. PROBLEM FORMULATION

In this section, we characterize the users' cooperative video downloading operations in the crowdsourced streaming 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.

A. Downloading Sequence

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

$$\mathcal{S}_n \triangleq \{\mathbf{s}_{n[1]}, \mathbf{s}_{n[2]}, \dots, \mathbf{s}_{n[k]}, \dots\}, \quad (1)$$

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

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

To distinguish different segments, we will also write $s_{n[k]}$ as $\{u_{n[k]}, z_{n[k]}, r_{n[k]}, t_{n[k]}^s, t_{n[k]}^e\}$ when needed.

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

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

$$C.1: t_{n[k]}^e \leq t_{n[k+1]}^s, \quad \forall k = 1, \dots, |\mathcal{S}_n|;$$

A strict inequality implies that user n waits for some time before starting to download the next segment $s_{n[k+1]}$, e.g., when the buffers of all users are full (see Section III-B).

(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 $[t_{n[k]}^s, t_{n[k]}^e]$. Hence, we have the following cellular link capacity constraint:

$$C.2: r \cdot \beta_u \leq \int_{t_{n[k]}^s}^{t_{n[k]}^e} h_n(t) dt, \quad \forall k = 1, \dots, |\mathcal{S}_n|,$$

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

(iii) *Encounter Constraint*: Each user can only download data for nearby ‘‘encountered’’ users. Hence, a segment with $s_{n[k]} = (u, z, r, [t^s, t^e])$ with $n \neq u$ is feasible only if users n and u are encountered during the interval $[t_{n[k]}^s, t_{n[k]}^e]$, i.e.,

$$C.3: e_{n,u}(t) = 1, \quad t \in [t_{n[k]}^s, t_{n[k]}^e], \quad \forall k = 1, \dots, |\mathcal{S}_n|.$$

B. Receiving Sequence

Given the feasible downloading sequences of all users, i.e., $\mathcal{S}_n, \forall n \in \mathcal{N}$, we can derive the segment receiving sequence of each user m as follows:³

$$\widehat{\mathcal{S}}_m = \bigcup_{n \in \mathcal{N}, k \in \{1, \dots, |\mathcal{S}_n|\}: u_{n[k]} = m} \{s_{n[k]}\} \quad (2)$$

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

$$\widehat{\mathcal{S}}_m \triangleq \{\hat{s}_{m[1]}, \hat{s}_{m[2]}, \dots, \hat{s}_{m[k]}, \dots\}, \quad (3)$$

with each element $\hat{s}_{m[k]} = (\hat{u}, \hat{z}, \hat{r}, [\hat{t}^s, \hat{t}^e])$ denoting the information of the k -th segment played by user m . Similarly, we will write $\hat{s}_{m[k]}$ as $\{\hat{u}_{m[k]}, \hat{z}_{m[k]}, \hat{r}_{m[k]}, \hat{t}_{m[k]}^s, \hat{t}_{m[k]}^e\}$ when needed. It is easy to see that $\hat{u}_{m[k]} = m$ for all $\hat{s}_{m[k]} \in \widehat{\mathcal{S}}_m$. To facilitate the later analysis, we further assume that $\hat{t}_{m[k]}^e \leq \hat{t}_{m[k+1]}^e, \forall k = 1, \dots, |\widehat{\mathcal{S}}_m|$, that is, user m receives the segments in $\widehat{\mathcal{S}}_m$ sequentially.⁴

As mentioned previously, each received segment is first stored in a buffer at the user’s device, and then fetched to

³We 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).

⁴Note that this can always be achieved by a proper schedule of downloading sequences with the full network information. For example, if $\hat{t}_{m[k]}^e > \hat{t}_{m[k+1]}^e$, i.e., the $k+1$ -th segment is received before the k -th segment, we can simply change their downloading orders.

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

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

where $[x]^+ = \max\{0, x\}$. Here $\hat{t}_{m[k]}^e - \hat{t}_{m[k-1]}^e$ is the time interval between receiving of $\hat{s}_{m[k-1]}$ and $\hat{s}_{m[k]}$, during which a period $\hat{t}_{m[k]}^e - \hat{t}_{m[k-1]}^e$ 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 at Q_m (seconds), we have the following *buffer constraint*:

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

C. User Payoff

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

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.

(i) *Video Quality*: A higher video quality (bitrate) brings a higher QoE for users. Let $g_n(r)$ denote the value that user n achieves from bitrate r during one unit of playback time.⁵ Then, the total value that user n achieves from all received segments $\widehat{\mathcal{S}}_n$ (each with a playback time of $\beta_{\hat{u}_{n[k]}} = \beta_n$) is:

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

Obviously, $g_n(\cdot)$ is an increasing function (as video quality monotonically increases with bitrate). As an example, we can adopt the following value function [16]: $g_n(r) = \log(1 + \theta_n \cdot r)$, 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 users’ 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 [16]. 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

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

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 $\hat{r}_{n[k-1]} - \hat{r}_{n[k]}$.

⁵Precisely speaking, this value is a function of quality. Nevertheless, under the assumption that there is a distinct and monotonic relationship between bitrate and quality, we write it as function of bitrate for notational convenience.

(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 (freeze) during playback greatly affects the users' QoE. By the buffer update rule (4), a rebuffering occurs when

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

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

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

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

$$U_n(\hat{\mathbf{S}}_n) \triangleq V_n(\hat{\mathbf{S}}_n) - L_n^{\text{QDEG}}(\hat{\mathbf{S}}_n) - L_n^{\text{REBUF}}(\hat{\mathbf{S}}_n). \quad (8)$$

2) **Energy Cost**: Users incur some energy cost in video streaming. Such energy cost mainly includes the energy consumption for data downloading on cellular links, and energy consumption for data exchange over local 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 [18]. 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 [18]:

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

(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 the local WiFi link, the energy consumption of which also depends on the transmitting time and the transmitted data volume [18]. 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 [18]:

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

where the indicator function $\mathbf{1}(u_{n[k]} \neq n) = 1$ if $u_{n[k]} \neq n$ (i.e., the segment $s_{n[k]}$ is downloaded for others), and 0 otherwise. Here we assume 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 \mathbf{S}_n and receiving sequence $\hat{\mathbf{S}}_n$ as follows:

$$C_n(\mathbf{S}_n, \hat{\mathbf{S}}_n) \triangleq E_n^{\text{CELL}}(\mathbf{S}_n) + E_n^{\text{WIFI}}(\mathbf{S}_n). \quad (9)$$

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

$$P_n(\mathbf{S}_n, \hat{\mathbf{S}}_n) \triangleq U_n(\hat{\mathbf{S}}_n) - C_n(\mathbf{S}_n, \hat{\mathbf{S}}_n) \quad (10)$$

The *social welfare* is the aggregate payoff of all users, i.e.,

$$W(\mathbf{S}_1, \dots, \mathbf{S}_N) \triangleq \sum_{n=1}^N P_n(\mathbf{S}_n, \hat{\mathbf{S}}_n), \quad (11)$$

where $\hat{\mathbf{S}}_n$ can be derived from $\mathbf{S}_n, \forall n \in \mathcal{N}$.

D. Problem Formulation

We consider an ideal scenario with *complete* network information in this work, and formulate the following *offline* social welfare maximization problem:⁶

$$\begin{aligned} \max_{\{\mathbf{S}_n, n \in \mathcal{N}\}} \quad & W(\mathbf{S}_1, \dots, \mathbf{S}_N), \\ \text{s.t.} \quad & \text{C.1} \sim \text{C.4}. \end{aligned} \quad (12)$$

The solution of (12), denoted by W^* , provides the theoretical performance bound (in term of social welfare) of the proposed crowdsourced system. However, directly solving (12) is very challenging due to the following reasons. First, users operate in an asynchronous manner. Namely, users may start to download new segments at different time. Second, (12) 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, (12) involves the integral operation (C.2), which makes it even more challenging to solve. Hence, in the next section, we will focus on finding upper-bound and lower-bound for this performance bound W^* .

IV. PERFORMANCE BOUND ANALYSIS

In this section, we propose a virtual *time-slotted download operation* scheme, under which the problem can be formulated as a linear programming, hence can be solved by many classic methods. We will show that the solution of (12) 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.**

A. 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 (e.g., 10 seconds). For convenience, we normalize the length of each slot to be one. Hence, there is a set of T time slots, denoted by $\mathcal{T} = \{1, 2, \dots, T\}$, with the τ -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

⁶Note that without complete (future) network information, we cannot formulate this offline social welfare maximization problem. In this case, we need to design online algorithms, where the downloading operation of each user is performed in an online and distributed manner.

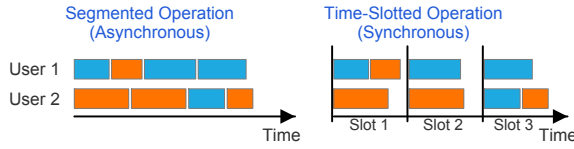


Fig. 2. Segmented vs Time-Slotted Operation.

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 between the segmented operation and the time-slotted operation in Figure 2. Blue blocks denote the user 1's data and orange blocks denote the user 2's data. Under the segmented operation scheme (left), users start to download data at different times, while under the time-slotted operation scheme (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:

$$\mathbf{K}_n \triangleq \{\kappa_{n,m}^z(\tau), \forall \tau \in \mathcal{T}, m \in \mathcal{N}, z \in \{1, \dots, Z\}\}, \quad (13)$$

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

Given the downloading vector \mathbf{K}_n , we can derive the total amount of data that user n downloads in each time slot τ :

$$x_n^{\text{DL}}(\tau) = \sum_{m=1}^N x_{n,m}(\tau) = \sum_{m=1}^N \sum_{z=1}^Z \kappa_{n,m}^z(\tau) \cdot \beta_m \cdot R_m^z, \quad (14)$$

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 \mathbf{K}_n :

$$\tilde{\text{C.2}}: x_n^{\text{DL}}(\tau) \leq H_n(\tau),$$

$$\tilde{\text{C.3}}: e_{n,m}(t) = 1, t \in [\tau - 1, \tau], \text{ if } x_{n,m}(\tau) > 0,$$

where $H_n(\tau) = \int_{\tau-1}^{\tau} h_n(t) dt$ is the aggregate cellular link capacity (in Mbps) of user n in time slot τ . Note that with the time-slotted operation, we do not need to consider the timing constraint (C.1), as the operation is already slot by slot.

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

$$y_m^{\text{RE}}(\tau) = \sum_{n=1}^N y_{n,m}(\tau) = \sum_{n=1}^N \sum_{z=1}^Z \kappa_{n,m}^z(\tau) \cdot \beta_m, \quad (15)$$

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 :⁷

$$q_m(\tau) = [q_m(\tau - 1) - 1]^+ + y_m^{\text{RE}}(\tau). \quad (16)$$

⁷Here 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:

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

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

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

$$\tilde{V}_n \triangleq \sum_{\tau=1}^T \sum_{m=1}^N \sum_{z=1}^Z \kappa_{m,n}^z(\tau) \cdot \beta_n \cdot g_n(R_n^z). \quad (17)$$

(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 the ascending order of bitrate. 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 (6), the value loss of user n induced by quality degradation is

$$\tilde{L}_n^{\text{QDEG}} \triangleq \sum_{\tau=2}^T \phi_n^{\text{QDEG}} \cdot [r_n^{\text{H}}(\tau - 1) - r_n^{\text{L}}(\tau)]^+, \quad (18)$$

(iii) **Rebuffering:** By the buffer update rule in (16), 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 (7), the value loss of user n induced by rebuffering is

$$\tilde{L}_n^{\text{REBUF}} \triangleq \sum_{\tau=2}^T \phi_n^{\text{REBUF}} \cdot [1 - q_m(\tau - 1)]^+. \quad (19)$$

(iv) **Energy Consumption for Video Downloading (via Cellular and Internet):** The energy consumption of user n for video downloading on cellular link (and Internet) is

$$\tilde{E}_n^{\text{CELL}} \triangleq \sum_{\tau=1}^T \left(c_n^{\text{TIME}} \cdot \frac{x_n^{\text{DL}}(\tau)}{H_n(\tau)} + c_n^{\text{DATA}} \cdot x_n^{\text{DL}}(\tau) \right), \quad (20)$$

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

(v) **Energy Consumption for Video Exchanging (via WiFi):** The energy consumption of user n for video exchanging on the local WiFi link is

$$\tilde{E}_n^{\text{WIFI}} \triangleq \sum_{\tau=1}^T \sum_{m=1, m \neq n}^N (w_n^{\text{TIME}} \cdot 0 + w_n^{\text{DATA}} \cdot x_{n,m}(\tau)). \quad (21)$$

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

$$\tilde{P}_n(\mathbf{K}_1, \dots, \mathbf{K}_N) \triangleq \tilde{V}_n - \tilde{L}_n^{\text{QDEG}} - \tilde{L}_n^{\text{REBUF}} - \tilde{E}_n^{\text{CELL}} - \tilde{E}_n^{\text{WIFI}}. \quad (22)$$

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

$$\max_{\{\mathbf{K}_n, n \in \mathcal{N}\}} \tilde{W} \triangleq \sum_{n=1}^N \tilde{P}_n(\mathbf{K}_1, \dots, \mathbf{K}_N), \quad (23)$$

$$\text{s.t. } \tilde{\text{C.2}} \sim \tilde{\text{C.4}}.$$

Similar to (12), this is an *offline* optimization problem and requires the complete network information. Moreover, (23) 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 (23) by \tilde{W}^* .

B. Performance Bound

Now we characterize the theoretical performance bound W^* under the segmented operation, by using the solution \widetilde{W}^* of (23) under the virtual time-slotted operation.

For convenience, we denote $\beta \triangleq (\beta_1, \dots, \beta_N)$ as the vector consisting of all users' segment lengths, and denote $W_{(\beta)}^*$ and $\widetilde{W}_{(\beta)}^*$ as the solutions of (12) and (23) 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, $\beta = (1, \dots, N)$ is an integer multiple of $\beta' = (0.5, \dots, N/2)$.

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

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

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

Proposition 2. *If $\beta \rightarrow \mathbf{0}$ (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 $\beta \rightarrow \mathbf{0}$, any downloading operation under the time-slotted operation can be equivalently achieved under the segmented operation, and vice versa.

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

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

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

Based on the above, we have the following theorem.

Theorem 1. *Given a segment length vector β , the theoretical performance bound $W_{(\beta)}^*$ is bounded by:*

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

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

V. CONCLUSION

In this work, we proposed a crowdsourced streaming framework for multi-user cooperative video streaming over mobile wireless networks, and analyzed the theoretical performance bound of the proposed crowdsourced streaming system. There are two important directions for the future extension of this

work. First, it is important to study the online scheduling algorithms for the practical implementation of the proposed crowdsourced streaming system in the scenario without complete future and global network information. Our performance bound analysis in this work can serve as a benchmark, and hence is an important first step towards the future online algorithm design for the crowdsourced streaming system. Second, incentive is a very important issue for a crowdsourced system, and is necessary for motivating video users participating the crowdsourced system to help others. Hence, it is also important to study the incentive issue in such a crowdsourced system. More specifically, in the complete information scenario, this can be achieved by a Nash bargaining between the receiver and the downloader (in each segment downloading), with which each of them can achieve a welfare no worse than that in the non-cooperative system. In the incomplete information scenario, an incentive compatible mechanism (e.g., auction) is necessary to elicit the private information of users first, and then divide the generated social welfare properly among the receiver and the downloader.

REFERENCES

- [1] S. Akhshabi, Ali C. Begen, and C. Dovrolis, "An Experimental Evaluation of Rate-Adaptation Algorithms in Adaptive Streaming over HTTP," *Proc. ACM MMSys*, 2011.
- [2] X. Yin, V. Sekar, and B. Sinopoli, "Toward a Principled Framework to Design Dynamic Adaptive Streaming Algorithms over HTTP," *Proc. ACM HotNets*, 2014.
- [3] T. Y. Huang, R. Johari, N. McKeown, *et al.*, "A buffer-based approach to rate adaptation: Evidence from a large video streaming service," *Proc. ACM SIGCOMM*, 2014.
- [4] Z. Li, X. Zhu, J. Gahn, *et al.*, "Probe and adapt: Rate adaptation for http video streaming at scale," *IEEE J. Sel. Areas in Commun.*, 2014.
- [5] J. Hao, R. Zimmermann, H. Ma, "GTube: geo-predictive video streaming over HTTP in mobile environments," *Proc. ACM MMSys*, 2014.
- [6] L. De Cicco, V. Calderalo, V. Palmisano, *et al.*, "Elastic: a client-side controller for dynamic adaptive streaming over http (dash)," *IEEE Workshop Packet Video*, 2013.
- [7] K. Miller, E. Quacchio, G. Gennari, *et al.*, "Adaptation algorithm for adaptive streaming over HTTP," *IEEE Workshop Packet Video*, 2012.
- [8] C. Zhou, C. W. Lin, X. Zhang, *et al.*, "Buffer-based smooth rate adaptation for dynamic HTTP streaming," *Proc. IEEE APSIPA*, 2013.
- [9] J. Jiang, V. Sekar, H. Zhang, "Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive," *Proc. ACM CoNEXT*, 2012.
- [10] S. Akhshabi, L. Anantakrishnan, A. C. Begen, and C. Dovrolis, "What happens when HTTP adaptive streaming players compete for bandwidth?" *ACM Workshop NOSSDAV*, 2012.
- [11] G. Iosifidis, L. Gao, J. Huang, and L. Tassiulas, "Incentive Mechanisms for User-Provided Networks," *IEEE Communications Magazine*, 2014.
- [12] G. Iosifidis, L. Gao, J. Huang, and L. Tassiulas, "Enabling Crowdsourced Mobile Internet Access," *Proc. IEEE INFOCOM*, 2014.
- [13] L. Gao, G. Iosifidis, J. Huang, *et al.*, "Hybrid Data Pricing for Network-Assisted User-Provided Connectivity," *Proc. IEEE INFOCOM*, 2014.
- [14] Q. Ma, L. Gao, Y.-F. Liu, and J. Huang, "A Game-Theoretic Analysis of User Behaviors in Crowdsourced Wireless Community Networks," *Proc. IEEE WiOpt*, 2015.
- [15] L. Gao, F. Hou, and J. Huang, "Providing Long-Term Participation Incentive in Participatory Sensing," *Proc. IEEE INFOCOM*, 2015.
- [16] V. Joseph and G. de Veciana, "NOVA: QoE-driven optimization of DASH-based video delivery in networks," *IEEE INFOCOM*, 2014.
- [17] J. Nash, "The Bargaining Problem," *Econometrica*, 1950.
- [18] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, "Energy consumption in mobile phones: a measurement study and implications for network applications," *Proc. ACM SIGCOMM*, 2009.