This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2736963, IEEE Transactions on Multimedia

View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Lancaster E-Prints

Mobile Live Video Streaming Optimization via Crowdsourcing Brokerage

Taotao Wu, *Member, IEEE,* Wanchun Dou, *Member, IEEE,* Qiang Ni, *Senior Member, IEEE,* Shui Yu, *Senior Member, IEEE,* Guihai Chen, *Senior Member, IEEE*

***Abstract*—Nowadays, people can enjoy a rich real-time sensing cognition of what they are interested in anytime and anywhere by leveraging powerful mobile devices such as smartphones. As a key support for the propagation of these richer live media contents, cellular-based access technologies play a vital role to provide reliable and ubiquitous Internet access to mobile devices. However, these limited wireless network channel conditions vary and fluctuate depending on weather, building shields, congestion, etc., which degrade the quality of live video streaming dramatically. To address this challenge, we propose to use crowdsourcing brokerage in future networks which can improve each mobile user's bandwidth condition and reduce the fluctuation of network condition. Further, to serve mobile users better in this crowdsourcing style, we study the brokerage scheduling problem which aims at maximizing the user's QoE (quality of experience) satisfaction degree cost-effectively. Both offline and online algorithms are proposed to solve this problem. The results of extensive evaluations demonstrate that by leveraging crowdsourcing technique, our solution can cost-effectively guarantee a higher quality view experience.**

***Index Terms*—Mobile Live Video Streaming, QoE, Crowdsourcing Broker, Scheduling Optimization.**

I. Introduction

With the proliferation of mobile devices such as smartphones and smart-watches, mobile live video streaming application has become more and more popular. By leveraging live video streaming, people can enjoy a real-time sensing cognition of what they are interested in. For example, in daily life, people can watch live activities of their friends or share their own activities to their friends on live video social applications such as IngKee [1]. Moreover, many other new mobile live video applications like 3D stereo video broadcast [2], mobile online cloud game [3] or ultra-high-definition (UHD) video [4] are on the rise. These new-fashioned video applications trigger higher bandwidth demand and exert more pressure on current network. Nowadays, some new network technologies such as SDN (Software Defined Networking) [5] and NFV (Network Function Virtualization) [6] are proposed to break through this dilemma by optimizing future Internet traffic. However, they usually lack the attention to the Internet access optimization for future ever-increasing mobile users, which is of vital importance for mobile live video streaming applications.

In order to provide a good QoE for mobile users in future networks, this live video paradigm will need good connection to the Internet anywhere and anytime. Normally, mobile live video users prefer a stable high-quality playback experience. Many recent researches [7-11] pointed out that a live video with poor performance such as lower bitrate and frequent freeze can annoy viewers and cause them to abandon the playback process. Further, H. Nam et al. found that even increasing bitrate can raise abandonment rates by a factor of four compared to keeping the bitrate constant [12]. These observations show that not only higher bitrate but also better bitrate switch are necessary. As such, a good network connection condition with higher bandwidth and lower volatility is preferable in mobile live video streaming correspondingly.

Nowadays, as a support of providing reliable and ubiquitous Internet access to mobile devices, cellular-based access technologies such as 3G/4G and Long Term Evolution play a vital role, since the cellular infrastructure is well-planned and widely available [13]. However, these wireless network channel conditions vary as users move and fluctuate depending on weather, building shields, congestion, etc [14]. Such random and dynamic characteristics of wireless network condition may damage both stability and fluency of live video streaming. Most existing works about online videos usually focus on individual bitrate adaptation under a specific network condition [15-17]. Actually, how to improve an individual wireless network channel condition is an important and challenging research issue on which few researches focus.

This problem tends to be more valuable for rich mobile live video streaming. Traditionally, caching techniques such as local caching are usually adopted to relieve the adverse effect of random and dynamic characteristic on on-demand video i.e., download more when network channel connectivity is good. However, it could be of little effect for mobile live video streaming in view of the unique hard real-time characteristic (*what you get is what is happening*). This unique characteristic provides little possibility for users to cache more future contents. Moreover, increasingly stronger and richer visual contents usually demand higher network channel bandwidth. Some recent reports [18][19] state that an access rate of 5.2Mbps is enough for the viewer to enjoy a

This research is supported by the EU FP7 CROWN project under grant number PIRSES-GA-2013-610524 and the National Science Foundation of China under Grant No. 61672276, the Key Research and Development Project of Jiangsu Province under Grant No. BE2015154 and BE2016120, the Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing University.

T. Wu, W. Dou and G. Chen are with the State Key Laboratory for Novel Software Technology, the Department of Computer Science and Technology, Nanjing University, China. E-mail: wutaotaoxpy@gmail.com; {douwc, gchen}@nju.edu.cn. W. Dou is the corresponding author.

Q. Ni is with the School of Computing and Communications, InfoLab21, Lancaster University, U.K. E-mail: q.ni@lancaster.ac.uk.

S. Yu is with the School of Information Technology, Deakin University, Australia. E-mail: syu@deakin.edu.au.

1520-9210 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2736963, IEEE Transactions on Multimedia

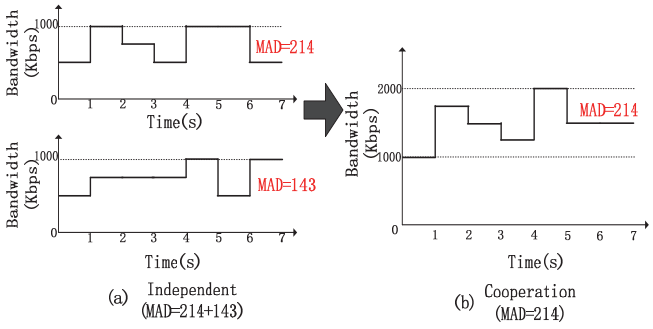


Fig. 1. Network condition optimization by cooperation between multiple users: the subfigure (a) shows the independent case without cooperation while the subfigure (b) shows the cooperative case.

live full High-Definition (FHD) video by using H.265/HEVC encoding technology while 21.4Mbps for a live 4K 2160p video. However, the individual 3G/4G throughput is usually limited, which hinders the popularization of high-quality visual contents. Therefore, in order to benefit future Internet content propagation, the improvement of individual network capacity and network volatility deserve more future research efforts.

In this paper, we address the aforementioned challenge by leveraging the crowdsourcing technique. Actually, considering individual diversity, mobile users usually enjoy different wireless network conditions because of different network operators, different locations or even different smartphones. We observe that integrating multiple users' available bandwidth can effectively improve these users' network conditions including network capacity and network volatility. Just as shown in Figure 1, two users A and B show different network condition at different times. When these two users do not cooperate with each other and only enjoy live video streaming by the single channel between local and cellular base directly, their available bandwidths are illustrated in the left subfigure (a). Here, we adopt the MAD (mean absolute deviation)[1] metric to quantitatively analyze the network volatility. User A's network has a MAD value of 214 while B is 143. Therefore, the whole system's MAD value is 214+143=357. Comparatively speaking, if there exists an intermediate node (broker) which can build connection with the cellular base and these two users simultaneously, the broker can then maintain the users' individual cellular network condition information. Based on the information, the broker can aggregate these two users' network resource and re-allocate/re-schedule network resource to the users by central control. In this case, the whole system's MAD value is only 214 (as shown in right subfigure (b)). In this paper, we dig more deeply into this phenomenon and provide theoretical results to show the advantages of user cooperation through crowdsourcing brokerage, including larger capacity provisioning and lower network volatility.

Furthermore, this crowdsourcing live video streaming paradigm can cause resource selection puzzlement because different types of resources usually have different prices. Actually, here the crowdsourcing bandwidth usually has a lower price to encourage this crowdsourcing style. In order to serve one mobile user more cost-effectively in this crowdsourcing situation, we study the brokerage scheduling problem and propose the corresponding algorithms. These algorithms focus on deciding one mobile user's quality levels (i.e., bitrate) in different locations at different times, which aims at maximizing the user's QoE satisfaction degree. One major challenge of such algorithm design lies in the mutual binding of quality levels between two adjacent time slots and total cost limitation. Moreover, due to the dynamics of crowdsourcing users, the crowdsourcing bandwidth capacity in a certain region varies and is usually unpredicted. This unpredictability makes it more challenging to design a globally optimal solution. In this paper, we progressively take these challenges into consideration and propose the corresponding offline and online algorithms.

To the best of our knowledge, our work is the first to study mobile live video streaming via crowdsourcing brokerage. Our main contributions are as follows:

(1) We design a live video streaming optimization paradigm via crowdsourcing brokerage which enables participants to share their idle bandwidths. We analyze the advantages of this paradigm and show that this crowdsourcing style can improve individual bandwidth capacity and the whole system's network stability, appealing to the live video streaming.

(2) We develop brokerage scheduling algorithms that adaptively determine the mobile user's spatial and temporal quality levels (i.e., bitrate) for both offline and online cases. The offline algorithm is a FPTAS (fully polynomial-time approximation scheme) algorithm, while the online algorithm can achieve an approximation optimal time-average utility. The corresponding gap between the approximation and the optimal one is $\mathcal{O}(1/V)$ where V is a variable input parameter. An arbitrarily large value of V can drive the approximation utility arbitrarily close to the optimal one.

(3) We perform extensive experiment evaluations using real data sets and the results demonstrate that by leveraging crowdsourcing technique, our solution can reduce the mobile user's cost and guarantee a higher quality viewing experience.

The rest of this work is organized as follows. Section II reviews the related work. Section III proposes the architecture of crowdsourcing brokerage and analyzes its advantages theoretically. Section IV presents the proposed brokerage scheduling problem and shows the corresponding algorithms for both offline and online cases. Section V evaluates our solutions and Section VI concludes this paper.

II. Related Work

Video streaming is one mainstream "killer" application over the Internet and accounts for more than half of the Internet traffic [20]. Optimizing video streaming to improve the user's QoE has been a hotspot to both industrial and academic circles in the past two decades.

Many pioneer works have been done to show that a good online video should have a high quality of experience. The traditional network QoS-based measurement is not adequate for the real-time evaluation on QoE. In view of this, by using

[1]The mean absolute deviation of a data set is the average of the absolute deviations from a central point. Here, we adopt the mean as the central point and the corresponding MAD value can be calculated by $\sum_{i=1}^{N} |x_i - \overline{x}|/N$.

1520-9210 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2736963, IEEE Transactions on Multimedia

machine learning techniques, E. Baik et al. [10] presented a visual acuity framework which can provide an accurate estimate of mobile video QoE. Coincidentally, H. Nam et al. [12] recently noted this and developed a monitoring tool named YouSlow which can detect various playback events while a video is being played. By leveraging this tool, a large number of views are collected from a testbed website YouTube. After analyzing these data, they found a surprising conclusion, i.e., even increasing bitrate can raise abandonment rates by a factor of four compared with keeping the bitrate constant. This observation is very valuable, as the existing works generally consider that a low-preforming video means starting slowly, playing at lower bitrates, and freezing frequently [7-11].

Traditionally, distributed video caching and bitrate adaptation are two common ways to improve the user's QoE. Distributed caching can effectively bring content close to the user and reduce the access latency correspondingly [21-22]. Furthermore, J. Dai et al. [37] studied the collaborative caching problem and proposed a solution based on Vickrey-Clarke-Groves (VCG) auction. This technique is very beneficial for non-live video streaming such as progressive streaming, as these videos have a lower real-time property [23]. Bitrate adaption is another means which appeals to the live or online video streaming optimization. Bitrate adaptation can adjust the video bitrate to cater to the current network condition. There has been a lot of recent works on bitrate adaption algorithms such as ELASTIC [16], PANDA [17] and BOLA [11]. These algorithms usually select the current video bitrate for individuals based on the buffer occupancy, network bandwidth and playback rate. Commonly, they all focus on bitrate adjustment and do not study how to improve the individual network condition. In view of the network variety such as Wi-Fi, WiMax and LTE, the paper [38] proposed to select the best access network for end users and correspondingly the authors designed a multi-technology simulator to validate their solutions. However, this can not improve the individual network bandwidth essentially.

Considering the scarcity of wireless resources, many researches start to focus on improving the resource coordination. Z. Guan et al. [39] presented an optimal and fair strategy for multiuser multimedia radio resource allocation based on coopetition, which is a judicious mixture of competition and cooperation. This co-opetition strategy can appeal to the changes of network conditions and provide a tradeoff between system efficiency and user fairness. The paper [40] also considered the fairness metric when optimizing video delivery to multiple users over a wireless channel. They proposed a novel cross-layer optimization framework for scalable video delivery over OFDMA wireless networks, which jointly addressed rate adaptation and resource allocation with the aim of maximizing the sum of the achievable rates while minimizing the distortion difference among multiple videos. Y. Liu et al. [41] took into account the stringent latency requirements of video flows when transmitted along inter-datacenter links shared with other types of traffic. Correspondingly, they proposed a delay-optimized traffic routing scheme to explicitly differentiate path selection for different sessions according to their delay sensitivities at the application layer.

Unlike these prior works, we consider the advantages of crowdsourcing brokerage and study brokerage scheduling problem under bitrate switch constraint and total cost limitation. A related but different work is done by A. Le et al [24]. In their proposed model MicroCast, all users in a group need to watch the same *non-live* video and share different video segments bya P2P style. Unlike [24], we study the *live* video streaming. In our proposed crowdsourcing brokerage, the idle user makes an autonomous decision on whether to provide his/her idle bandwidth to form a crowdsourcing resource pool. Then the broker would decide how to use these resources to serve the mobile viewer better.

III. Crowdsourcing Brokerage

In this section, we illustrate our crowdsourcing brokerage and explain the specific working process. Furthermore, we analyze its advantages theoretically, including volatility optimization and capacity provisioning.

A. Crowdsourcing Broker

1) Brokerage overview: The proposed crowdsourcing brokerage is just shown in Figure 2. While the anchor is in live broadcasting, the live broadcasting video is dynamically partitioned into small chunks in sequence. Each chunk is then encoded in a number of different bitrates to accommodate different network conditions and stored in the Repository. At the same time, the *viewer* can request the chunks at an appropriate bitrate based on the current total available bandwidth. Here, the current total available bandwidth consists of two parts: his/her own cellular bandwidth and the additional bandwidth from the *crowdsourcing broker*. The crowdsourcing broker maintains a virtual resource (bandwidth) pool where the bandwidth is collected from two sources: one is the *participant device group* comprising of other idle mobile users[2] and the other is the *cloudlet servers*[3] which are deployed by the network operator [25].

To realize this, the viewer's device should integrate multiple interfaces. One practical example is listed as follows: Tom's smartphone is equipped with two interfaces: cellular interface and WiFi interface. When he is enjoying live video, the smartphone build connections with the cellular base through the cellular interface (cellular link) and the crowdsourcing broker[4] through the WiFi interface (broker link) simultaneously. At backstage, Tom's live video packet transmissions are assigned onto cellular link and broker link. Furthermore, on the broker link, the crowdsourcing broker apportions the corresponding packet transmission between some other idle mobile users or cloudlet servers. Therefore, there exist multiple paths for the live video streaming from the anchor to Tom. The new standardized transport protocol Multi-Path TCP (MPTCP) [26]

[2]The broker will charge the viewer and pay these idle mobile users based on usage. The idle mobile user profitably provisions the redundant or even normal bandwidth from his data plan.

[3]Cloudlet servers are usually deployed in WLANs or WMANs for multiple paid services such as task offloading, bandwidth provisioning, etc.

[4]The broker may deploy multiple access points or leverage existing WiFi hotspots to expand its coverage by constructing large campus-sized WLANs. Moreover, WMANs can also use WiFi to provide Internet access.

1520-9210 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2736963, IEEE Transactions on Multimedia

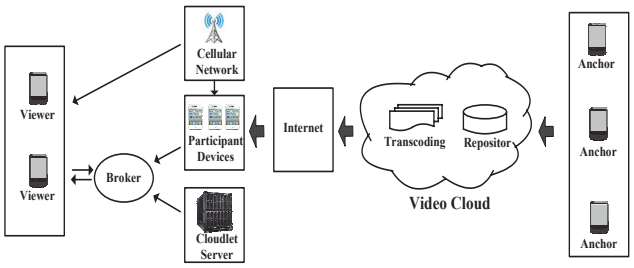


Fig. 2. Crowdsourcing brokerage: the viewers enjoy the anchors' living streaming services by leveraging cellular network or/and crowdsourcing broker.

enables and supports the simultaneous advantage of multiple network interfaces and the utilization of path diversity in the network.

2) Message/signaling exchange: In the above brokerage, there are four main roles cooperating with each other, i.e., crowdsourcing broker, viewer, participant device group and cloudlet server. We will show the specific working process and their message exchange when one viewer enjoys this crowdsourcing service.

We consider a discrete time-slotted model in this paper. Suppose the time is divided into proper time slots $t = 1, 2, \ldots$. At the beginning of each time slot, the participant device group need to evaluate/estimate their own cellular bandwidth. Then, they would report their evaluated cellular bandwidth status to the crowdsourcing broker through the WiFi connection[5]. Also, the crowdsourcing broker should inquire the bandwidth status of cloudlet server. Therefore, the crowdsourcing broker maintains the crowdsourcing cellular bandwidth status and cloudlet server bandwidth status.

When one viewer comes in, he would send his cellular bandwidth status and request information to the crowdsourcing broker. Then the cooperation between crowdsourcing broker and the viewer starts: At the beginning of each time slot, the viewer reports his own cellular bandwidth and the broker allocates the corresponding additional bandwidth; The viewer requests the chunk based on current total available bandwidth.

In this architecture, both probe-based [27] and signal-strength-based [28] approaches can be adopted to estimate the current cellular bandwidth. And different from traditional self-organized paradigm, we advocate that the mobile users do not communicate with each other and instead they build connections directly to the crowdsourcing broker. This is mainly based on the consideration of three reasons. First, the communication between mobile users especially multi-hop communication incurs longer delay due to node mobility. Fesehaye et al. [29] showed that when the maximum number of wireless hops in a group is larger than two, accessing group users incurs longer data transfer delay than directly accessing remote video cloud through 3G/4G network. Second, multi-hop communication between mobile users can cause unreliable task dissemination and retrieval. Z. Lu et al. [30] found that the delivery success probability exponentially decreases with the number of hops increasing. Finally, crowdsourcing broker can have a global view and ensure a better bandwidth allocation and scheduling plan. Actually, in our architecture, the brokerage can be owned by a third party just like the role of agent accelerator corporations. This third party can own the global information and be responsible for the coordination.

[5]Note that, in this message exchange process, the broker required status updates of participating devices every time slot. If the time slot is too short, the status updates would be too frequent. Therefore, we should reasonably set the slot length. Furthermore, in order to decrease the possible significant overhead, some estimation methods [36] can be introduced.

B. Formal Analysis

We formally analyze the advantages of the crowdsourcing brokerage on improving network condition for mobile live video streaming. Here, we take one area covered by one crowdsourcing broker into consideration. Suppose there are I participants and their available network bandwidths can be denoted as $b_i[t]$ at time t, the corresponding bandwidth demands are $u_i[t]$. Here, $u_i[t] > b_i[t]$ means the current network condition of the user i can not meet his bandwidth demand at time t and $(u_i[t] - b_i[t])^+ \triangleq \max\{u_i[t] - b_i[t], 0\}$ would be the supply and demand gap.

In the following analysis, we compare two cases, i.e., with crowdsourcing and without crowdsourcing and show the difference between them.

1) Volatility Optimization: The crowdsourcing broker can decrease the whole volatility of network condition.

Lemma 1. *(Volatility optimization)*

$$\sum_{i=1}^{I} \frac{\|\boldsymbol{b}_i - \bar{\boldsymbol{b}}_i\|_p}{T} \geq \frac{1}{T} \| \sum_{i=1}^{I} \boldsymbol{b}_i - \sum_{i=1}^{I} \bar{\boldsymbol{b}}_i \|_p \tag{1}$$

where $\boldsymbol{b}_i = (b_i[1], b_i[2], \ldots, b_i[T])$, $\bar{\boldsymbol{b}}_i = \bar{b}_i \mathbf{1}$ *where* $\bar{b}_i = \frac{1}{T}\sum_{t=1}^{T} b_i[t]$ *and* $\mathbf{1}$ *is* $1 \times T$ *vector composed of ones. The symbol* $\|\cdot\|_p$ *means* p*-norm.*

Proof. This lemma can be proved by the definition of the norm.

Based on triangle inequality of the norm, we can get the following equation, i.e., $\sum_{i=1}^{I} \|\mathbf{a}_i\|_p \geq \| \sum_{i=1}^{I} \mathbf{a}_i \|_p$.

Then, let $\mathbf{a}_i = (\mathbf{b}_i - \bar{\mathbf{b}}_i)/T$, we have,

$$\sum_{i=1}^{I} \| \frac{\mathbf{b}_i - \bar{\mathbf{b}}_i}{T} \|_p \geq \| \frac{1}{T} \sum_{i=1}^{I} (\mathbf{b}_i - \bar{\mathbf{b}}_i) \|_p. \tag{2}$$

Then, applying absolute homogeneity of the norm, we have this lemma. ■

The right-hand-side of (1) stands for the whole system's time-average volatility with crowdsourcing broker and the left-hand-side means the case without crowdsourcing broker. Lemma 1 shows that participant cooperation by crowdsourcing broker can have a smaller volatility value. Moreover, we can get an extended conclusion here, i.e, integrating all participants's available network bandwidths would have the smallest volatility. This extended conclusion can be shown by the simple iteration style. We give an instance to interpret this lemma. Suppose there are three users A, B and C. Their available bandwidths in three time slots are $\mathbf{b}_A = (1, 2, 0)$, $\mathbf{b}_B = (2, 1, 1)$, $\mathbf{b}_C = (1, 0, 2)$. Their demands are $\mathbf{u}_A = (2, 1, 0)$, $\mathbf{u}_B = (1, 2, 1)$ and $\mathbf{u}_C = (1, 1, 1)$. Then, $\bar{\mathbf{b}}_A = (1, 1, 1)$, $\bar{\mathbf{b}}_B = (4/3, 4/3, 4/3)$ and $\bar{\mathbf{b}}_C = (1, 1, 1)$.

1520-9210 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2736963, IEEE Transactions on Multimedia

The supply and demand gaps are $(1,0,0),(0,1,0),(0,1,0)$. Therefore, $MAD_{\{A\}} = \frac{\|\mathbf{b}_A - \bar{\mathbf{b}}_A\|_1}{3} = \frac{\|(1,2,0)-(1,1,1)\|_1}{3} = \frac{2}{3}$, $MAD_{\{B\}} = \frac{4}{9}$, $MAD_{\{C\}} = \frac{2}{3}$, $MAD_{\{A,B\}} = \frac{8}{9}$, $MAD_{\{B,C\}} = \frac{8}{9}$, $MAD_{\{A,C\}} = 0$, $MAD_{\{A,B,C\}} = \frac{4}{9}$. We can find that $MAD_{\{A,B,C\}} < \sum_{e\in\{A,B,C\}} MAD_{\{e\}}$ as $\frac{4}{9} < \frac{16}{9}$.

2) Capacity Provisioning: The crowdsourcing broker can improve the whole system's serving capacity.

Lemma 2. *(Capacity provisioning)*

$$\sum_{i=1}^{I}\sum_{t=1}^{T}(u_i[t]-b_i[t])^+ \geq \sum_{t=1}^{T}(\sum_{i=1}^{I}u_i[t]-\sum_{i=1}^{I}b_i[t])^+ \quad (3)$$

where $x^+ = \max\{x, 0\}$.

Proof. First, it is easy to get that $\sum_{i=1}^{I}(x_i)^+ \geq (\sum_{i=1}^{I}x_i)^+$. Then, let $x_i = u_i[t] - b_i[t]$,

$$\sum_{i=1}^{I}(u_i[t]-b_i[t])^+ \geq (\sum_{i=1}^{I}u_i[t]-\sum_{i=1}^{I}b_i[t])^+ \quad (4)$$

Summing up the above inequality over time $\{1,\ldots,T\}$, we have this lemma. ■

Lemma 2 indicates that by leveraging the crowdsourcing broker, the gap of bandwidth supply and demand in the whole system can reduce. In other words, the bandwidth utilization in this system can be improved. Just observe the above example again, we can find that the demand and supply gaps of A, B, C are all 1, i.e., $CAP_{\{A\}} = CAP_{\{B\}} = CAP_{\{C\}} = 1$. Therefore, the total demand and supply gap without crowdsourcing is $\sum_{e\in\{A,B,C\}} CAP_{\{e\}} = 3$ while the gap with crowdsourcing is only $CAP_{\{A,B,C\}} = 1$. Indeed, as shown in Lemma 2, crowdsourcing can decrease the whole system's supply and demand gap.

From Lemma 2, we also find that there may exist a certain gap of bandwidth supply and demand which can not be eliminated by the participant cooperation (e.g., $CAP_{\{A,B,C\}} = 1$ in the above example). In this paper, we assume that this remaining part is filled by the cloudlet servers.

IV. Brokerage Scheduling Optimization

The theoretical analysis in Section III can demonstrate the advantages of crowdsourcing brokerage. However, it does not show the specific scheduling optimization process when a mobile viewer enjoys this service. Actually, in order to encourage the crowdsourcing behavior, the crowdsourcing bandwidths from the participant device group and cloudlet servers are priced differentially. Here, we take this incentive style into consideration and present the formulation of brokerage scheduling optimization (abbr. BSO) problem. The key issue of this problem is to design the optimal cost-effective scheduling plan for one mobile viewer under bitrate switch constraint and total cost limitation. In this formulation, by considering this multi-path streaming occasion, not only the mobile viewers can enjoy a better cost-effective service, but also the utilization of network operators' resources can be improved. To realize this, we progressively discuss the offline case with the complete knowledge about the problem condition and the online case with no future information.

A. Problem Formulation

In this subsection, we give the detailed formulation of the brokerage scheduling optimization problem. The problem model is illustrated in Figure 3. Here, we interpret some reasons why the videos are not streamed all from the cloudlet server. As cloudlet servers are usually deployed at network edge for multiple paid services such as task offloading, its capacity is relatively rarer especially for the large user scale. Moreover, in reality, crowdsourcing resources usually have a lower price then the dedicated resources from network operators. Therefore, this hybrid service style would be more attractive and cost-effective for the mobile viewers.

1) User and video model: The whole map $\mathcal{M}$ is divided into a set of grid regions, numbered as $\mathcal{J} = \{1, 2, ..., J\}$. Each grid region $\mathcal{M}_j, j \in \mathcal{J}$ is associated with one crowdsourcing broker. The total available network bandwidth of participant device group is $r_j[t]$ for each time t and the unit cost is $p_j[t]$ correspondingly. Moreover, we assume the available bandwidth capacity of cloudlet servers operated by network operator in each grid region is infinite and priced at $q_j[t]$[6]. In order to encourage crowdsourcing operation, in this paper, we assume that $p_j[t]$ is always no larger than $q_j[t]$, i.e., $p_j[t] \leq q_j[t]$. Moreover, suppose that $q_j[t] \leq q_{max}$.

The location of the viewer v at time t, i.e., the grid region v stays in is denoted by the variable $n[t]$ (or n_t). Simultaneously, the cellular connectivity condition of viewer v at time t is $b[t]$. Here, we think the cost of this part is covered by the viewer's normal data plan and without loss of generality, is assumed to be priced at 0.

We assume the live video v enjoys is delivered with $L \in N_+$ quality levels (i.e., bitrate). The bandwidth requirement of the l-th quality level of this video is denoted as $g(l)$. Here, $g(.)$ is monotonically increasing [22]. We use $l[t] \in \{1, 2, ..., L\} \triangleq \mathcal{L}$ to represent the quality level of viewer v at time t.

2) Bitrate switch constraint: Many papers [7-11] have stated that an online video playing at lower bitrates and freezing frequently will annoy the viewer. Furthermore, a recent empirical research conducted by H. Nam et al. [12] showed that even increasing bitrate can raise abandonment rates by a factor of four compared with keeping the bitrate constant. To explain and characterize this phenomenon, we think it is related to the human's visual experience. Actually, the abrupt bitrate switch can cause an evident flickering effect which can be discerned by the human eyes. Weber's law states that a small constant difference is usually too negligible to incur adverse interference to the user [31]. That is, if the decrement for a quality level change does not exceed a threshold, the viewer will notice no or little adverse interference.

Therefore, we use two parameters to formulate this phenomenon, i.e., $\alpha \in N$ represents for differential decrement bound and $\beta \in N_+$ represents for the lowest tolerable quality level. Formally, the bitrate switch constraint is denoted as:

$$l[t] \geq \beta, \quad |d[t]| \leq \alpha. \quad (5)$$

[6]Here, the value of $q_j[t]$ can be viewed as a relaxation of this assumption. If the actual available bandwidth of cloudlet servers is limited and scarce, $q_j[t]$ can be set to be a higher value.

1520-9210 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2736963, IEEE Transactions on Multimedia

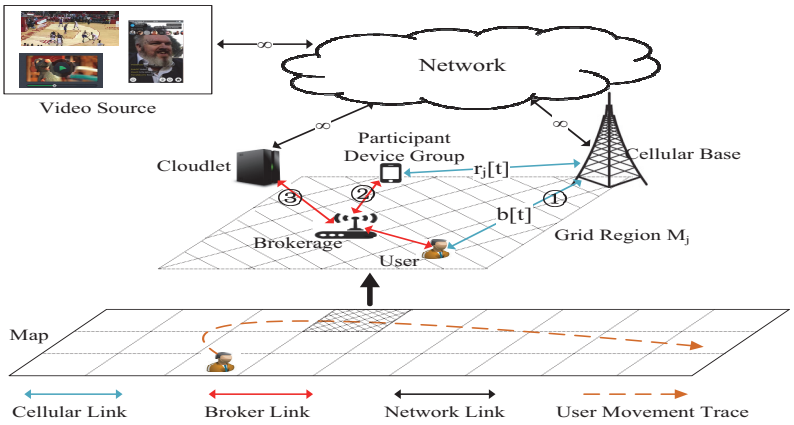


Fig. 3. The illustration of user mobility-based brokerage scheduling optimization model where four main roles cooperate with each other. The crowdsourcing broker is the core decision maker and the bandwidths from different sources ①②③ are priced differentially.

where $d[t] \triangleq l[t] - l[t-1], \forall t \in \{1, \ldots, T\}$ and $l[0] \triangleq l[1]$.

3) Total cost limitation: The gap between $g(l[t])$ and $b[t]$ is compensated by the bandwidth from crowdsourcing device group and cloudlet servers. In order to provision cost-effective services, the viewer v needs to spend crowdsourcing cost as less as possible. Therefore, considering that $q_j[t] \geq p_j[t]$, the bandwidth from the participant device group would be used in preference. Only when no bandwidth available from the participant device group, the bandwidth from the cloudlet server would be used. As such, the bandwidth from the cloudlet server (denote as $b_2[t]$) is:

$$b_2[t] = \max\{g(l[t]) - (b[t] + r_{n_t}[t]), 0\}. \tag{6}$$

Where $b[t] + r_{n_t}[t]$ is just the sum of the viewer's own cellular bandwidth and the maximum available bandwidth from the participant device group.

As $g(l[t])$ is composed of three parts of bandwidths and $b[t] + b_2[t]$ stands for the sum of the viewer's own maximum available part and cloud server part, therefore, their gap should be filled by the bandwidth from the participant device group. So the bandwidth from the participant device group (denote as $b_1[t]$) is:

$$b_1[t] = \max\{g(l[t]) - (b[t] + b_2[t]), 0\}. \tag{7}$$

Where $b_1[t] = 0$ if $g(l[t]) \leq b[t]$. This means when the viewer's own cellular bandwidth is enough, no crowdsourcing bandwidth would be allocated.

Then, the crowdsourcing cost at time t for the viewer v (denote as $c[t]$) can be calculated by:

$$c[t] = b_1[t]p_{n_t}[t] + b_2[t]q_{n_t}[t]. \tag{8}$$

Commonly, the viewer will set a budget threshold and demand that the total cost should not exceed this threshold. In this paper, we use a nonnegative time-averaged budget threshold $C \in R_0^+$ to stand for this limitation.

4) BSO problem formulation: For viewer v, the crowdsourcing broker would aim at cost-effectively optimizing video quality under bitrate switch constraint and total cost limitation. In order to evaluate video quality quantitatively, we introduce a function $w : \mathcal{L} \to [0, \infty)$. The function $w(.)$ is monotonically increasing and stands for the QoE satisfaction degree of v to the current quality level.

Definition 1 (Brokerage Scheduling Optimization Problem)**.** *The brokerage scheduling optimization problem solved by the crowdsourcing broker is formulated as follows:*

$$\begin{aligned} \max_{\boldsymbol{l}} \quad & W = \frac{1}{T}\sum_{t=1}^{T} w(l[t]), \\ \text{s.t.} \quad & (a)\ \ l[t] \geq \beta, |d[t]| \leq \alpha, \quad \forall t, \\ & (b)\ \frac{1}{T}\sum_{t=1}^{T} c[t] \leq C. \end{aligned} \tag{9}$$

where $\boldsymbol{l} = \{l[t] | t = 1, 2, \ldots, T\}$ and the parameters T, α, β and C are constant.

In the problem formulation, (a) and (b) are the bitrate switch constraint and the total cost limitation, respectively. The decision variables are quality levels at different time slots. Based on formula (6) and (7), the corresponding bandwidth allocation can be easily derived. Further, in practical use, C and T can be declared by the viewer and put in the request information while α, β and $w(.)$ can be set by the actual measurement, which is not the focus of this paper. Note that instead of putting cost factor into the goal function and finding a weighted trade-off between them, here we treat cost factor as a constraint where the mobile viewers themselves can adjust this threshold. Actually, by adjusting C value, the mobile viewers can enjoy more personalized services within individual financial plans.

B. Offline Brokerage Scheduling via DP Technique

As there are T decision variables and each decision variable can have L possible values, it is easy to see that the solution space of this problem is L^T, which requires an exponential time complexity to solve by exhaustive search. Therefore, we need a more computationally efficient solution. Here, we design an offline brokerage scheduling algorithm via dynamic programming.

1520-9210 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2736963, IEEE Transactions on Multimedia

Let k denote the current time slot in consideration.

We use $Z(k,l,m)$ to denote the optimal time-average QoE satisfaction degree in the first k time periods when ***i)*** the total cost in the first k time periods is no more than m; and ***ii)*** the current (i.e., k-th) quality level is l. Here, $k \in \{1,\ldots,T\}$, $l \in \mathcal{L}$ and $m \in R_0^+$.

Based on this definition of $Z(k,l,m)$, the Bellman equation can be written as:

$$kZ(k,l,m) = w(l) + (k-1) \max_{i,l \text{ obey } (a)} \{Z(k-1,i,m-c_k^l)\}. \tag{10}$$

where c_k^l means the crowdsourcing cost of staying at quality level l at the k-th time slot, which can be calculated by Eq. (8).

Lemma 3. $\max_{l\in\mathcal{L},l\geq\beta} Z(T,l,CT)$ *is equivalent to the original BSO problem.*

Proof. These two problems have the same constraints and goal. In $\max_{l\in\mathcal{L},l\geq\beta} Z(T,l,CT)$, the Bellman equation (10) for each $Z(T,l,CT)$ ensures $l[t] \geq \beta$, $|d[t]| \leq \alpha$ when $t \in \{1,\ldots,T-1\}$. Then, the condition $l \in \mathcal{L}, l \geq \beta$ in the final comparison process ensures the hold of $l[t] \geq \beta$, $|d[t]| \leq \alpha$ when $t = T$. Moreover, based on the definition of $Z(T,l,CT)$, the constraint (b) in BSO problem also remains and the goal is also in accord. Hence, this lemma holds. ■

Then, the original BSO problem can be transformed to find the maximum value in $\{Z(T,l,CT)|l \in \mathcal{L}, l \geq \beta\}$. We use W^* to denote this maximum value, i.e., $W^* = \max_{l\in\mathcal{L},l\geq\beta} Z(T,l,CT)$.

However, based on Eq. (10), in order to solve $\max_{l\in\mathcal{L},l\geq\beta} Z(T,l,CT)$, we need to firstly solve infinite number of sub-problems $Z(k,l,m)$ for all $m \in [0,CT]$ which is almost impossible. Actually, the key difficulty of this problem is the continuous cost interval $[0,CT]$. In view of this, we adopt a uniform discretization trick and map the continuous cost value onto discrete integer value.

The discretization interval length is set to be a positive real constant θ, also called scaledown factor in this paper. We define $\widehat{c}_k^l \triangleq \lceil \frac{c_k^l}{\theta} \rceil$, which stands for *discretized crowdsourcing cost* of staying at quality level l at the k-th time slot. Also we set the *total discretized cost bound* to be $\widehat{C}_{max} = \lceil \frac{CT-1}{\theta} \rceil$. Note that the total discretized cost bound is not set to be $\lceil \frac{CT}{\theta} \rceil$ directly and the main reason is to keep the feasibility of the solution, shown later in the proof of Theorem 1.

After these scale-down operations, the Bellman equation (10) can be adjusted to be as follows:

$$kZ(k,l,m) = w(l) + (k-1) \max_{i,l \text{ obey } (a)} \{Z(k-1,i,m-\widehat{c}_k^l)\}. \tag{11}$$

where $k \in \{1,\ldots,T\}$, $l \in \mathcal{L}$ and $m \in \{0,1,\ldots,\widehat{C}_{max}\}$.

Under this scale-down discretization technique, we turn our steps to solve a new problem $\max_{l\in\mathcal{L},l\geq\beta} Z(T,l,\widehat{C}_{max})$ and use $\widehat{W}^*$ to denote its maximum goal value i.e., $\widehat{W}^* = \max_{l\in\mathcal{L},l\geq\beta} Z(T,l,\widehat{C}_{max})$.

This discretized problem can be solved easily by dynamic programming method. We summarize our method in Algorithm 1. In Algorithm 1, we first calculate all c_k^l and discretize these real number cost (Line 1-2), then use a dynamic programming technique to solve the discretized problem (Line 3-4). Here, the subproblems $Z(k,l,m)$ are solved in sequence from $k=1$ to $k=T$. The boundary condition (Line 3) can ensure all subproblems can be iterated to solvable subproblem. Therefore, Algorithm 1 achieves $\widehat{W}^*$.

ALGORITHM 1: DP-based Offline Scheduling

Input: Scale-down factor θ;
Output: Quality level $l[t], \forall t$.
1 Calculate all $c_k^l, \forall k \in \{1,\ldots,T\}, l \in \mathcal{L}$ based on (8);
2 Set the values of $\widehat{C}_{max}$ and $\widehat{c}_k^l$: $\widehat{C}_{max} \leftarrow \lceil \frac{CT-1}{\theta} \rceil$, $\widehat{c}_k^l \leftarrow \lceil \frac{c_k^l}{\theta} \rceil, \forall k,l$;
3 Initialize the boundary condition: $Z(k,l,m) \leftarrow 0, \forall l < \beta$ or $\forall m < \widehat{c}_k^l, l \geq \beta$ and $Z(1,l,m) \leftarrow w(l), \forall m \geq \widehat{c}_1^l, l \geq \beta$;
4 Calculate all other subproblems $Z(k,l,m)$ based on (11): $Z(k,l,m) \leftarrow \frac{1}{k} w(l) + \frac{k-1}{k} \max_{(a)} \{Z(k-1,i,m-\widehat{c}_k^l)\}$;
5 Get the solution $l[t]$ corresponding to $\max_{l\in\mathcal{L},l\geq\beta} Z(T,l,\widehat{C}_{max})$;

In Algorithm 1, in order to discretize the continuous cost domain and reduce the subproblem size, we use a scaledown discretization technique. This means Algorithm 1 is just an approximate algorithm and there may exist a gap between $\widehat{W}^*$ and W^*. Next, we analyze the effect of the discretization operation on the optimality. Theorem 1 shows us its theoretical performance bound.

Theorem 1. *Algorithm 1 admits a* $(1-\epsilon)$ *approximation ratio.*

Proof. Suppose l^* is the optimal solution to the original problem and l^0 is the solution achieved by Algorithm 1.

First we prove that l^0 is a feasible solution to the original problem. Suppose that $l^0 = (s_1, s_2, \cdots, s_T)$. Therefore, we have,

$$\widehat{C}_{max} \geq \sum_{k=1}^{T} \widehat{c}_k^{s_k}, \tag{12}$$

Substituting the above equation (12) with the scale-down functions,

$$\lceil \frac{CT-1}{\theta} \rceil \geq \sum_{k=1}^{T} \lceil \frac{c_k^{s_k}}{\theta} \rceil \geq \lceil \sum_{k=1}^{T} \frac{c_k^{s_k}}{\theta} \rceil \tag{13}$$

Where the second inequality holds because $\lceil x \rceil + \lceil y \rceil \geq \lceil x + y \rceil$. From (13), considering $CT > \lceil CT-1 \rceil$, we have

$$CT > \lceil CT-1 \rceil \geq \lceil \sum_{k=1}^{T} c_k^{s_k} \rceil \tag{14}$$

Dividing T and combining $\lceil \sum_k c_k^{s_k} \rceil \geq \sum_k c_k^{s_k}$, therefore,

$$C > \frac{1}{T} \lceil \sum_k c_k^{s_k} \rceil \geq \frac{1}{T} \sum_k c_k^{s_k}. \tag{15}$$

That is, l^0 would not violate the constraint (b) in the original problem. As Eq. (11) ensures the constraint (a), we conclude that l^0 is a feasible solution to the original problem.

1520-9210 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2736963, IEEE Transactions on Multimedia

Next, we analyze the performance bound of Algorithm 1. As l^0 is a feasible solution and l^* is the optimal solution,

$$W^*(CT) \geq \widehat{W}^*(\lceil \frac{CT-1}{\theta} \rceil) \tag{16}$$

Suppose $f(\theta)$ is the least additional budget which can make Algorithm 1's solution better than l^*, i.e.,

$$\widehat{W}^*(\lceil \frac{CT-1+f(\theta)}{\theta} \rceil) = W^*(CT) \tag{17}$$

As in each time slot, the cost is over estimated by at most θ, we have $f(\theta) \leq T\theta$. Let $h_{max} \triangleq \max_{l>\beta} \frac{w(l)-w(l-1)}{g(l)-g(l-1)}$, $p_{min} \triangleq \min_{j,t} p_j[t]$, then $z = \frac{h_{max}}{p_{min}}$ stands for the maximum possible increment of QoE satisfaction degree per unit cost.

$$\begin{aligned} W^*(CT) &\leq \widehat{W}^*(\lceil \frac{CT-1+f(\theta)}{\theta} \rceil) \\ &\leq \widehat{W}^*(\lceil \frac{CT-1}{\theta} \rceil) + \frac{zf(\theta)}{T} \end{aligned} \tag{18}$$

Note that $W^*(CT) \geq w(\beta)$ as $l[t] \geq \beta$ always holds, therefore from (18) and combining $f(\theta) \leq T\theta$, we have

$$W^*(CT) \leq \widehat{W}^*(\lceil \frac{CT-1}{\theta} \rceil) + \frac{\theta z}{w(\beta)} W^*(CT). \tag{19}$$

As such,

$$\widehat{W}^*(\lceil \frac{CT-1}{\theta} \rceil) \geq (1 - \frac{\theta z}{w(\beta)}) W^*(CT). \tag{20}$$

Let $\epsilon = \frac{\theta z}{w(\beta)}$, we have this theorem. ■

Moreover, there are at most $TL\widehat{C}_{max}$ subproblems to be solved in the scale-down DP technique. And to solve each subproblem, we need to compare at most $2\alpha + 1$ iterative subproblems based on Eq. (11). Therefore, the over time complexity is bounded by $\mathcal{O}((2\alpha+1)TL\widehat{C}_{max}) = \mathcal{O}(\frac{2\alpha T^2 LCz}{w(\beta)\epsilon})$. Combining this with Theorem 1, we can conclude that Algorithm 1 is a FPTAS algorithm.

C. Online Brokerage Scheduling via Lyapunov Technique

The above DP-based brokerage scheduling method needs complete future information, which may be hard to obtain. In order to overcome this challenge, we propose an online brokerage scheduling method by leveraging Lyapunov optimization framework [32], which requires no priori knowledge about the problem condition. In the Lyapunov optimization framework, the original problem can be transformed into an optimization problem of minimizing the Lyapunov drift-plus-penalty in each time slot. By greedily solving this transformed problem, the performance of the original problem can be bounded explicitly.

In this paper, we consider the BSO problem with $T \to \infty$. With this relaxation, the control decision can be independent of the time slot index and only associated with the current network condition and occupied crowdsourcing cost.

We use a virtual queue $Q[t]$ to accumulate the additional crowdsourcing budget needed to satisfy the bitrate switch constraint in each time slot t and set $Q[0] = 0$. The update of virtual queue $Q[t]$ is given by:

$$Q[t+1] = \max\{Q[t] - C, 0\} + c[t], \tag{21}$$

ALGORITHM 2: Lyapunov-based Online Scheduling

Input: Control knob V;
Output: Quality level $l[t], \forall t$.
Initialization: $Q[0] \leftarrow 0$;
for *each time slot* $t = 1, 2, \cdots, T$ **do**
  Calculate the value $l[t] \in \mathcal{L}$ by solving (31):
  $l[t] \leftarrow \arg\min_l \; Q[t](c[t] - C) - Vw(l[t])$;
  Update the virtual queue $Q[t]$ based on (21):
  $Q[t+1] \leftarrow \max\{Q[t] - C, 0\} + c[t]$;
end

Based on the Lyapunov optimization framework, the stability of virtual queue $Q[t]$ can ensure the time average constraint. Actually, $\lim_{T\to\infty} Q[T]/T = 0$ is equivalent to require the arrival rate is no larger than the departure rate, i.e., $\lim_{T\to\infty} \sum_t c[t]/T \leq C$.

Lemma 4. *If the virtual queue Q is stable, then the time average cost constraint (b) of BSO problem can be satisfied. That is:*

$$\lim_{T\to\infty} \frac{Q[T]}{T} = 0 \Rightarrow \lim_{T\to\infty} \sum_{t=1}^{T} c[t]/T \leq C. \tag{22}$$

Proof. From Eq. (21), we have,

$$Q[t+1] \geq Q[t] - C + c[t], \tag{23}$$

Summing up the above inequality over time $\{1, \ldots, T\}$,

$$Q[T+1] - Q[1] \geq \sum_{t=1}^{T} c[t] - TC, \tag{24}$$

Dividing T and let $T \to \infty$, we have,

$$\lim_{T\to\infty} \frac{Q[T+1] - Q[1]}{T} \geq \lim_{T\to\infty} \sum_{t=1}^{T} c[t]/T - C, \tag{25}$$

Note that $Q[0] = 0$, from Eq. (21), therefore $Q[1] = c[1] \leq g(L)q_{max}$ where the second equality holds iff the viewer is at quality level L and all bandwidths are from the cloudlet server, priced q_{max}. As such, $\lim_{T\to\infty} \frac{Q[T]}{T} = 0$ would make LHS of (25) be 0. Therefore, $\lim_{T\to\infty} \sum_{t=1}^{T} c[t]/T - C \leq 0$. The lemma holds. ■

To keep the virtual queue stable, we adopt the drift-plus-penalty trick.

First, we define the Lyapunov function:

$$L[t] = \frac{1}{2}(Q[t])^2, \tag{26}$$

The one-slot Lyapunov drift in each slot t is defined as:

$$\Delta L[t] = L[t+1] - L[t], \tag{27}$$

Intuitively, by minimizing this one-slot Lyapunov drift, we can push the backlog of virtual queue Q towards a low level and keep the virtual queue stable. However, to more easily minimize the one-slot Lyapunov drift, we first calculate the upper bound of $\Delta L[t]$.

1520-9210 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2736963, IEEE Transactions on Multimedia

From Eq. (21), we have,

$$(Q[t+1])^2 \leq (Q[t])^2 + C^2 + (c[t])^2 + 2Q[t] \cdot (c[t] - C), \tag{28}$$

Applying this and Eq. (26) to Eq. (27):

$$\begin{aligned} \Delta L[t] &= \frac{1}{2}(Q[t+1])^2 - \frac{1}{2}(Q[t])^2 \\ &\leq Q[t] \cdot (c[t] - C) + B, \end{aligned} \tag{29}$$

where $B \triangleq \frac{1}{2}(C^2 + c_{max}^2)$ is a constant and $c_{max} = g(L)q_{max}$.

Considering that our original goal is to maximize the viewer's QoE satisfaction degree, we add a penalty to both sides of (29), i.e.,

$$\Delta L[t] - Vw(l[t]) \leq Q[t](c[t] - C) + B - Vw(l[t]), \tag{30}$$

Where V is a non-negative parameter which can control the tradeoff between the optimality and queue backlog.

Following the Lyapunov optimization framework, the final goal is to minimize upper-bound of the drift-plus-penalty performance, i.e., the RHS of (30). By eliminating the constant part, the original problem with $T \to \infty$ can be therefore transformed as,

$$\begin{aligned} \min_{l[t]} \quad & Q[t](c[t] - C) - Vw(l[t]), \\ \text{s.t.} \quad & l[t] \geq \beta, \ |d[t]| \leq \alpha, \quad \forall t. \end{aligned} \tag{31}$$

Now, by leveraging Lyapunov optimization framework, the original problem ($T \to \infty$) is transformed to be a much simpler problem i.e., (31). In (31), the goal function and constraint are relevant to the current state ($c[t]$, $l[t]$, $Q[t]$, $d[t]$) where $Q[t]$ and $d[t]$ are only influenced by the last one state. As such, this problem (31) can be solved in each time slot. Actually, it can be solved very easily by comparing at most $L - \beta + 1$ possible values of $l[t]$ which only needs a small computation complexity. We summarize our online algorithm in Algorithm 2 where the crowdsourcing broker iteratively solves (31) and updates the virtual queue. Specifically, Line 3 compares the goal value with all $L - \beta + 1$ possible values $\{\beta, \ldots, L\}$ of $l[t]$ at $t = 1$ and then compares the goal value with $l[t]$ from $\max\{l[t-1] - \alpha, \beta\}$ to $\min\{L, l[t-1] + \alpha\}$ at $t \geq 2$. Line 4 updates the virtual queue log which will be used in next time slot.

In Theorem 2, we prove that this online algorithm can approach the optimal solution of the original problem within an arbitrarily small gap, which can be controlled by the parameter V.

Theorem 2. *Let W^* denote the optimal time-average viewer QoE satisfaction degree value in the original BSO problem with $T \to \infty$ and $l^0[t]$ is the corresponding quality level calculated by Algorithm 2, then we have*

$$\lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} w(l^0[t]) \geq W^* - \frac{B}{V}. \tag{32}$$

Proof. The proof follows the standard Lyapunov optimization theory [32]. ■

Note that, the controllable variable V determines not only the approximation error bound $\mathcal{O}(1/V)$ but also the backlog of the virtual queue Q. Actually, a larger V would mean a larger virtual queue Q length. So in order to keep a cost-effective service for the viewer, we can not set V to be too large.

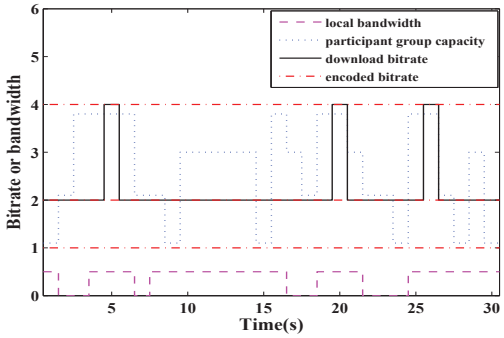


Fig. 4. Live video watching sample using the Lyapunov-based online scheduling algorithm.

D. An Example for Online Scheduling Algorithm

We illustrate a simple example to show how the online scheduling algorithm runs. In this example, Tom stays in an open deck and watches an imaginary live broadcasting video. The video would last for 30 seconds and is partitioned into 1-second chunks. All chunks are encoded at 3 different bitrates and the corresponding bitrates are 1, 2, 4Mbps. Tom's QoE satisfaction degrees on these three bitrates are 1, 1.7, 2 respectively. The bandwidth from participant device group is priced 0.02$/MB while the bandwidth from cloudlet server is priced 0.04$/MB. Tom's maximum cost budget is 0.3$, i.e., 0.01$/s and his differential decrement bound α and lowest tolerable quality β are both 1.

We use a synthetic cellular network profile profile as shown in Figure 4. Magenta dashed line shows Tom's own network profile and blue dotted line shows the crowdsourcing bandwidth capacity profile. Pick $V = 7 \times 10^{-5}$ and we choose the quality level to minimize $Q[t](c[t] - C) - Vw(l[t])$ at each time slot t. The generated algorithm result is also painted in Figure 4 (black solid line) and the corresponding queue backlog status in Q is shown in Figure 5. Here, we explain briefly the main changes in these two figures. In the beginning, Tom's own bandwidth and crowdsourcing bandwidth from participant device group are not good, therefore Q records a little more additional budget for current quality level. Then with the increasing of crowdsourcing bandwidth, the benefit per unit cost is improved, resulting in a less queue backlog (Figure 5(a)). Then, as both Tom and crowdsourcing bandwidth are good, Tom can cost-effectively achieve a higher total bandwidth, hence a higher quality level (the first peak in Figure 4). And this operation uses more future budget. Figure 5(b) shows this feedback loop and another two such feedback loops are shown in Figure 5(c) and (d).

This example also presents the QoE improvement of this live broadcasting video for Tom. Tom's own bandwidth can not support this live broadcasting video as even the lowest encoded bitrate 1Mbps surpasses his bandwidth. This broker provides Tom with a cost-effective enhancement service. By leveraging

1520-9210 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2736963, IEEE Transactions on Multimedia

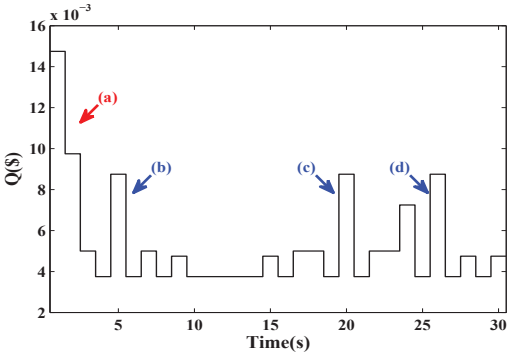


Fig. 5. The change of queue backlog status with time. Mark (a-d) show the four main changes in this live video watching sample.

the broker, Tom can watch this live broadcasting video at a moderate bitrate ($\bar{l}$=2) and there is almost no strong jitter during this view process (l approximates to be a constant).

E. Discussion

In this subsection, we discuss some practical problems when one viewer enjoys this crowdsourcing service including hand-off among different regions and input parameter setting for the above two algorithms.

1) Hand-off among different regions: In our model, the viewer is allowed to go through multiple regions. As one crowdsourcing broker is only responsible for its own coverage region, therefore the handoff operation would be carried out. To realize this, two feasible manners can be considered, i.e., self-negotiation and broker cooperation. In self-negotiation, the hand-off is carried out by the viewer himself and there is no direct communication between brokers. When the viewer enters into a new region, he will send a new request information to the new broker and start a new communication phase. As this manner demands the viewer himself to negotiate with the new broker, the viewer should enquire the status information, including current quality level and remaining budget, from the last broker and then transmit this information to the new broker. This process can degrade the video quality dramatically on account of the hard real-time characteristic of live broadcasting video. Comparatively speaking, in broker cooperation, the hand-off is carried out by the cooperation between brokers, which is similar to [33]. This manner requires that the brokers in the neighboring regions all know the viewer's current status. Therefore, when the viewer enters into a new region, the corresponding broker in this region can serve seamlessly him immediately. Usually, broker cooperation is more transparent to the viewer and appeals to the hard real-time characteristic of live broadcasting video.

2) Algorithm input parameter setting: In offline algorithm, θ is a scaledown factor, which maps infinite continuous value onto finite discrete value. Therefore, one intuitive optimal setting rule for θ is that: *different continuous values in original problem should be mapped onto different finite discrete values*. Therefore, in practical, we can first calculate all c_k^j values and sort them in an ascending order. And then we calculate the difference between any two contiguous items where the minimum is picked out, denote as δc_{min}. Finally, we can set θ to be δc_{min}. In online algorithm, V is a knob to control approximation error bound and queue backlog. We give a simple analysis on determining V. According to the queue update equality (21), we expect that C can effectively cover the crowdsourcing cost in each time slot, i.e., $Q[t] = c[t]$. Then, the goal function can be simplified as $c[t](c[t]-C)-Vw(l[t])$.

Usually the two items in this goal function should be comparable, i.e., $|c[t](c[t] - C)| \approx |Vw(l[t])|$. That is, $V \approx |c[t](c[t] - C)/w(l[t])|$. Then, we can set V to be *around* $|\bar{c}(\bar{c} - C)/w(\bar{l})|$, where $\bar{l}$ is the expected average quality level and $\bar{c}$ is the corresponding estimated crowdsourcing cost in each time slot. In the example IV(D), $\bar{l} = 2$ and $\bar{c} = 0.0043$, hence, $|\bar{c}(\bar{c} - C)/w(\bar{l})| = 1.44 \times 10^{-5}$, which has the same order of magnitudes with our picked V value (7×10^{-5}).

The above analysis provides two feasible guidelines for the input parameter settings of offline and online algorithms. In the following experiments, we completely evaluate the effects of different input parameter values on the algorithm performance.

V. Performance Evaluation

In this section, we conduct trace-driven simulations to evaluate the performance optimization of mobile live video streaming by leveraging crowdsourcing brokerage. Specifically, we first briefly introduce our simulation setup including performance metrics and comparative methods. Then, we evaluate the influence of crowdsourcing style to the system's network condition. Finally, we illustrate the evaluation results on our offline and online scheduling algorithms.

A. Simulation Setup

In our simulation, the live broadcasting video is encoded in 10 quality levels and sliced in 3-second chunks, which is consistent with the bitrates used in [11]. Also, we set one time slot length to be the live video chunk slice length (i.e., 3 seconds). While we only require the QoE satisfaction function $w(.)$ to be monotonically increasing, it is more suitable to consider increasing concave functions on account of the law of diminishing return. We take the logarithmic function as our choice here, i.e. $w(l) = \log(g(l)/g(1))$. Table I shows the bitrate for each quality level, the mean chunk video size and the corresponding QoE satisfaction value.

We use a real-world 3G mobile bandwidth trace [34] to simulate the viewer and participants' mobile network conditions. This trace contains 11 groups of logs (i.e., a set of 86 3G mobile bandwidth traces) on different routes in and around Oslo (Norway) with different transportations i.e., metro, tram, train, bus, ferry, etc. We randomly choose 10 groups of logs and treat each one as a scene. We set $p_j[t] = 1$ and $q_j[t]/p_j[t] = 5$. That is, 1KB/s bandwidth in participants' pool is priced at unit cost. The viewer's time-average maximum cost budget is considered to be moderate, i.e., $C = 100$. Moreover, the lowest tolerable quality β and differential decrement bound α are both set to be 1. That is, the live broadcasting video can be at least played back and does not change too much.

In order to do comparative analysis, we realize another three benchmark algorithms besides our proposed offline and online algorithms:

1520-9210 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2736963, IEEE Transactions on Multimedia

TABLE I
THE BITRATES USED FOR OUR LIVE VIDEO STREAMING TEST WHERE $L = 10$.

Quality Level	1	2	3	4	5	6	7	8	9	10
Bitrate(KB/s)	28	41	59	86	123	178	257	370	628	750
Chunk(Mbits)	0.69	0.99	1.43	2.06	2.97	4.28	6.17	8.89	15.08	18.00
QoE Satisfaction	0	0.3640	0.7294	1.0957	1.4606	1.8253	2.1904	2.5555	3.0845	3.2614

- *Optimal*: Brute-Force algorithm is natural to be considered as the optimal algorithm. It would search L^T possible solutions. However, its time complexity is unacceptable. We choose a replacement of Brute-Force algorithm. As in our simulation the cost is integer, therefore, we set $\theta = 1$ in our offline algorithm and treat it as the bound of Brute-Force algorithm.
- *Greedy*: We keep a temporary quality level array and iteratively pick the next most cost-efficient quality level increment. That is, let $\mathbf{l}_0 = \{l_0[t]|t = 1, \ldots, T\}$ be the temporary quality level array. $\Delta w(l_0[t]) = w(l_0[t] + 1) - w(l_0[t])$ and $\Delta c(l_0[t]) = c_t^{l_0[t]+1} - c_t^{l_0[t]}$. We search the time slot $t^* = \arg\max_t \frac{\Delta w(l_0[t])}{\Delta c(l_0[t])}$ and update $l_0[t^*] = l_0[t^*] + 1$. This process is repeated until the total cost limitation is violated.
- *Online without switch bound*: This algorithm stands for a class of existing bitrate adaptation algorithms which do not take the bitrate switch constraint into consideration. Therefore, we adjust our online algorithm by removing the bitrate switch constraint in (31).

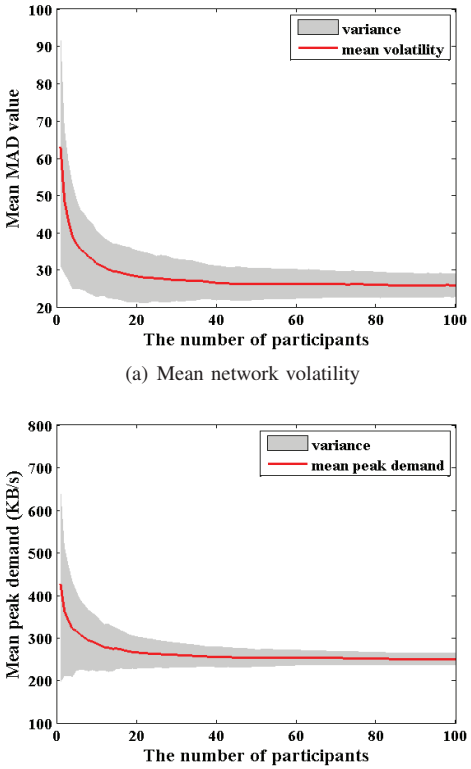
The following two main performance metrics are used in our comparative evaluations.

- *QoE satisfaction*: The time-average satisfaction degree of the viewer on the live broadcasting video, which is just our goal value W.
- *Average differential quality level*: The time-average change of quality levels in two consecutive time slots, calculated by $\sum_{t=2}^{T} |l(t) - l(t-1)|/T$.

These two metrics comprehensively reflect the QoE/QoS quality of this live broadcasting video. The first metric stands for the video's average definition while the second one shows the video's average jitter degree.

B. Crowdsourcing Network Condition Evaluation

The aforementioned Section III has proved the advantages of crowdsourcing live video streaming. We further give a detailed quantitative evaluation on the influence of scene diversity and crowdsourcing scale to the network condition optimization. Here, in order to measure the scene diversity, we introduce the well-known Shannon-Wiener index [35], which is a quantitative measure that reflects the number of different types in a dataset, and simultaneously considers the evenness degree of the basic entities distributed among those types. Here, we calculate its value by $H = -\sum_{i=1}^{10} d_i \ln d_i$ where d_i stands for the percent of participants from the i-th scene. The value of H increases both when the number of scenes increases and when evenness increases. In our evaluation, we classify $H < 1.9$ to be low diversity, $H > 2.1$ to be high diversity and choose $H = 2.05$ to be the representative value in medium diversity. The corresponding evaluation results are shown in Table II and Figure 6.



(a) Mean network volatility

(b) Mean network bandwidth peak demand

Fig. 6. The influence of crowdsourcing scale, i.e., the participants' number, on network condition including the system's stability and self-supply capacity.

In Table II, the second column is the system's average MAD value without crowdsourcing while the third column is the system's average MAD value with crowdsourcing. The third column shows their ratio between the latter and the former. It is obvious that the improvement ratio of MAD value is nearly 50 percent for all three diverse H cases. That is, the system's network stability improves with crowdsourcing. Similarly, we evaluate the system's average peak value in the fifth and sixth column. The average peak value reflects the system's self-supply capacity. A smaller peak value stands for a higher self-supply capacity. Their ratio shows that the peak demand for the whole system's bandwidth decreases about 35 percent for all three different diversities. This evaluation indicates that the crowdsourcing style can indeed dramatically improve the system's network condition. Moreover, Table II shows that

1520-9210 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2736963, IEEE Transactions on Multimedia

TABLE II
THE NETWORK CONDITION EVALUATION.

Shannon-Wiener Index (H)	Mean MAD value(before)	Mean MAD value(after)	Ratio	Mean peak demand(before)	Mean peak demand(after)	Ratio
Low(H <1.9)	62.2647	32.2218	51.75%	410.3632	279.7231	68.16%
Medium(H =2.05)	63.8326	29.4862	46.19%	415.7991	264.4499	63.60%
High(H >2.1)	63.3216	26.9790	42.61%	410.8927	252.9230	61.55%

with the increase of H, both the ratio of average MAD value and mean peak value decrease. That is, the system's network stability and network self-supply capacity turns better with the increase of system diversity. This finding has a very valuable realistic meaning: when we deploy a broker, the coverage regions with higher participants' diversity are preferred.

Figure 6 illustrates the influence of the participants' number to the system's performance. Both the two sub-figures present the similar change trend, i.e., with the increase of participants, the system's stability (MAD value) and self-supply capacity (peak value) become higher (lower). Moreover, we find that both the two red curves in subfigure (a) and (b) turns to be a constant value. That is, when the participants' number exceeds a threshold, the system's average volatility and average bandwidth peak demand tend to be a constant value. This is also an interesting observation and based on this, we can conclude another principle of broker deployment: it is unnecessary for a broker to cover too much participants. A moderate number of participants are good as more participants means more communication/scheduling operations and higher hardware requirement.

C. Offline Scheduling Performance Evaluation

In this subsection, we first compare our offline algorithm (DP) with Optimal and Greedy algorithms under four different occasions of participants' pool, i.e., when the average bandwidth capacity of participants' pool is 0 (Local), 30KB/s (Low), 170KB/s (Normal), 665KB/s (High), respectively. Then we show the effect of parameter θ on DP algorithm. The corresponding results are shown in Figure 7 and Figure 9, respectively.

1) Comparative analysis: Based on Figure 7(a), we find that with the increase of average bandwidth capacity of participants' pool, all these three methods can achieve a better QoE satisfaction. This conforms to our intuition as a higher capacity of participants' pool stands for a higher opportunity to optimize the viewer's network bandwidth. Figure 7(b) illustrates that by taking the switch bound into account, DP and Optimal algorithms can achieve a lower differential quality level value than Greedy method. Moreover, with the scale-down operation, the optimality of DP methods is influenced and there exists a satisfaction gap between the optimal solution and DP's solution. Correspondingly, the execution time consumption of DP method decreases dramatically compared to Optimal method (Figure 7(c)). Therefore, we then analyze the effect of parameter θ on the optimality and execution time consumption of DP method.

2) The effect of parameter θ: We consider the case where the average capacity of participants' pool is normal. Figure 9 shows that the performance of DP method can linearly converge to the optimum with the decrease of θ. Moreover, the corresponding execution time consumption can increase, obeying the inverse proportional relation. This phenomenon is in accordance with our theoretical analysis in Theorem 1. Therefore, θ can be indeed treated as a trade-off knob.

D. Online Scheduling Performance Evaluation

In this subsection, we first compare our online algorithm with another online algorithm without considering switch bound constraint under the four different occasions of participants' pool. We then show the effect of parameter V on our online algorithm. The corresponding results are shown in Figure 8 and Figure 10, respectively.

1) Comparative analysis: Based on Figure 8(a), with the increase of average bandwidth capacity of participants' pool, these two methods can both achieve better QoE satisfaction, in accordance with the offline evaluation. Figure 8(b) illustrates that by taking the switch bound into account, our online algorithm can achieve a lower average differential quality level value which never exceeds the previously set threshold $\alpha = 1$. Moreover, with the introduction of switch bound constraint, Lyapunov virtual queue has a smaller backlog (as shown in Figure 8(c)) and seems to be more stable.

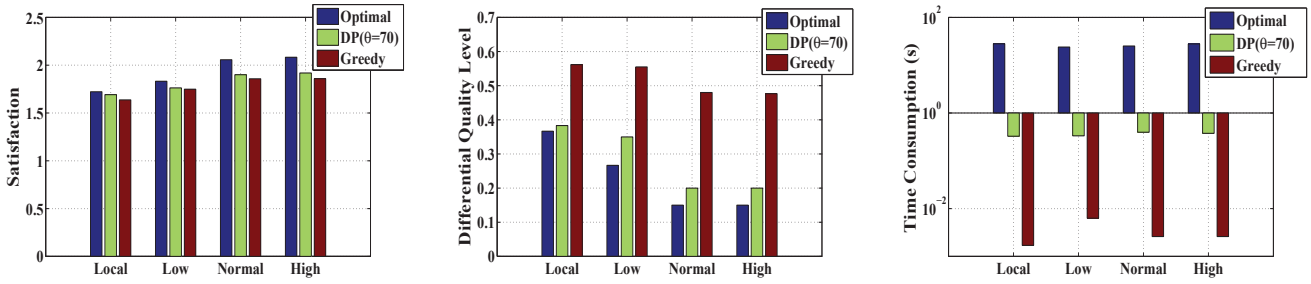
2) The effect of parameter V: As aforementioned in Section IV(C), V is a control parameter to realize a cost-effective service for the viewer. Figure 10(a) shows that a bigger V can usually result in a higher optimality, i.e., QoE satisfaction, but the relation is not necessarily monotonously increasing and may have some small fluctuation. Simultaneously, from Figure 10(b), we find that the corresponding queue backlog increases linearly with the increase of V. Considering that our online algorithm is an approximation with $T \to \infty$, therefore, to be adapted to the finite case, we can not set V too large.

VI. CONCLUSION

Cellular-based wireless network conditions show the spatio-temporal fluctuation, which influence the quality of live video streaming dramatically. To address this challenge, in this paper, we have advocated the introduction of crowdsourcing brokerage in future networks, and analyzed the advantages of this crowdsourcing brokerage on improving the mobile users' wireless network conditions. The theoretical analysis shows that this brokerage can improve both individual bandwidth capacity and the whole systems network stability, appealing to

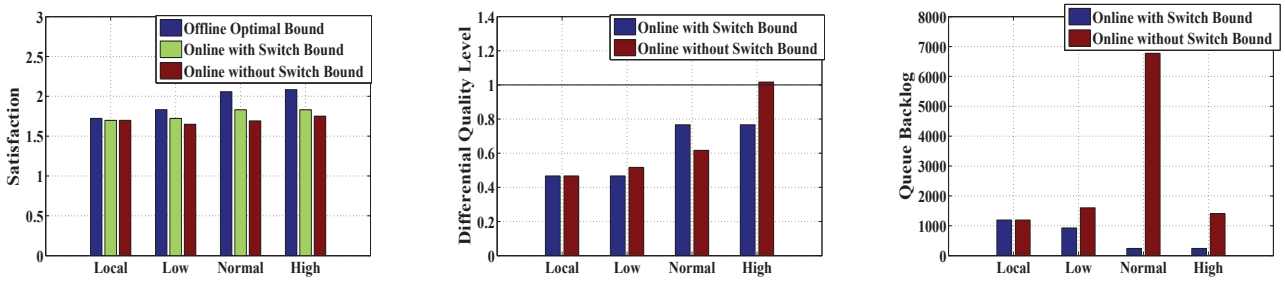
1520-9210 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2736963, IEEE Transactions on Multimedia



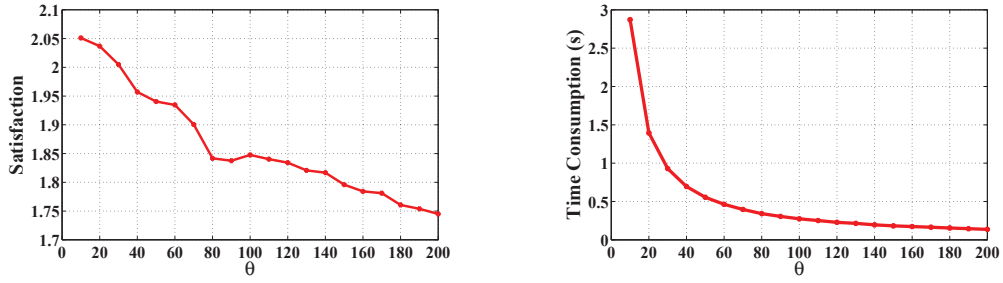
(a) QoE satisfaction (b) Average differential quality level (c) Execution time consumption

Fig. 7. The comparison between our offline algorithm (*DP*), *Optimal* and *Greedy* algorithms under four different occasions of participants' pool.



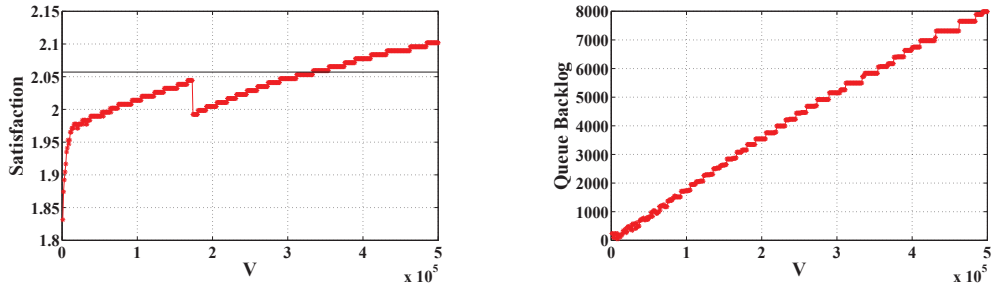
(a) QoE satisfaction (b) Average differential quality level (c) Queue backlog

Fig. 8. The comparison between our online and *online without switch bound* algorithm under four different occasions of participants' pool (V=10000).



(a) QoE satisfaction with the increase of θ (b) Time consumption with the increase of θ

Fig. 9. The effect of parameter θ on the optimality and execution time of DP method when the average crowdsourcing bandwidth capacity is normal.



(a) QoE satisfaction with the increase of V (b) Queue backlog with the increase of V

Fig. 10. The effect of parameter V on the optimality and queue backlog of online method when the average crowdsourcing bandwidth capacity is normal.

mobile live video streaming. Further, we studied the brokerage scheduling problem under this crowdsourcing brokerage style and proposed the corresponding offline (with complete future information)and online (with no future information) algorithms. The effectiveness of our algorithms has been evaluated by simulations over realistic mobile network profiles. The results demonstrate the crowdsourcing brokerage can cost-effectively guarantee a higher quality view experience.

1520-9210 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2736963, IEEE Transactions on Multimedia

REFERENCES

[1] IngKee. http://www.ingkee.com.

[2] Viasat. https://www.viasat.se.

[3] GamingCloud. http://www.gamingcloud.com.

[4] Y. Ye, P. Andrivon, "The scalable extensions of HEVC for ultra-high-definition video delivery," *IEEE MultiMedia*, Vol. 21, No. 3, 2014.

[5] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberget, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, Vol. 103, No. 1, 2015.

[6] R. Mijumbi, J. Serrat, J. Gorricho, N. Bouten, F. D. Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Communications Surveys & Tutorials*, Vol. 18, No. 1, 2016.

[7] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang, "Understanding the impact of video quality on user engagement," *ACM Special Interest Group on Data Communication (SIGCOMM)*, 2011.

[8] S. S. Krishnan and R. K. Sitaraman, "Video stream quality impacts viewer behavior: Inferring causality using quasi-experimental designs," *ACM Internet Measurement Conference (IMC)*, 2012.

[9] R. K. Sitaraman, "Network performance: Does it really matter to users and by how much?" *IEEE International Conference on Communication Systems and Networks (COMSNETS)*, 2013.

[10] E. Baik, A. Pande, C. Stover, and P. Mohapatra, "Video acuity assessment in mobile devices," *IEEE International Conference on Computer Communications (INFOCOM)*, 2015.

[11] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "BOLA: Near-optimal bitrate adaptation for online videos," *IEEE International Conference on Computer Communications (INFOCOM)*, 2016.

[12] H. Nam, K. H. Kim, and H. Schulzrinne, "QoE matters more than QoS: Why people stop watching cat videos," *IEEE International Conference on Computer Communications (INFOCOM)*, 2016.

[13] N. Wang and J. Wu, "Opportunistic WiFi offloading in a vehicular environment: waiting or downloading now?" *IEEE International Conference on Computer Communications (INFOCOM)*, 2016.

[14] F. Liu, P. Shu, and J. C.S. Lui, "AppATP: An energy conserving adaptive mobile-cloud transmission protocol," *IEEE Transactions on Computers*, Vol. 64, No. 11, 2015.

[15] E. Baik, A. Pande, Z. Zheng, and P. Mohapatra, "VSync: cloud based video streaming service for mobile devices," *IEEE International Conference on Computer Communications (INFOCOM)*, 2016.

[16] L. De Cicco, V. Caldaralo, V. Palmisano, and S. Mascolo, "ELASTIC: A client-side controller for dynamic adaptive streaming voer HTTP(DASH)," *International Packet Video Workshop (PV)*, 2013.

[17] Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A. Begen, and D. Oran, "Probe and adapt: Rate adpatation for HTTP video streaming at scale," *IEEE Journal on Selected Areas in Communications*, Vol. 32, No. 4, 2014.

[18] J. van der Hooft, S. Petrangeli, T. Wauters, R. Huysegems, P. Rondao Alface, T. Bostoen, and F. De Turck, "HTTP/2-based adaptive streaming of HEVC video over 4G/LTE networks," *IEEE Communicatioon Letters*, Vol. 20, No. 11, 2016.

[19] Big Buck Bunny Movie. https://peach.blender.org/.

[20] B. Li, Z. Wang, J. Liu, and W. Zhu, "Two decades of Internet video streaming: A retrospective view, ACM Transactions on Multimedia Computing, Communications and Applications," *Special Issue on 20th Anniversary of ACM SIGMM Multimedia*, Vol. 9, No. 1, 2013.

[21] C. Wu, X. Chen, Y. Zhou, N. Li, X. Fu, and Y. Zhang, "Spice: Socially-driven learning-based mobile media prefetching," *IEEE International Conference on Computer Communications (INFOCOM)*, 2016.

[22] K. Poularakis, G. Iosifidis, A. Argyriou, I. Koutsopoulos, and L. Tassiulas, "Caching and operator cooperation policies for layered video content delivery," *IEEE International Conference on Computer Communications (INFOCOM)*, 2016.

[23] F. Chen, C. Zhang, F. Wang, and J. Liu, "Crowdsourced live streaming over the cloud," *IEEE International Conference on Computer Communications (INFOCOM)*, 2015.

[24] A. Le, L. Keller, H. Seferoglu, B. Cici, C. Fragouli, and A. Markopoulou, "MicroCast: Cooperative video streaming using cellular and local connections," *IEEE/ACM Transactions on Networking*, to appear, 2015.

[25] Z. Xu, W. Liang, W. Xu, M. Jia, and S. Guo, "Efficient algorithms for capacitated cloudlet placements," *IEEE Transactions on Parallel and Distributed Systems*, to appear, 2015.

[26] Y. Lim, Y. Chen, E. M. Nahum, D. Towsley, R. J. Gibbens, and E. Cecchet, "Design, implementation, and evaluation of energy-aware Multi-path TCP," *ACM International Conference on Emerging Networking Experiments and Technologies (CoNEXT)*, 2015.

[27] A. Nicholson, Y. Chawathe, M. Chen, B. Noble, and D. Wetherall, "Improved access point selection," *ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2006.

[28] M. Ra, J. Paek, A. Sharma, R. Govindan, M. Krieger, and M. Neely, "Energy-delay tradeoffs in smartphone applications," *ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2010.

[29] D. Fesehaye, Y. Gao, K. Nahrstedt, and G. Wang, "Impact of cloudlets on interactive mobile cloud applications," *IEEE International Enterprise Distributed Object Computing Conference (EDOC)*, 2012.

[30] Z. Lu, X. Sun, and T. L. Porta, "Cooperative data offloading in opportunistic mobile networks," *IEEE International Conference on Computer Communications (INFOCOM)*, 2016.

[31] T. Acharya and A. K. Ray, *Image processing: Principles and applications.* John Wiley & Sons, 2005.

[32] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, Vol. 3, No. 1, 2010.

[33] W. Bao and B. Liang, "Stochastic geometric analysis of handoffs in user-centric cooperative wireless networks," *IEEE International Conference on Computer Communications (INFOCOM)*, 2016.

[34] H. Riiser, P. Vigmostad, C. Griwodz, and P. Halvorsen, "Commute path bandwidth traces from 3G networks: analysis and applications," *ACM Multimedia Systems Conference (MMsys)*, 2013.

[35] Diversity Index. http://www.wikipedia.org.

[36] Y. Zhao, H. Jiang, K. Zhou, Z. Huang, and P. Huang, "Meeting service level agreement cost-effectively for video-on-demand applications in the cloud," *IEEE Conference on Computer Communications (INFOCOM)*, 2014.

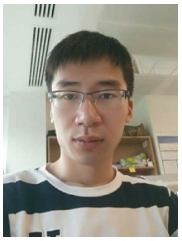
[37] J. Dai, F. Liu, B. Li, B. Li, and J. Liu, "Collaborative caching in wireless video streaming through resource auctions," *IEEE Journal on Selected Areas in Communications*, Vol. 30, No. 2, 2012.

[38] Q. Nguyen-Vuong, N. Agoulmine, E. Cherkaoui, and L. Toni, "Multicriteria optimization of access selection to improve the quality of experience in heterogeneous wireless access networks," *IEEE Transactions on Vehicular Technology*, Vol. 62, No. 4, 2013.

[39] Z. Guan, D. Yuan, and H. Zhang, "Optimal and fair resource allocation for multiuser wireless multimedia transmissions," *EURASIP Journal on Wireless Communications and Networking*, 2009.

[40] S. Cicalo and V. Tralli,"Distortion-fair cross-layer resource allocation for scalable video transmission in OFDMA wireless networks," *IEEE Transactions on Multimedia*, Vol. 16, No. 3, 2014.

[41] Y. Liu, D. Niu, and B. Li, "Delay-optimized video traffic routing in software-defined inter-datacenter networks," *IEEE Transactions on Multimedia*, Vol. 18, No. 5, 2016.



Taotao Wu received the B.S. degree in computer science from the Nanjing University of Science and Technology, Nanjing, China. He is currently working toward the Ph.D. degree in the Department of Computer Science and Technology, Nanjing University, Nanjing, China. His research interests include cloud computing and applications, multimedia computing and communications.

1520-9210 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2736963, IEEE Transactions on Multimedia

Wanchun Dou received the Ph.D. degree in mechanical and electronic engineering from the Nanjing University of Science and Technology, China, in 2001. He is currently a Full Professor of the State Key Laboratory for Novel Software Technology, Nanjing University. From April 2005 to June 2005 and from November 2008 to February 2009, he respectively visited the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, as a Visiting Scholar. Up to now, he has chaired three National Natural Science Foundation of China projects and published more than 60 research papers in international journals and international conferences. His research interests include workflow, cloud computing, and service computing.

Qiang Ni received the B.Sc., M.Sc., and Ph.D. degrees from Huazhong University of Science and Technology, China, all in engineering. He is a Professor and the Head of Communication Systems Group, School of Computing and Communications, Lancaster University, InfoLab21, Lancaster, U.K. Previously, he led the Intelligent Wireless Communication Networking Group, Brunel University London, Middlesex, U.K. His research interests include future generation communications and networking, including green communications and networking, cognitive radio network systems, heterogeneous networks, small cell and ultra dense networks, 5G, SDN, energy harvesting, wireless information and power transfer, IoTs and vehicular networks in which areas he had already published over 180 papers. He was an IEEE 802.11 Wireless Standard Working Group Voting member and a Contributor to the IEEE WIRELESS STANDARDS.

Shui Yu received his B.Eng. and M.Eng. degrees from University of Electronic Science and Technology of China, Chengdu, China, in 1993 and 1999, respectively. He received his Ph.D. degree from Deakin University, Victoria, Australia, in 2004. He is currently a Senior Lecturer with the School of Information Technology, Deakin University. He is a member of the Deakin University Academic Board (2015U2016), a member of AAAS and ACM, the Vice Chair of the Technical Subcommittee on Big Data Processing, Analytics, and Networking of the IEEE Communication Society, and a member of the IEEE Big Data Standardization Committee. His research interest includes security and privacy in networking, big data, and cyberspace, and mathematical modeling.

Guihai Chen earned his B.S. degree from Nanjing University in 1984, M.E. degree from Southeast University in 1987, and Ph.D. degree from the University of Hong Kong in 1997. He is a distinguished professor of Shanghai Jiao Tong University, China. He had been invited as a visiting professor by many universities including Kyushu Institute of Technology, Japan in 1998, University of Queensland, Australia in 2000, and Wayne State University, USA during September 2001 to August 2003. He has a wide range of research interests with focus on sensor networks, peer-to-peer computing, high-performance computer architecture and combinatorics. He has published more than 200 peer-reviewed papers, and more than 120 of them are in well-archived international journals such as IEEE Transactions on Parallel and Distributed Systems, Journal of Parallel and Distributed Computing, Wireless Networks, The Computer Journal, International Journal of Foundations of Computer Science, and Performance Evaluation, and also in well-known conference proceedings such as HPCA, MOBIHOC, INFOCOM, ICNP, ICPP, IPDPS and ICDCS.

1520-9210 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.