Adaptive Wireless Video Streaming: Joint Transcoding and Transmission Resource Allocation

Shuoyao Wang, *Member, IEEE*, Suzhi Bi, *Senior Member, IEEE*, and Ying-Jun Angela Zhang, *Fellow, IEEE*

***Abstract*—The emerging mobile edge computing (MEC) technology has been recently applied to improve adaptive bitrate (ABR) streaming service quality under time-varying wireless channels. In this paper, we consider a heterogeneous multi-user MEC-enabled video streaming network with time-varying wireless channels in sequential time frames. In particular, we aim to design an online joint transcoding and transmission resource allocation algorithm to maximize the ABR streaming user's quality of experience (QoE) subject to the bandwidth and CPU constraints. The algorithm is "online" in the sense that the bitrate and resource allocation decisions made at each frame depend only on the observation of past events. We formulate the problem as a mixed integer non-linear programming (MINLP) that jointly determines bitrate adaptation, bandwidth allocation, and CPU cycle assignment. To cope with the challenge arising from the coupling decisions of adjacent frames, we propose a low-complexity online algorithm, named OCCA. Specifically, by introducing queueing model constraints, we transform the offline non-conex MINLP problem into a multi-frame problem. Then, we analytically decouple the multi-stage(frame) problem to multiple per-frame convex subproblems that can be solved with high robustness and low computational complexity. We perform simulations with realistic scenarios to evaluate the performance of the proposed algorithm. Results manifest that compared with state-of-the-art approaches, our proposed algorithm can provide 97.84% (on average) extra QoE.**

***Index terms*—** Adaptive video streaming, Mobile edge computing, Resource allocation, Wireless networks, Online optimization

I. Introduction

Recent years have witnessed the sky-rocketing increasing network traffic. Driven by the popularity of smart user devices and communication technologies, monthly mobile data traffic is expected to exceed 396 exabytes per month by 2022, up from 122 exabytes per month in 2017. Video traffic will quadruple by 2022 and make up an even larger percentage of total traffic than before—up to 82 percent from 75 percent [1]. This trend enforces network operators to design proper resource allocation methods to assure desirable user's *quality of experience* (QoE) in increasingly congested wireless networks. The user diversity in terms of network conditions (e.g., 4G cellular, 5G cellular, or WiFi links) and QoE requirements (e.g., related to device screen sizes and user preferences) poses challenges for video service providers to assure satisfactory QoE. The situation becomes more severe for mobile users with time-varying wireless channels and highly diverse hardware/software specifications [2]. Specifically, the multiscreens like TVs, smart phones, and laptops creates a demand to adapt the video into the appropriate video specification ensuring different QoE metrics such as delay and bitrate [3].

Adaptive bitrate (ABR) video streaming [4] has been recognized as a prominent video quality adaptation technique in HTTP networks, where a video is divided into several sequential segments and each segment is encoded with different bitrates. For each segment, the video user can request the appropriate bitrate version based on such factors as screen resolution, current and predicted channel conditions. In the past decade, many research efforts [5]-[9], from both academic and industry, have been devoted to ABR streaming in wireless networks. However, conventional ABR streaming typically relies on *mobile cloud computing* (MCC) for transcoding adaptation with cloud server located far away from the user's geographic location. This incurs unacceptable latency (e.g, hundreds of milliseconds) [10] and very high volumes of data traversing in the core network [11]. Moreover, the MCC-enabled ABR streaming only provides pre-determined bitrate adaptation strategies, which only promises best effort service and has to rely on the client to adapt to the highly dynamic network conditions [12].

The recent development of *mobile edge computing* (MEC) technology provides storage and computing capabilities at servers located at the edge of radio access network (RAN), e.g., cellular base station and WiFi access point. MEC removes long backhaul latency and enjoys server load balancing performance, which is considered as an promising solution to enhance ABR streaming under time-varying wireless channel. Ref. [13] presented comprehensive surveys on the advantages of edge computing for ABR streaming, i.e., low latency, low bandwidth cost, improved QoE, and targeted services.

Compared with MCC, the MEC system is normally constrained by limited communication and computation resources. In a multi-user network, resource allocation is a paramount topic in MEC-enabled ABR. Existing approaches are broadly classified into three categories, *channel-based algorithms,*

This work is supported in part by the National Natural Science Foundation of China (Project number 62101336 and 61871271), the Tencent Rhinoceros BirdsScientific Research Foundation for Young Teachers of Shenzhen University, General Research Funding (Project number 14208017 and 14201920) from the Research Grants Council of Hong Kong, the Guangdong Province Pearl River Scholar Funding Scheme 2018, and the Key Project of Department of Education of Guangdong Province (No. 2020ZDZX3050), and National Key Research and Development Program under Project 2019YFB1803305.

S. Wang and S. Bi are with the College of Electronic and Information Engineering, Shenzhen University, Shenzhen, China (E-mail:sywang@szu.edu.cn, bsz@szu.edu.cn). S. Wang is also with the Guangdong Province Engineering Laboratory for Digital Creative Technology, Shenzhen, China, and the Shenzhen Key Laboratory of Digital Creative Technology, Shenzhen, China. S. Bi is also with the Peng Cheng Laboratory, Shenzhen, China, 518066.

Y. J. Zhang is with the Department of Information Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong (E-mail:yjzhang@ie.cuhk.edu.hk). Y. J. Zhang is also with the Shenzhen Research Institute, The Chinese University of Hong Kong, China.

Table I: Related works on adaptive bitrate video streaming.

Work	Model					Assumption		Method	
	Trans-coding	Fading Channel	Buffer Dynamics	Re-buffering Loss	Degradation Loss	Only Causal Inf.	Continuous Stochastic Channel	Computation Allocation	Bandwidth Allocation
[6]	×	×	√	√	√	√	√	×	×
[7]	×	√	×	√	×	×	-	×	√
[13]	√	×	×	×	×	×	-	×	×
[14]	×	√	×	×	×	-	-	×	√
[15]	×	√	√	×	×	√	-	×	×
[16]	×	×	√	√	√	×	-	√	×
[17]	√	√	√	√	√	×	×	√	×
[18]	√	√	√	√	×	√	×	√	×
This	√	√	√	√	√	√	√	√	√

buffer-based algorithms, and channel-buffer-based algorithms. As an example, [14] proposed a joint power allocation and sub-carrier assignment scheme for QoE maximization when the channel information from all users is available. Ref. [15] directly chooses the video rate only based on the current buffer occupancy. In practice, however, both the channel condition and playback buffer affect the QoE in ABR steaming due to the cascading effect of resource optimization. Moreover, the current channel condition directly influences the playback buffer in the next time frame, and thus the channel and buffer are highly coupled. Alternatively, based on both the playback buffer and channel condition, a model predictive control approach was proposed in [16] to make video quality adaptation decisions based on the channel prediction. Likewise, [13] formulated two subproblems, namely bitrate allocation and bandwidth adjustment, to improve the QoE assuming that the channel quality does not vary with time. In practice, wireless channel conditions are fast time varying, which may result in severely degraded performance when the prediction is inaccurate. To cope with time-varying channel conditions, [17] formulated the joint optimization problem of computational resource assignment and video bitrate adaption as a Markov decision process, and make the decision with the Asynchronous Advantage Actor-Critic (A3C) algorithm. Assuming that the distributional information of the channel conditions is not avaiable, [18] proposed a deep Q-network (DQN) based scheme to simultaneously optimize energy consumption and QoE for an MEC-enabled video streaming. Such approaches, however, assume that the channel follows a Markov process, which does not hold in most scenarios. Moreover, the existing work largely ignores the bandwidth allocation problem, and thus cannot address the fundamental challenge of bandwidth limitation [19].

To complement the previous work, we investigate the allocation of transcoding and transmission resources, including bandwidth and CPU cycle to maximize the video users' QoE in an MEC-enabled RAN when the wireless channel is stochastic. To clarify, we compare the existing ABR streaming works and our work in Table I. We aim to design an online resource allocation algorithm that jointly optimizes the transcoding and transmission resource allocation for each frame without the need to predict the future channel conditions. The online algorithm, referred to as Online joint Computation-Communication[1] resource allocation Algorithm (OCCA), is practical in the sense that we consider a multi-factor QoE and the decisions for each time frame are only based on the past events. To the authors' best knowledge, this paper is the first work that develops an online method for joint communication and computation resource allocation in MEC-enabled streaming. Our main contributions are detailed as follows.

- *Adaptation Optimization and Resource Allocation:* Our primary contribution is a principled approach for jointly optimizing the communication and computation resource allocation for QoE maximization in MEC-enabled streaming simultaneously. In particular, we consider a QoE metric consisting of three factors, including the quality gain of the video, the degradation loss of the quality, and the re-buffering time loss. We formulate the problem as a non-convex mixed integer non-linear programming (MINLP) problem to maximize the long-term QoE of all the users, subject to the sequential transmission and transcoding constraints as well as the resource constraints.
- *Queueing Model Constraints:* To allocate resource in real time, we introduce four queues to characterize the sequential transmission and transcoding constraints, namely the transmission task queue, communication capacity queue, transcoding task queue, and computation capacity queue. Through rigorous analysis, we establish the equivalence between the offline non-convex MINLP and the multi-frame problem based on the queueing models.
- *Low-complexity Online Algorithm:* To deal with the time-varying downloading capacity caused by random wireless channel, we reformulate the multi-frame joint resource allocation problem into sequential per-frame subproblems. Based on the per-frame analysis, we propose a low-complexity optimal online algorithm to solve the per-frame subproblems frame-by-frame without knowing the future channel conditions. Compared to conventional MINLP solver, OCCA takes advantage of convex optimization, and thus ensures the real-time accessibility and optimality. The computational time is less than 0.06 seconds in our experiments.
- *Performance Evaluation:* We carry out extensive evaluations to verify and analyze the performance of OCCA

[1]In this paper, we use computation-communication and transcoding-transmission alternatively.

Notations	
$\mathcal{R}$	The set of the bitrates (Mbps)
$r_{i,j} \in \mathcal{R}$	The bitrate of the jth-segment of user i (Mbps)
$\mathcal{I}$	The set of the users
$i \in \mathcal{I}$	The index of a user
$\mathcal{J}_i$	The set of segment of user i
$j \in \mathcal{J}_i$	The index of the segments
$\hat{\tau}^s_{i,j}$	The starting time of transcoding the j-th segment of user i
$\hat{\tau}^f_{i,j}$	The finishing time of transcoding the j-th segment of user i
$\tau^s_{i,j}$	The starting time of transmission the j-th segment of user i
$\tau^s_{i,j}$	The finishing time of transmission the j-th segment of user i
B	The total bandwidth of the considered wireless network (MHz)
$B_{i,t}$	The bandwidth allocated to user i at time t (MHz)
$h_{i,t}$	The channel gain of the link between the RRU and user i at time t
σ	The amount of data that can be processed during one CPU cycle (bits/CPU cycle)
$o_{i,t}$	The transcoding CPU cycles allocated to user i at time t (cycle)
$o^{\max}$	The total CPU cycles per second on the edge server (cycle)
$u_{i,j}$	The buffer of user i after completely receiving the jth-segment (second)
P	The transmission power (Watt)
N_0	The power spectral density of additive white Gaussian noise (Watt/Hz)

with open source implementation. Compared with the state-of-the-art algorithms (e.g., DQN [18], A3C [17]), our results manifest that OCCA provides extra 97.84% average QoE, *which enables users to watch high-QoE videos even under highly dynamic wireless channel.*

The rest of the paper is organized as follows. We introduce the system model and QoE formulation in Section II. Through rigorous analysis in Section III, we formulate the joint resource allocation problem and introduce the queueing model constraints to simplify the formulation. In Section IV, we decouple the offline problem into per-frame subproblems and propose the OCCA algorithm. The simulation results are presented in Section V. Finally, the paper is concluded in Section VI.

II. System Model

A. Network Model

As shown in Fig. 1, we consider an MEC network over a time horizon that is divided into T time frames. The system consists of a Remote Radio Unit (RRU), a Building Base band Unite (BBU), and $|\mathcal{I}|$ mobile users, where the RRU and users have a single antenna each. Here, we denote by $\mathcal{I}$ the set of mobile users. In particular, an edge server is co-located with the BBU, which is much closer to the users than the traditional cloud server. The heterogeneous mobile users request videos through the MEC-enabled RAN (Fig. 1). The videos are processed on the edge server, encoded based on the ABR technology, and downloaded via the wireless channel between the RRU and mobile users. Here, we denote by $\mathcal{J}_i$ the set of video segments requested by user $i \in \mathcal{I}$.

To reduce the network traffic across the core network and lower the streaming latency, the service providers only cache a single source video stream (e.g., a 4K version in Fig. 1) at the edge server rather than one stream for each viewer [11]. The

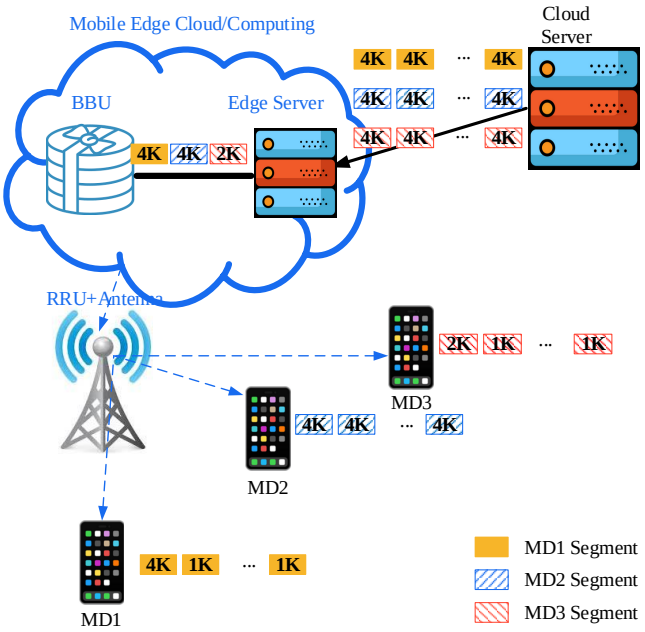


Figure 1: MEC-enabled mobile video streaming system. The bitrate is related to multiple factors, such as resolution and color clarity. We use 1K, 2K, and 4K resolutions to represent different bitrates for intuition.

edge server then transcodes[2] the video segments into multiple bitrate versions to serve the mobile users according to the their preference and real-time channel conditions.

B. Adaptive Bitrate Streaming

In particular, we consider a typical ABR streaming protocol in the mobile video streaming system. Its key features are summarized as follows.

1) Video Segmentation: A source video is partitioned into a sequence of small segments, each containing a piece of the source video with a fixed playback time β (e.g., 2 seconds).

2) Multi-Bitrate Encoding: A segment is encoded or transcoded into multiple copies with different bitrates, so that an user can select the most suitable bitrate for each segment. For notation simplicity, we denote the finite bitrate set as $\mathcal{R} = \{R_1, R_2, ..., R_{|\mathcal{R}|}\}$ (unit: bits/second). Without loss of generality, we assume that $R_1 > R_2 > ... > R_{|\mathcal{R}|}$. In this paper, the service providers cache the single source streaming with R_1 bitrate and can transcode it into $\{R_2, ..., R_{|\mathcal{R}|}\}$ based on real-time channel conditions and individual preferences if necessary.

3) Buffer Dynamics: To smoothen the playback, each downloaded segment is saved in a buffer at the user's device before playing. The video player on the user's device fetches segments from the buffer sequentially for playback.

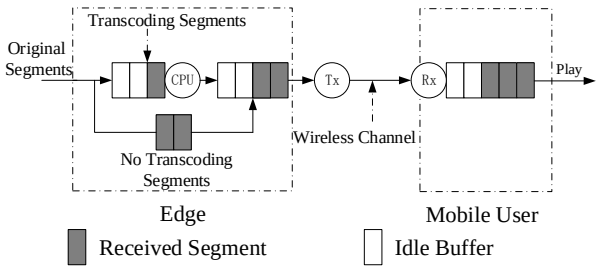


Figure 2: Transcoding and transmission model.

C. Video Transcoding

Let $r_{i,j} \in \mathcal{R}$ denote the requested bitrate of the j-th segment for the user i. As shown in Fig. 2, if the selected bitrate $r_{i,j} < R_1$, the edge server transcodes the segment from R_1 version to $r_{i,j}$ version before transmission. In this paper, we consider the allocation of the server's computation resource for executing the transcoding computational task. Suppose that the amount of data that can be processed during one CPU cycle is σ bits/CPU cycle. Let $o_{i,t}$ denote the CPU cycles allocated to user i at time t. Due to the finite computation capacity, the total CPU cycles per second on the edge server is upper bounded by $o^{\max}$. The assigned computation resources are constrained by

$$\sum_{t\in[\hat{\tau}^s_{i,j},\hat{\tau}^f_{i,j}]} o_{i,t} \geq \frac{\mu(r_{i,j})\beta}{\sigma},\ i \in \mathcal{I}, \tag{1a}$$

$$\sum_{i\in\mathcal{I}} o_{i,t} \leq o^{\max},\ t \in (0,T], \tag{1b}$$

where $\hat{\tau}^s_{i,j}$ and $\hat{\tau}^f_{i,j}$ are the starting and finishing time of the transcoding for j-th video segment of user i, respectively. Here, we denote by $\mu(r)$ the required CPU cycles per second to transcode the video segment from R_1 to r.

D. Video Transmission

Given the bitrate decision $r_{i,j}$ of the j-th segment, a total of $r_{i,j}\beta$ bits are downloaded by the user i through the wireless network. Let $\tau^s_{i,j}$ and $\tau^f_{i,j}$ denote the starting and finishing time of the downloading, respectively. On one hand, the transcoding of the segment should be completely finished before the downloading, i.e.,

$$\hat{\tau}^f_{i,j} \leq \tau^s_{i,j}. \tag{2}$$

On the other hand, the user downloads the video segments in an sequential order: the jth-segment start downloading immediately after the $(j-1)$th-segment has been completely downloaded, i.e.,

$$\tau^s_{i,j} = \begin{cases} \tau^f_{i,j-1} + 1, & \text{for } j > 1, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

In this paper, we suppose that the users access the downlink spectrum through FDMA to avoid mutual interferences with other users. Let $h_{i,t}$[3] and $B_{i,t}$ denote the channel gain of the link between the RRU and user i, and the bandwidth allocated to user i at time t, respectively. The Signal-to-Noise Ratio (SNR) from the RRU to the user i at time t is

$$\text{SNR}_{i,t} = \frac{Ph_{i,t}}{N_0 B_{i,t}}, \tag{4}$$

where P and N_0 denote the transmission power and the power spectral density of additive white Gaussian noise, respectively. According to the Shannon theorem, the downlink channel capacity of user i during time t is

$$c_{i,t} = B_{i,t} \times \log_2\left(1 + \text{SNR}_{i,t}\right). \tag{5}$$

It follows that the bandwidth allocation are constrained by

$$r_{i,j}\beta \leq \sum_{t=\tau_{i,j}}^{\hat{\tau}_{i,j}} c_{i,t},\ i \in \mathcal{I}, j \in \mathcal{J}_i, \tag{6a}$$

$$\sum_{i\in\mathcal{I}} B_{i,t} \leq B,\ t \in (0,T], \tag{6b}$$

where B denotes the total bandwidth of the considered wireless network.

E. User Quality of Experience

Due to the limited computation resource and time-varying wireless channel, the bitrates and downloading time vary across time, which appreciably influences the mobile user QoE. Similar to many existing works [6], we assume that the QoE $Q\left(\boldsymbol{r}, \boldsymbol{\tau}^s, \boldsymbol{\tau}^f, \boldsymbol{u}\right)$ consist of three terms, namely the quality gain of the video (segment) $G(\boldsymbol{r})$, the degradation loss of the quality $L_1(\boldsymbol{r})$, and the re-buffering time loss $L_2(\boldsymbol{\tau}^s, \boldsymbol{\tau}^f, \boldsymbol{u})$. Specifically,

$$Q(\boldsymbol{r}, \boldsymbol{\tau}^s, \boldsymbol{\tau}^f, \boldsymbol{u}) = G(\boldsymbol{r}) - L_1(\boldsymbol{r}) - L_2(\boldsymbol{\tau}^s, \boldsymbol{\tau}^f, \boldsymbol{u}), \tag{7}$$

where $\boldsymbol{u} = (u_{i,j}, i \in \mathcal{I}, j \in \mathcal{J}_i)$, and $u_{i,j}$ denotes the buffer occupancy of user i when the jth-segment is completely received. Intuitively, the user's viewing gain of each video segment increases with its bitrate. The user's overall viewing gain $G(\boldsymbol{r})$ is the summation of all the segment gains:

$$G(\boldsymbol{r}) = \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}_i} g_i(r_{i,j}), \tag{8}$$

where $g_i(\cdot)$ is a nondecreasing concave function [20], [21].

Remark 1. *The viewing gain functions $g(\cdot)$ of different users can differ from each other greatly. For instance, the smart phone users achieve higher viewing gain than the others when the bitrate is low. Conversely, TV users achieve higher viewing gain when the bitrate is relatively high. This is mainly because that the screen size and resolution of a smart phone is smaller than that of a TV or a laptop. In Fig. 3, we plot several examples of viewing gain functions [21].*

[2]Transcoding includes bitrate adjustment, spatial resolution reduction, temporal resolution reduction, and compression standard (codec) conversion. Since the wireless network is considered, we focus on bitrate adjustment in this paper.

[3]The channel condition transition can follow a general random process other than a Markov process.

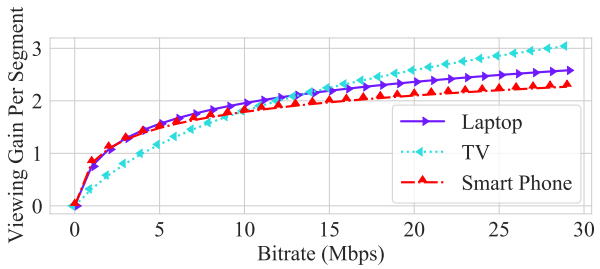


Figure 3: A few examples of viewing gain and degradation loss.

In practice, most users prefer smooth changes of video bitrates between consecutive segments. We define the degradation loss as the summation of the magnitude changes in the bitrate between consecutive segments:

$$L_1(\boldsymbol{r}) = \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}_i/\{1\}} l_1\left(r_{i,j} - r_{i,j-1}\right), \tag{9}$$

where $l_1(\cdot)$ is a nondecreasing convex function [6]. This is a generalization of those works considering linear fluctuation loss [22], [23].

For a smooth playback, each downloaded segment is saved in a buffer at the user's device before playing. Intuitively, the buffers of all the users at the beginning of the time horizon is zero, i.e., $u_{i,0} = 0, \forall i \in \mathcal{I}$. Having completely received the jth-segment, the buffer of user i is updated as:

$$u_{i,j} = [u_{i,j-1} - (\tau^f_{i,j} - \tau^s_{i,j})]^+ + \beta. \tag{10}$$

In equation (11), if the buffer occupancy $u_{i,j-1}$ is no smaller than the downloading period, then the buffer occupancy $u_{i,j}$ is the sum of the remaining buffer occupancy at time $\tau^f_{i,j}$ and the newly received buffer, i.e., β. If the buffer occupancy becomes empty, then buffer occupancy is the newly received segment. Correspondingly, if the next segment is scheduled to be played when the buffer is empty, re-buffering occurs. The re-buffering loss is the QoE loss when the users see a "blank" screen and have to wait for re-buffering. It is proportional to the re-buffering time [6]:

$$L_2(\boldsymbol{\tau}^s, \boldsymbol{\tau}^f, \boldsymbol{u}) = l_2 \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}_i} [\tau^f_{i,j} - \tau^s_{i,j} - u_{i,j-1}]^+, \tag{11}$$

where $[x]^+ = \begin{cases} x, & \text{for } x \geq 0, \\ 0, & \text{otherwise} \end{cases}$, and l_2 is the unit loss if the user suffers a one-second re-buffering.

As we can see, the use's QoE is mainly determined by two factors: bitrate and re-buffering time. Obviously, the QoE is better when the bitrates of the segments are high and stable, and the re-buffering time is small. Moreover, the QoE increases more sensitively when the bitrate is relatively low and decreases more rapidly when the re-buffering time is relatively low.

III. Problem Formulation

To jointly coordinate the transcoding and transmission with the above factors, we first determine the decision vectors $\phi_1 = \{\boldsymbol{r}, \boldsymbol{B}, \boldsymbol{o}, \hat{\boldsymbol{\tau}}^s, \hat{\boldsymbol{\tau}}^f, \boldsymbol{\tau}^s, \boldsymbol{\tau}^f\}$, the objective function, and the constraints. Without loss of generality, we denote the following decision vectors: $\boldsymbol{r} = (r_{i,j}, i \in \mathcal{I}, j \in \mathcal{J}_i)$, $\boldsymbol{\tau}^s = \left(\tau^s_{i,j}, i \in \mathcal{I}, j \in \mathcal{J}_i\right)$, $\boldsymbol{\tau}^f = \left(\tau^f_{i,j}, i \in \mathcal{I}, j \in \mathcal{J}_i\right)$, $\hat{\boldsymbol{\tau}}^s = \left(\tau^s_{i,j}, i \in \mathcal{I}, j \in \mathcal{J}_i\right)$, $\hat{\boldsymbol{\tau}}^f = \left(\tau^f_{i,j}, i \in \mathcal{I}, j \in \mathcal{J}_i\right)$, $\boldsymbol{B} = (B_{i,t}, i \in \mathcal{I}, t = 1, ..., T)$, and $\boldsymbol{o} = (o_{i,t}, i \in \mathcal{I}, t = 1, ..., T)$. In particular, the QoE maximization problem for heterogeneous users is formulated as:

$$\text{(P1)} \quad \max_{\phi_1} \quad Q\left(\boldsymbol{r}, \boldsymbol{\tau}^s, \boldsymbol{\tau}^f, \boldsymbol{u}\right), \tag{12a}$$

$$\text{s.t.} \quad r_{i,j} \in \{R_1, ..., R_m\}, \forall i, j, \tag{12b}$$

$$r_{i,j}\beta \leq \sum_{t=\tau^s_{i,j}}^{\tau^f_{i,j}} B_{i,t} \log_2\left(1 + \text{SNR}_{i,t}\right), \forall i, j, \tag{12c}$$

$$u_{i,j} = [u_{i,j-1} - (\tau^f_{i,j} - \tau^s_{i,j})]^+ + \beta, \forall i, j, \tag{12d}$$

$$\sigma \sum_{t\in[\hat{\tau}^s_{i,j}, \hat{\tau}^f_{i,j}]} o_{i,t} \geq \mu(r_{i,j})\beta, \forall i, j, \tag{12e}$$

$$\tau^s_{i,j} = \begin{cases} \tau^f_{i,j-1} + 1, & \text{for } k > 1, \\ 0, & \text{otherwise.} \end{cases} \forall i, j, \tag{12f}$$

$$\sum_{i\in\mathcal{I}} B_{i,t} \leq B, \forall t, \tag{12g}$$

$$\sum_{i\in\mathcal{I}} o_{i,t} \leq o^{\max}, \forall t, \tag{12h}$$

$$0 \leq B_{i,t}, 0 \leq \tau^s_{i,j}, \tau^f_{i,j}, \hat{\tau}^s_{i,j}, \hat{\tau}^f_{i,j} \leq T, \quad \hat{\tau}^f_{i,j} \leq \tau^s_{i,j}, \forall i, t, j. \tag{12i}$$

Problem (P1) is a non-convex MINLP problem with nonlinear and undifferentiable constraints (12b-e). (12c) denotes the sequential transmission constraint, which ensures that the segments are downloaded sequentially and the transmission data rate does not exceed the channel capacity for error-free transmissions. (12d) denotes the buffer dynamics. (12e) denotes the sequential transcoding constraint, which ensures that the segments are transcoded sequentially and the assigned CPU cycles are sufficient for transcoding. (12f and i) ensures the transmission times $(\boldsymbol{\tau}^s, \boldsymbol{\tau}^f)$ and transcoding times $(\hat{\boldsymbol{\tau}}^s, \hat{\boldsymbol{\tau}}^f)$ are in non-decreasing order. (12g) corresponds to the total bandwidth constraint. (12h) corresponds to the total CPU cycles constraint.

Besides being non-convex, the sequential transmission and transcoding constraints in (P1) make the decision variables in each frame highly coupled. Fortunately, we observe that the QoE (7) is a function of $(\boldsymbol{r}, \boldsymbol{\tau}^s, \boldsymbol{\tau}^f, \boldsymbol{u})$, and that the transcoding time decision $(\hat{\boldsymbol{\tau}}^s, \hat{\boldsymbol{\tau}}^f)$ is only related to the feasible space of $(\boldsymbol{r}, \boldsymbol{\tau}^s, \boldsymbol{\tau}^f, \boldsymbol{u})$. Therefore, given a feasible $(\boldsymbol{r}, \boldsymbol{\tau}^s, \boldsymbol{\tau}^f, \boldsymbol{u})$ decision, (12a) is independent of $(\hat{\boldsymbol{\tau}}^s, \hat{\boldsymbol{\tau}}^f)$. In Theorem 1, we prove that it suffices to optimize the transmission times $(\boldsymbol{\tau}^s, \boldsymbol{\tau}^f)$ instead of both the transcoding and transmission times $(\hat{\boldsymbol{\tau}}^s, \hat{\boldsymbol{\tau}}^f, \boldsymbol{\tau}^s, \boldsymbol{\tau}^f)$ for joint resource allocation optimization.

Theorem 1. *Problem (P1) is equivalent to the following problem:*

$$\begin{aligned}
\textit{(P2)}\quad \max_{\phi_2}\quad & Q\left(\boldsymbol{r},\boldsymbol{\tau}^s,\boldsymbol{\tau}^f,\boldsymbol{u}\right), && \text{(13a)}\\
\textit{s.t.}\quad & \sum_{j'=1,\dots,j}\mu(r_{i,j'})\beta\le\sigma\sum_{\tau=1,\dots,\tau^s_{i,j}}o_{i,\tau},\forall i,j, && \text{(13b)}\\
& 0\le B_{i,t},0\le\tau^s_{i,j},\tau^f_{i,j}\le T,\forall i,j,t, && \text{(13c)}\\
& \textit{constraints (12b-d, f-h)}, && \text{(13d)}
\end{aligned}$$

where $\phi_2=\{\boldsymbol{r},\boldsymbol{B},\boldsymbol{o},\boldsymbol{\tau}^s,\boldsymbol{\tau}^f\}$. In particular, given an optimal solution ϕ_2^* to (P2), we can always find a pair $(\hat{\boldsymbol{\tau}}^{s*},\hat{\boldsymbol{\tau}}^{f*})$ such that the decision $(\phi_2^*,\hat{\boldsymbol{\tau}}^{s*},\hat{\boldsymbol{\tau}}^{f*})$ is an optimal solution to (P1).

Proof. Please see the detailed proof in Appendix A. □

Thanks to Theorem 1, the number of decision variables in Problem (P2) is much smaller than (P1). Moreover, the transformed constraint (13b) entails that the summation of the total transcoding bits before time t must be no more than the summation of the allocated computation resource from time 0 to t. Inspired by the causality forms on both sides of (13b), we introduce an equivalent formulation in Lemma 1, which summarizes the causalities by virtual queues and transforms the problem from segment-by-segment to frame-by-frame. Here, by frame-by-frame, we mean that the constraints are imposed for each t. The frame-by-frame formulation facilitates the online algorithm design in Section V, which will decouple the decisions in each frame.

Specifically, we introduce four queues, namely transcoding task queue $q_{i,1}(t)$, computation capacity queue $q_{i,2}(t)$, transmission task queue $q_{i,3}(t)$, and communication capacity queue $q_{i,4}(t)$, for each user i to characterize the constraints (12b,d) in a frame-by-frame form:

$$\begin{aligned}
q_{i,1}(t)&=\beta\sum_{j\in\{j|\tau^s_{i,j}\le t\}}\mu(r_{i,j})\\
&=q_{i,1}(t-1)+\beta\sum_{j\in\{j|\tau^s_{i,j}=t\}}\mu(r_{i,j}), && \text{(14a)}\\
q_{i,2}(t)&=\sigma\sum_{\tau=1,\dots,t}o_{i,\tau}=q_{i,2}(t-1)+\sigma o_{i,t}, && \text{(14b)}\\
q_{i,3}(t)&=\beta\sum_{j\in\{j|\tau^f_{i,j}\le t\}}r_{i,j}\\
&=q_{i,3}(t-1)+\beta\sum_{j\in\{j|\tau^f_{i,j}=t\}}r_{i,j}, && \text{(14c)}\\
q_{i,4}(t)&=\sum_{\tau=1}^{t}B_{i,\tau}\log_2\left(1+\mathrm{SNR}_{i,\tau}\right)\\
&=q_{i,4}(t-1)+\log_2\left(1+\mathrm{SNR}_{i,t}\right). && \text{(14d)}
\end{aligned}$$

Accordingly, we prove in Lemma 1 that solving Problem (15) is sufficient to obtain an optimal solution of Problem (P1).

Lemma 1. *Problem (P2) is equivalent to the following problem:*

$$\begin{aligned}
\textit{(P3)}\quad \max_{\phi_2}\quad & Q\left(\boldsymbol{r},\boldsymbol{\tau}^s,\boldsymbol{\tau}^f,\boldsymbol{u}\right), && \text{(15a)}\\
\textit{s.t.}\quad & q_{i,1}(t)\le q_{i,2}(t),\forall t,i, && \text{(15b)}\\
& q_{i,3}(t)\le q_{i,4}(t),\forall t,i, && \text{(15c)}\\
& \textit{constraints (12b, d, f-h)(13c)}. && \text{(15d)}
\end{aligned}$$

Proof. Please see the detailed proof in Appendix B. □

Given the channel realizations throughout the considered period, we can solve Problem (15) to obtain the optimal transcoding and resource allocation control in an offline fashion. In practice, however, the fast-varying channel condition requires real-time decision-making in each time frame based on the causal channel information that is known so far. In the next section, we propose a novel online algorithm that solves (15) with both high robustness and efficiency in Section IV.

IV. Online Algorithm

In this section, we propose an online algorithm that jointly optimizes the communication and computation resource allocations to maximize the users' QoE, for real-time decision-making without the knowledge of the future channel conditions. In particular, we first decouple Problem (15) into sequential per-frame subproblems by approximating the expected future channel condition and CPU assignment with the current observations. Accordingly, we solve the per-frame deterministic subproblems sequentially with a robust and low-complexity online algorithm.

A. Multi-frame Problem

Problem (15) shows that, at the beginning of each time t, the provider only needs to determine the decisions performed at time t. Let $\mathcal{I}_t=\{i|\exists j\in\mathcal{J}_i,\tau^f_{i,j}=t-1\}$ and $j_{i,t}$ denote the set of users whose segments start to transcode at t, and the index such that $\tau^f_{i,j_{i,t}-1}=t-1$, respectively. Then, the decisions at time t can be expressed as the bitrate selections $\boldsymbol{r}^t=\begin{cases}(r_{i,j_{i,t}},i\in\mathcal{I}_t),\text{for }|\mathcal{I}_t|>0,\\0,\text{otherwise}\end{cases}$, the bandwidth allocation $\boldsymbol{b}^t=(B_{i,t},i\in\mathcal{I})$, and the CPU allocation $\boldsymbol{o}^t=(o_{i,t},i\in\mathcal{I})$. According to (15), the decision $(\boldsymbol{r}^t,\boldsymbol{b}^t,\boldsymbol{o}^t)$ is related to

$$\begin{aligned}
\boldsymbol{r}^{t-}&=\begin{cases}(r_{i,(j_{i,t}-1)},i\in\mathcal{I}_t), & \text{for }|\mathcal{I}_t|>0,\\0, & \text{otherwise,}\end{cases} && \text{(16a)}\\
\boldsymbol{u}^t&=(u_{i,t},i\in\mathcal{I}), && \text{(16b)}\\
\tilde{\boldsymbol{c}}^t&=(\tilde{c}_{i,t},i\in\mathcal{I}), && \text{(16c)}\\
\tilde{\boldsymbol{r}}^t&=(\tilde{r}_{i,t},i\in\mathcal{I}), && \text{(16d)}
\end{aligned}$$

where $\tilde{c}_{i,t}$ and $\tilde{r}_{i,t}$ denote the residual computation tasks and the residual transmission tasks of user i at time t, respectively.

Notice that the viewing gain $g_i(\cdot)$ and the degradation loss $l_1(\cdot)$ are deterministic given the bitrates, while the rebuffering loss is related to the future channel conditions and

the resource allocation decisions. As a result, the provider needs to maximize the expected QoE at each time t:

$$\tilde{Q}\left(\boldsymbol{r}^t,\boldsymbol{b}^t,\boldsymbol{o}^t|\boldsymbol{r}^{t-},\boldsymbol{u}^t,\tilde{\boldsymbol{c}}^t,\tilde{\boldsymbol{r}}^t\right)= \sum_{i\in\mathcal{I}_t}\left(g_i(r_{i,j_{i,t}})-l_1(r_{i,j_{i,t}}-r_{i,j_{i,t}-1})\right)- l_2\mathbb{E}_{\bar{d}_i,\bar{o}_i|\boldsymbol{r}^t,\boldsymbol{b}^t,\boldsymbol{o}^t}\left[\sum_{i\in\mathcal{I}}\left[\frac{\tilde{r}_{i,t}}{\bar{d}_i}+\frac{\tilde{c}_{i,t}}{\sigma\bar{o}_i}-u_{i,j-1}\right]^+\right], \tag{17}$$

where $\bar{\boldsymbol{d}}=\left(\bar{d}_i,\forall i\right)$ and $\bar{\boldsymbol{o}}=(\bar{o}_i,\forall i)$ denote the average downlink channel capacity and the average assigned CPU cycles, respectively.

B. Per-frame Objective Function

Considering that the distributional information of $h_{i,t}$ is not available, we replace the expectation function $\tilde{Q}$ by an estimation function $\hat{Q}$. More specifically, the re-buffering loss is estimated as:

$$l_2\mathbb{E}_{\bar{d}_i,\bar{o}_i|\boldsymbol{r}^t,\boldsymbol{b}^t,\boldsymbol{o}^t}\left[\sum_{i\in\mathcal{I}}\left[\frac{\tilde{r}_{i,t}}{\bar{d}_i}+\frac{\tilde{c}_{i,t}}{\sigma\bar{o}_i}-u_{i,j-1}\right]^+\right] \tag{18a}$$

$$\geq l_2\mathbb{E}_{\bar{d}_i,\bar{o}_i|\boldsymbol{r}^t,\boldsymbol{b}^t,\boldsymbol{o}^t}\left[\sum_{i\in\mathcal{I}}\left(\frac{\tilde{r}_{i,t}}{\bar{d}_i}+\frac{\tilde{c}_{i,t}}{\sigma\bar{o}_i}-u_{i,j-1}\right)\right] \tag{18b}$$

$$= l_2\sum_{i\in\mathcal{I}}\left[\mathbb{E}\left[\frac{\tilde{r}_{i,t}}{\bar{d}_i}\right]+\mathbb{E}\left[\frac{\tilde{c}_{i,t}}{\sigma\bar{o}_i}\right]-u_{i,j-1}\right] \tag{18c}$$

$$\approx l_2\sum_{i\in\mathcal{I}}\left[\frac{\tilde{r}_{i,t}}{b_i^t\log_2\left(1+\mathrm{SNR}_{i,t}\right)}+\frac{\tilde{c}_{i,t}}{\sigma o_{i,j}}-u_{i,j-1}\right]. \tag{18d}$$

- First, we eliminate the operator $[\cdot]^+$ involved in the re-buffering loss in (18b). This relaxation can help with characterizing the impact of the re-buffering in the subsequent frames. For example, the operator $[\cdot]^+$ may fail to capture the difference between a one-second re-buffering and a ten-seconds re-buffering, i.e., $[-1]^+=[-10]^+$. However, different re-buffering times may lead to different adaptation decisions in the subsequent frames.
- Second, the transmission and computation times of the segments in $\hat{Q}$, i.e., $\frac{\tilde{r}_{i,t}}{b_i^t\log_2(1+\mathrm{SNR}_{i,t})}$ and $\frac{\tilde{c}_{i,t}}{\sigma o_{i,j}}$ in (18d), are approximations of the actual transmission and computation times in $\tilde{Q}$. These approximations are introduced to estimate the actual time according to the causal information, and is close to the actual time [6][24].

Namely, the estimation $\hat{Q}$ can be expressed as:

$$\hat{Q}\left(\boldsymbol{r}^t,\boldsymbol{b}^t,\boldsymbol{o}^t|\boldsymbol{r}^{t-},\boldsymbol{u}^t,\tilde{\boldsymbol{c}}^t,\tilde{\boldsymbol{r}}^t\right)= \sum_{i\in\mathcal{I}_t}\left(g_i(r_{i,j_{i,t}})-l_1(r_{i,j_{i,t}}-r_{i,j_{i,t}-1})\right)- l_2\sum_{i\in\mathcal{I}}\left[\frac{\tilde{r}_{i,t}}{b_i^t\log_2\left(1+\mathrm{SNR}_{i,t}\right)}+\frac{\tilde{c}_{i,t}}{\sigma o_{i,j}}-u_{i,j-1}\right]. \tag{19}$$

For notation simplicity, we use $\hat{Q}\left(\boldsymbol{r}^t,\boldsymbol{b}^t,\boldsymbol{o}^t\right)$ to denote (19) in the rest of this paper. Based on the above mentioned properties of the estimation function $\hat{Q}$, the provider can make real-time decisions frame-by-frame, rather than solving the multi-frame problem as a whole.

C. Per-frame Subproblem

At time t, given $(\boldsymbol{r}^{t-},\boldsymbol{u}^t,\tilde{\boldsymbol{c}}^t,\tilde{\boldsymbol{r}}^t)$, the per-frame subproblem is to determine the bitrate decision, the bandwidth resource allocation decision, and the computation resource allocation decision by maximizing $\hat{Q}\left(\boldsymbol{r}^t,\boldsymbol{b}^t,\boldsymbol{o}^t\right)$, i.e.,

$$\text{(P4)}\quad \max_{\boldsymbol{r}^t,\boldsymbol{b}^t,\boldsymbol{o}^t}\quad \hat{Q}\left(\boldsymbol{r}^t,\boldsymbol{b}^t,\boldsymbol{o}^t\right), \tag{20a}$$

$$\text{s.t.}\quad r_i^t\in\{R_1,...,R_m\},i\in\mathcal{I}_t, \tag{20b}$$

$$\beta r_i^t\leq b_i^t\log_2\left(1+\mathrm{SNR}_{i,t}\right),i\in\mathcal{I}_t, \tag{20c}$$

$$\sum_{i\in\mathcal{I}}b_i^t\leq B, \tag{20d}$$

$$\sum_{i\in\mathcal{I}}o_i^t\leq o^{\max}, \tag{20e}$$

$$b_i^t,o_i^t\geq 0,\forall i. \tag{20f}$$

Afterwards, at time $t+1$, the provider updates the system state to $(\boldsymbol{r}^{(t+1)-},\boldsymbol{u}^{t+1},\tilde{\boldsymbol{c}}^{t+1},\tilde{\boldsymbol{r}}^{t+1})$ and then solves the per-frame subproblem at time $t+1$.

Remark 2. *The number of constraints in Problem (P4) is significantly reduced as follows: i) Since all the past decisions are feasible, we have $q_{i,3}(t-1)\leq q_{i,4}(t-1)$. This, together with the fact that $q_{i,3}(t-1)+\beta r_i^t=q_{i,3}(t)$ and $q_{i,4}(t-1)+b_i^t\log_2\left(1+SNR_{i,t}\right)=q_{i,4}(t)$, implies the sufficiency of (20c). ii) If $\tilde{c}_i^t\leq c_i^t$, we have $q_{i,1}(t)=q_{i,2}(t)$; if $\tilde{c}_i^t>c_i^t$, we have $q_{i,1}(t)=q_{i,1}(t-1)\leq q_{i,2}(t-1)\leq q_{i,2}(t)$. As a result, we can safely remove the sequential computation constraints (15c). iii) We also eliminates the decision variables $(\boldsymbol{\tau}^s,\boldsymbol{\tau}^f)$ and the corresponding constraints. This is because that under the online fashion, we have $\tau^s_{i,(j_{i,t}-1)}=t-1$, $\tau^s_{i,j_{i,t}}=t$, and $\tau^s_{i,j}\geq t+1,\forall j>j_{i,t}$ at time t for $i\in\mathcal{I}_t$.*

D. Low-Complexity Online Algorithm

To solve the Problem (P4) efficiently, we introduce a hidden decision variable $\boldsymbol{\gamma}^t=(\gamma_i^t=\frac{1}{b_i^t\log_2(1+\mathrm{SNR}_{i,t})},\forall i)$, and replace the discrete constraint (20b) with $R_1\leq r_{i,j}\leq R_m$. Accordingly, (P4) can be further transformed to

$$\text{(P5)}\quad \min_{\boldsymbol{r}^t,\boldsymbol{b}^t,\boldsymbol{\gamma}^t,\boldsymbol{o}^t}\quad -\hat{Q}\left(\boldsymbol{r}^t,\boldsymbol{b}^t,\boldsymbol{o}^t\right), \tag{21a}$$

$$\text{s.t.}\quad \frac{1}{b_i^t\log_2\left(1+\frac{P_o h_{i,t}}{b_i^t N_0}\right)}-\gamma_i^t\leq 0,\forall i, \tag{21b}$$

$$R_1\leq r_i^t\leq R_m,\forall i, \tag{21c}$$

$$\sum_{i\in\mathcal{I}}b_i^t\leq B,\forall i, \tag{21d}$$

$$\sum_{i\in\mathcal{I}}o_{i,t}\leq o^{\max}, \tag{21e}$$

$$b_i^t,o_i^t\geq 0,\forall i. \tag{21f}$$

We observe that the optimization objective is a convex function over $(\boldsymbol{r}^t,\boldsymbol{b}^t,\boldsymbol{\gamma}^t,\boldsymbol{o}^t)$. First, $-\hat{Q}\left(\boldsymbol{r}^t,\boldsymbol{b}^t,\boldsymbol{o}^t\right)$ is linear with $\boldsymbol{\gamma}^t$. Second, $g_i(\cdot)$ is concave [20] and $l_1(\cdot)$ is convex [6]. By the definition of convexity, any linear combination of convex functions is convex. We also observe the constraints (21b-f) are linear or convex functions of the decision variables. This enables the gradient descent methods to find the global optimum $(\boldsymbol{r}^*,\boldsymbol{b}^*,\boldsymbol{\gamma}^*,\boldsymbol{o}^*)$.

However, $\boldsymbol{r}^*$ obtained through (P5) does not sufficiently satisfy the discrete constraint (20a). In general, rounding the optimal real number solutions is not sufficient to obtaining the optimal integer solutions. Fortunately, we manifest that the optimal solution $\boldsymbol{r}^o$ to Problem (P4) can be easily obtained by rounding the optimal $\boldsymbol{r}^*$ of (P5) to the nearest integers in the following Lemma.

Lemma 2. *Given the optimal solution $(\boldsymbol{r}^*, \boldsymbol{b}^*, \boldsymbol{\gamma}^*, \boldsymbol{o}^*)$ and $(\boldsymbol{r}^o, \boldsymbol{b}^*, \boldsymbol{o}^*)$ are an optimal solution to Problem (P5) and Problem (P4), then $\boldsymbol{r}^o$ must be in the set $\theta = \{\boldsymbol{r} | r_i = \lfloor r_i^* \rfloor$ or $\lceil r_i^* \rceil, \forall i\}$. Here, $\lfloor x \rfloor$ and $\lceil x \rceil$ denote the largest integer no greater than x and the smallest integer no less than x.*

Proof. Please see the detailed proof in Appendix C. □

Lemma 2 indicates that we can obtain the optimal $\boldsymbol{r}^o$ by rounding $\boldsymbol{r}^*$. The next question we need to solve is how to calculate the optimal $(\boldsymbol{b}^*|_{\boldsymbol{r}^o}, \boldsymbol{\gamma}^*|_{\boldsymbol{r}^o}, \boldsymbol{o}^*|_{\boldsymbol{r}^o})$ given $\boldsymbol{r}^o$ efficiently. According to the formulation of Problem (P4) and (P5), we derive the following Corollary 1 by simple algebra. The following Corollary 1, together with the convexity of Problem (P4), implies that $(\boldsymbol{b}^*|_{\boldsymbol{r}^o}, \boldsymbol{\gamma}^*|_{\boldsymbol{r}^o}, \boldsymbol{o}^*|_{\boldsymbol{r}^o})$ can be easily obtained using convex optimization methods.

Corollary 1. *Given the optimal solution $(\boldsymbol{r}^o, \boldsymbol{b}^*, \boldsymbol{o}^*)$ of (P4), the decision $(\boldsymbol{r}^o, \boldsymbol{\gamma}|_{\boldsymbol{b}^*}, \boldsymbol{b}^*, \boldsymbol{o}^*)$ is also an optimal solution of (P5).*

In the following, we propose a 5-step online algorithm, called Online joint Computation-Communication resource allocation Algorithm (OCCA). The Lemma 2, Corollary 1, and the convexity of Problem (20) imply that OCCA is robust and converged. A pseudo code of OCCA is presented in Algorithm 1 and explained as follows.

- *Step 1*: At the beginning of each time frame, the provider solves (P5) with the updated parameters.
- *Step 2*: The provider constructs the candidate optimal solution set as θ.
- *Step 3*: For each candidate in θ, the provider solves (P4).
- *Step 4*: The provider selects the one with maximum QoE from the parallel problems, and determine the bitrate selection, and the bandwidth and computation resource allocation according to the selected solution in this time frame.
- *Step 5*: The provider updates the segment index, the last bitrate, the current buffer length $\boldsymbol{u}$, the residual transcoding task $\tilde{\boldsymbol{c}}$, and the residual transmission task $\tilde{\boldsymbol{r}}$.

Algorithm 1 Online joint Computation-Communication resource allocation Algorithm (OCCA)

Initialization: $\boldsymbol{u} = \boldsymbol{0}, \tilde{\boldsymbol{c}} = \boldsymbol{0}, \tilde{\boldsymbol{r}} = \boldsymbol{0}$, index$= \boldsymbol{0}$, last$= \boldsymbol{0}$
1: **while** $\exists i \in \mathcal{I}, \text{index}_i \leq |\mathcal{J}_i|$ or $\tilde{\boldsymbol{c}}_i > 0$ or $\tilde{\boldsymbol{r}}_i > 0$ **do**
2: Solve (P5) with the optimal solution $(\boldsymbol{r}^*, \boldsymbol{b}^*, \boldsymbol{\gamma}^*, \boldsymbol{o}^*)$.
3: Set $\theta = \{\boldsymbol{r} | r_i = \lfloor r_i^* \rfloor$ or $\lceil r_i^* \rceil, \forall i\}$.
4: For each candidate in θ, solve (P4).
5: Select the one with the maximum QoE: $\boldsymbol{r}^o$, and set the bitrate selection, and bandwidth and computation resource allocation accordingly.
6: Updates $\boldsymbol{u}, \tilde{\boldsymbol{c}}, \tilde{\boldsymbol{r}}$, index log and last bitrate.
7: **end while**
Output: $(\boldsymbol{r}^o, \boldsymbol{b}^*|_{\boldsymbol{r}^o}, \boldsymbol{o}^*|_{\boldsymbol{r}^o})$

V. Performance Evaluation

In this section, we investigate the performance of our proposed OCCA algorithm. All the computations are executed on a machine with an Intel i7-10700KF CPU, a Nvidia RTX 2070S GPU, and 64GB RAM. The reinforcement learning (RL) benchmark algorithms are constructed using Pytorch [25] for computational speed boost. In the following, we first introduce the datasets and experiment setup. Then, we present the simulation results, including the video streaming instance using the proposed OCCA algorithm, and the performance comparison between OCCA and benchmark methods under different ß. The complete source code implementing OCCA and benchmark algorithms are available at https://github.com/wsyCUHK/computation-and-communication-resource-allocation.

Table II: Parameters of different types of video streaming users.

Type	TV	Smart Phone	Laptop
a	1	0.5	0.55
b	0.25	5	2.5
c	2	3	2.5

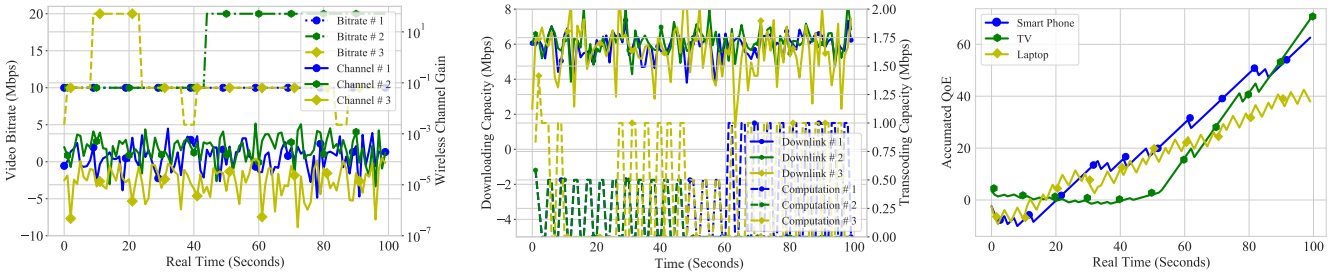
A. Experiment Setup

In all simulations, we use the parameters of Huawei BTS3900A. Without loss of generality, we set the transmission power $P = 120W$ (Watt) and the carrier frequency $f_c = 915$ MHz. Unless otherwise stated, we consider a Rayleigh fading channel model, where the channel gain $h_{i,t} = \bar{h}_i \alpha$. Here, $\bar{h}_i$ denotes the average channel gain determined by the location of the user i, and α denotes an independent exponential random variable of unit mean. Specifically, $\bar{h}_i$ follows the free-space path loss model, i.e.,

$$\bar{h}_i = G_A \left(\frac{3 \times 10^8}{4\pi f_c d_i} \right)^{d_e}, i \in \mathcal{I}, \tag{22}$$

where G_A and d_i denote the antenna gain and the distance from the user i to the transmit antenna, respectively. Here, d_e denotes the path loss exponent. In addition, we set $G_A = 79.43$ and $d_e = 2$ in our simulation.

We set the video parameters according to the public datasets: i) the segment length β is set to 2 seconds according to the dataset in [26]; ii) the video streaming time window is set to 100 seconds according to Instagram [27] and YouTube Highlight [28], which are two of the most popular mobile video applications. There are four video layers of the video, with R_1 as the cached highest layer at the MEC server. The bitrates of the three transcoding layers are $R_2 = 10$ Mbps, $R_3 = 5$ Mbps, and $R_4 = 1$ Mbps [18]. Video transcoding from R_1 to R_2, R_3, and R_4 needs 1, 2, and 6 CPU billion cycles per second [18], respectively. The viewing gain is modeled as $g_i(r) = a_i \log_{c_i}(b_i \cdot r + 1)$. The users are divided into three types, namely TV, smart phone, laptop



(a) Bitrate and Wireless Channel Gain (b) Downloading and Transcoding Capacity (c) Accumulated QoE

Figure 4: A video streaming instance with OCCA algorithm. The first, second, and third users are smart phone, TV, and laptop user, respectively.

users. The parameters are listed in Table II. The degradation loss function and re-buffering time loss are modeled as $l_1(\delta) = \begin{cases} 0.01\delta^2 + 0.05\delta, \text{if } \delta > 0, \\ 0, \text{otherwise} \end{cases}$ and $l_2 = 0.5$, respectively.

We evaluate OCCA to answer the following questions:

- How does OCCA coordinate the transcoding and transmission tasks in the heterogeneous network, especially under highly dynamic wireless channel? (Section V.B)
- Does OCCA improve the QoE of the adaptive streaming for mobile users, especially under limited network resources? (Section V.C)
- How does each component of the QoE contribute to the overall QoE improvement? (Section V.C)

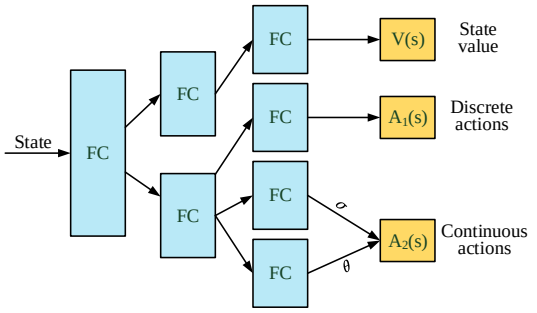


Figure 5: The mixed continuous-discrete actions A3C network in our simulation. FC: Fully-Connected Layer; θ: the means of the actions; σ: the variance of the actions.

B. Streaming Instance Analysis

In the first experiment, we investigate how the adaptive video streaming works under the proposed OCCA algorithm.

In Fig. 4, we demonstrate an instance to visualize the computation-communication joint optimized video streaming system under the proposed OCCA algorithm. To illustrate, we consider a system with 3 users and 2 MHz total bandwidth. The distance d_i from the user to the base station is independently generated from a truncated Gaussian distribution as $d_i = \min\left(\max\left(X, \bar{d} - 30\right), \bar{d} + 30\right)$, where X is a Gaussian random variable and $\bar{d}$ denotes the average distance. Specifically, there are one smart phone user (# 1), one TV user (# 2), and one laptop user (# 3).

From the wireless channel's perspective, we observe that when the wireless channel gain is large, the downloading capacity is relatively high and the transcoding data rate σo is relatively small, and vice versa. On one hand, when the wireless channel gain increasing rapidly, the video streaming may adapt to a higher bitrate. On the other hand, when the wireless channel gain varies within a certain range, the video streaming prefers the current bitrate. That is, *enabled by OCCA, the users prefer to pre-download and cache the future segments with the current bitrates to cancel out the channel variation* unless the channel has been increasing by a wide margin.

From the heterogeneous users' perspective, the smart phone user prefers a smooth playback because the marginal viewing gain is limited when the bitrate is already high. Instead, the laptop and TV users are more likely to adapt to a higher bitrate when the channel condition is good. Correspondingly, as shown in Fig. 4c, the laptop and TV users can achieve a high accumulated QoE when the channel is relatively good, but may suffer a low QoE due to high degradation when the channel is relatively poor.

In brief, we observe an inverse correlation between the downloading capacity and the transcoding rate over heterogeneous users. This indicates that OCCA successfully coordinates the transcoding and transmission tasks in the heterogeneous MEC-enabled video streaming system to achieve a higher QoE. Moreover, the laptop user still achieves *a relatively high QoE (Fig. 4c) under highly dynamic wireless channel (Fig. 4a) due to the stable adaptation under OCCA* .

C. System-Level Evaluation

In the second experiment, we compare the proposed OCCA algorithm with four representative benchmark methods[4]:

- Deep Q-network (DQN) [18]: the provider uses deep learning to estimate the state-value function and performs the decision based on the approximated state-value function. According to [18], there are five layers in the deep neural network, including one input layer, one output

[4]The Pytorch implementations of the four benchmark algorithms are available at https://github.com/wsyCUHK/computation- and-communication-resource-allocation.

layer, one convolution layer with 3×3 filters, and two hidden layers in our simulation.

- Asynchronous Advantage Actor-Critic (A3C) [17]: the provider uses Advantage Actor-Critic framework to estimate the state function and find the action policy, i.e., the critic and the actor, that maximize the defined reward in an asynchronous manner. Different from the continuous actor in [17], we implement a mixed continuous-discrete actor due to the discrete bitrate action space, and the continuous bandwidth and computation allocation action spaces. The detailed structure of the four-layer deep neural network is shown in Fig. 5.
- Buffer-Aware Algorithm (BAA) [15]: the provider controls the video quality adaptation only according to the playback buffer. In our simulation, we assume equal allocation of bandwidth. The buffer-aware algorithm selects the lowest bitrate if the playback buffer is no greater than 10 seconds, and selects the highest bitrate if the playback buffer excesses 30 seconds.
- Without Transcoding Policy (WTP): the provider always assigns the highest bitrate and only optimizes the bandwidth allocations. In our simulation, we set $\boldsymbol{r}^t = (R_1, ..., R_1)$ and $\boldsymbol{o}^t = (0, ..., 0)$, and optimize the bandwidth allocation in (P5).

In Fig. 6 and 7, we compare the average QoE, bitrate, degradation loss, and re-buffering loss of different schemes under different network setups. Each point in Fig. 6 and 7 is an average performance of 100 independent user placement sets, while the value of each placement is accumulated or averaged over 100 sequential Rayleigh channel fading realizations. For the RL methods, i.e., DQN and A3C, we first train the models at the beginning 10000 seconds for different locations with independent random initial weights. From the 10001st second to 10100th second, we fix the parameters and test the RL methods.

1) Effect of Users: In Fig. 6, we set the total bandwidth as 2 MHz and compare the performance metrics, i.e., QoE, bitrate, degradation loss, and re-buffering loss, when the number of video streaming users $|\mathcal{I}|$ increases from 1 to 5. From Fig. 6a, we observe that on one hand the QoEs achieved by OCCA and A3C increase as $|\mathcal{I}|$ increases. On the other hand, the QoEs achieved by BBA, WTP, and DQN decrease as $|\mathcal{I}|$ increases. In comparison, the gaps between OCCA and the benchmark algorithms widen when $|\mathcal{I}|$ increases. This is because that with a large number of users, the OCCA and A3C methods have more flexibility to transcode the segments and adapt the bitrates. In contrast, other benchmark algorithms, especially the BBA method, suffer large degradation and re-buffering loss when more users share the limited bandwidth and CPU resources.

Overall, the proposed OCCA method always achieves the best QoE for all values of $|\mathcal{I}|$, providing 97.84% extra QoE compared to the best benchmark A3C. The performance advantage is especially notable when $|\mathcal{I}|$ is large. On average, the QoEs of OCCA are 78.9%, 86.8%, 99.6%, 104.6%, 119.3% higher than the ones of A3C for $|\mathcal{I}| = 1, 2, 3, 4, 5$, respectively. The WTP and DQN methods achieve (near) negative QoEs. This is because that i) A3C undergoes a very large degradation loss (Fig. 6c) due to the aggressive bitrate adaptation; ii) WTP suffers a very large re-buffering loss (Fig. 6d) because the downloading data rate is much less than the video bitrate especially when the wireless channel is poor; iii) DQN achieves both large re-buffering and degradation loss for discritizing the continuous space actions, which leads to a prohibitively large Q-table in value-based RL methods.

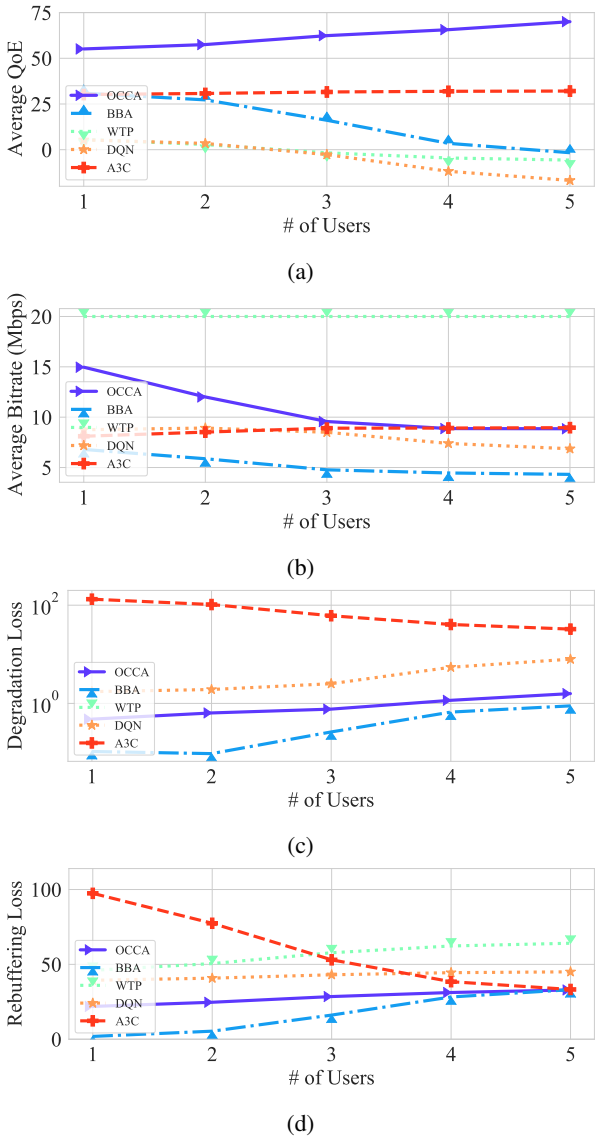


Figure 6: Comparisons between OCCA and the benchmark methods versus the number of users.

2) Effect of Total Bandwidth: In Fig. 7, we compare the performance metrics when $|\mathcal{I}| = 3$ and the total bandwidth varies from 1 to 5 MHz. As shown in Fig. 7a, the QoEs increase as the total bandwidth increases. Specifically, the QoE achieved by DQN increases rapidly when the total bandwidth increases from 1 to 4 MHz. Meanwhile, there is always an evident performance gap between OCCA and benchmark methods. On average, the QoE of OCCA is 85.8% higher than A3C. This is due to the following three reasons. First, the benchmark algorithms suffer large re-buffering loss (Fig. 7d) and thus a

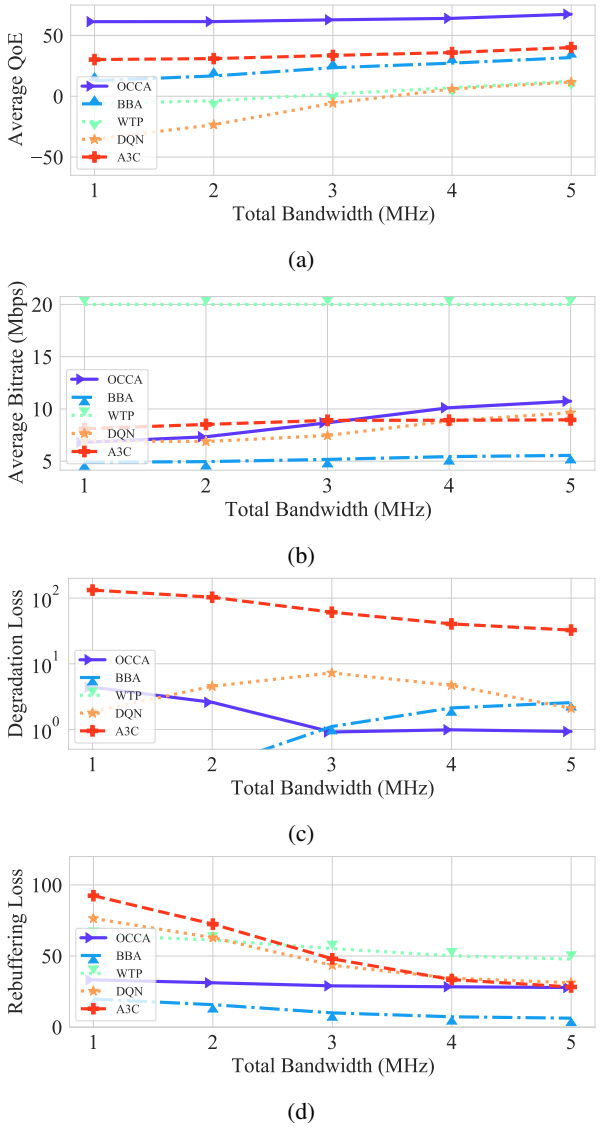


Figure 7: Comparisons between OCCA and the benchmark methods versus the total bandwidth.

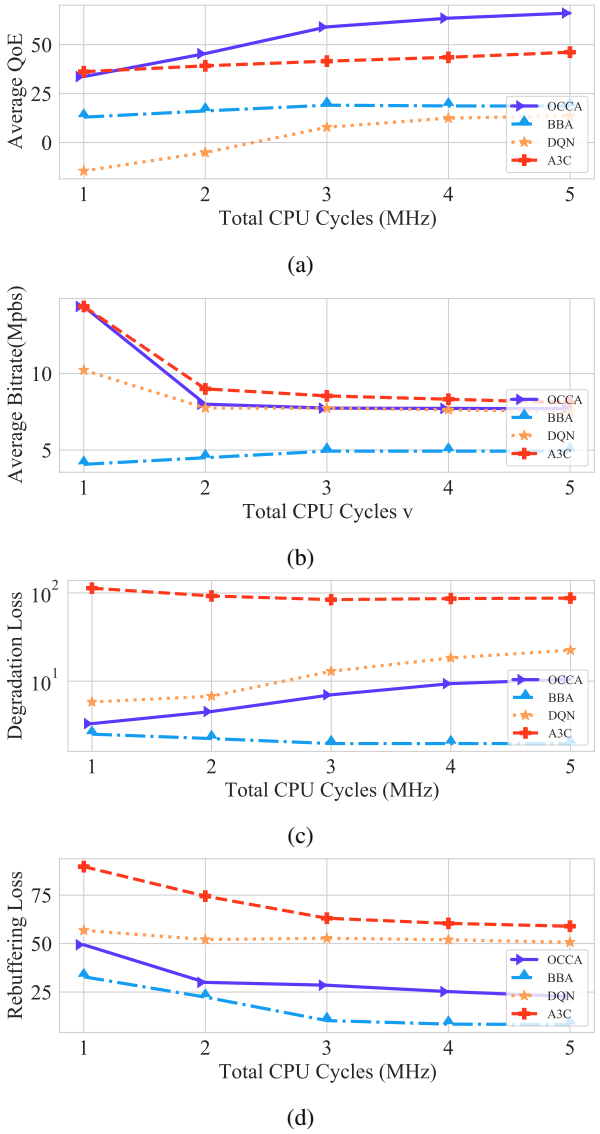


Figure 8: Comparisons between OCCA and the benchmark methods versus the CPU resource.

gracious bandwidth allocation could significantly improves the re-buffering. Second, *the bandwidth and transcoding coordination could not only achieve a smooth playback with less re-buffering, but also shrink the degradation (Fig. 7c) through stabilizing the downloading capacity by bandwidth allocation*. Third, the RL methods typically converge to a local optimal policy that sacrifices the average bitrate (Fig. 7b) and the degradation loss (Fig. 7c) for the re-buffering loss, especially for highly dynamic wireless channel.

In Fig. 7, overall, we see that the proposed OCCA method significantly outperforms the benchmark methods under different total bandwidth settings. *The performance advantage is especially notable when the communication resource is limited, i.e., the total bandwidth is small.*

3) Effect of CPU: In Fig. 8, we compare the performance metrics when $|\mathcal{I}| = 3, B = 2MHz$ and $o^{\max}$ varies from 1 to 5 MHz. The performance metrics of WTP are omitted, since the WTP method does not perform any transcoding. As shown in Fig. 8a, the QoEs increase as the total available CPU cycles increase. Overall, we see that the proposed OCCA method always achieves the best QoE for all values of $o^{\max}$. The performance advantage is especially notable when $o^{\max}$ is large. On average, the QoEs of OCCA are 0.2%, 19.7%, 42.1%, 43.3%, and 45.9% higher than the ones of A3C for $o^{\max} = 1, 2, 3, 4, 5$ MHz, respectively. This is due to the following two reasons. First, when the transcoding capability is low, all the methods suffer a large re-buffering loss due to the high bitrates. The re-buffering losses of OCCA and BBA are relatively small because of the proper bandwidth allocation by Algorithm 1. Second, when the transcoding capability is increasing, OCCA, A3C, and DQN benefit from online transcoding and enjoy smoother playbacks. However, the A3C and DQN suffer large degradation losses due to the high bitrate fluctuations.

In Fig. 8, overall, we see that the proposed OCCA method significantly outperforms the benchmark methods under dif-

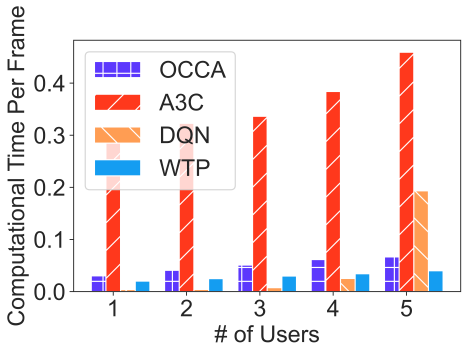


Figure 9: Computational complexity comparison.

ferent total CPU cycles settings. The average bitrate of A3C is higher than OCCA. This implies that *the users prefer not only higher bitrates but also smoother playbacks, fully utilizing the downloading capacity in each time frame may not be the optimal solution.*

4) Summarization: In both Fig. 6 and 7, we observe that WTP achieves similar rebuffering loss compared to DQN and A3C. It is noteworthy that the transmission data rate is much high without transcoding. Therefore, the experiment also evidences that *a gracious bandwidth allocation can significantly improve the video streaming QoE.*

To summarize, the performance of the considered benchmark methods (Fig. 6 and 7) are sensitive to the network parameters, i.e., the number of users and the total bandwidth. The benchmark methods may produce very poor performance in some practical setups, e.g., limited total bandwidth and massive video streaming users. In contrast, regardless of the network condition, the proposed OCCA method jointly coordinates the communication and computation among the users. As a result, the proposed OCCA method always achieves a high bitrate (Fig. 6b and 7b), a relatively low degradation loss (Fig. 6c and 7c), and a limited re-buffering loss (Fig. 6d and 7d).

D. Computational Complexity

In the third experiment, we compare the proposed OCCA algorithm with the non-threshold benchmarks, i.e., A3C, DQN, and WTP. Here, we plot the average computational time per frame consumed by OCCA and the benchmark algorithms. Specifically, for each frame in WTP, the bitrate is fixed and Algorithm 1 is also executed to solve the bandwidth allocation optimization problem in Problem (P4). Each bar in Fig. 9 is an average performance of 100 independent user placement sets, while the value of each placement is averaged over 100 sequential Rayleigh channel fading realizations. For fairness, all the methods are under the only-CPU model.

In Fig. 9, we observe that the computational times in both OCCA and WTP increase with $|\mathcal{I}|$ at a sub-linear rate. In contrast, the ones in A3C and DQN exponentially increase with $|\mathcal{I}|$. This is because, on one hand, the action space in A3C and DQN increases exponentially with the number of users; on the other hand, the dimension of the decision variables in OCCA and WTP increases linearly with the number of users. When $|\mathcal{I}| = 5$, the average computational time achieved by OCCA, A3C, DQN, and WTP are 0.067, 0.459, 0.193, and 0.040 seconds, respectively. This implies that OCCA is *a scalable and efficient online algorithm even in large-sized networks*.

VI. Conclusions

In this paper, we considered a joint bitrate adaptation, transcoding and transmission resource allocation problem for MEC-enabled streaming. We formulated the joint bitrate selection, bandwidth allocation, and CPU cycle assignment problem as a non-convex MINLP problem. To reduce the computational complexity, we introduced four queueing model constraints and analytically proved the queueing model optimality. In absence of the distributional information, we further decoupled the non-convex MINLP problem and derived a low-complexity online algorithm to maximize QoE for heterogeneous users, called OCCA. The experimental evaluation performed on a practical scenario manifested that OCCA considerably outperforms the benchmark algorithms in terms of QoE, while owning a high average bitrate, reducing the re-buffering time, and shrinking the bitrate fluctuation. Furthermore, the results indicated that the joint communication and computation coordination enables users to watch high-QoE videos even under highly dynamic wireless channels.

References

[1] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update, 2017-2022," in *White Paper*, Feb 2019.

[2] E. Maani, P. V. Pahalawatta, R. Berry, T. N. Pappas, and A. K. Katsaggelos, "Resource allocation for downlink multiuser video transmission over wireless lossy networks," *IEEE Transactions on Image Processing*, vol. 17, no. 9, pp. 1663–1671, 2008.

[3] L. Wei, J. Cai, C. H. Foh, and B. He, "Qos-aware resource allocation for video transcoding in clouds," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 1, pp. 49–61, 2017.

[4] Y. Sani, A. Mauthe, and C. Edwards, "Adaptive bitrate selection: A survey," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2985–3014, 2017.

[5] M. Tang, H. Pang, S. Wang, L. Gao, J. Huang, and L. Sun, "Multi-dimensional auction mechanisms for crowdsourced mobile video streaming," *IEEE/ACM Transactions on Networking*, vol. 26, no. 5, pp. 2062–2075, 2018.

[6] M. Tang and V. W. S. Wong, "Online bitrate selection for viewport adaptive 360-degree video streaming," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2020.

[7] J. Zhang, H. Wu, X. Tao, and X. Zhang, "Adaptive bitrate video streaming in non-orthogonal multiple access networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 3980–3993, 2020.

[8] Apple. HTTP live streaming. Accessed on Oct. 10, 2020. [Online]. Available: https://developer.apple.com/streaming/

[9] I. Standard, "Dynamic adaptive streaming over HTTP (DASH)-part 1: Media presentation description and segment formats," *ISO/IEC*, pp. 23 009–1, 2014.

[10] W. Shi and S. Dustdar, "The promise of edge computing," *Computer*, vol. 49, no. 5, pp. 78–81, 2016.

[11] L. Yu, H. Shen, K. Sapra, L. Ye, and Z. Cai, "Core: Cooperative end-to-end traffic redundancy elimination for reducing cloud bandwidth cost," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 2, pp. 446–461, 2017.

[12] H. Zhang, A. Zhou, J. Lu, R. Ma, Y. Hu, C. Li, X. Zhang, H. Ma, and X. Chen, "Onrl: Improving mobile video telephony via online reinforcement learning," in *Proceedings of ACM MobiCom 2020*, New York, NY, USA, Sep. 2020.

[13] D. Wang, Y. Peng, X. Ma, W. Ding, H. Jiang, F. Chen, and J. Liu, "Adaptive wireless video streaming based on edge computing: Opportunities and approaches," *IEEE Transactions on Services Computing*, vol. 12, no. 5, pp. 685–697, 2019.

[14] M. Rugelj, U. Sedlar, M. Volk, J. Sterle, M. Hajdinjak, and A. Kos, "Novel cross-layer QoE-aware radio resource allocation algorithms in multiuser ofdma systems," *IEEE Transactions on Communications*, vol. 62, no. 9, pp. 3196–3208, 2014.

[15] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A buffer-based approach to rate adaptation: Evidence from a large video streaming service," in *Proceedings of ACM SIGCOMM 2014*, New York, NY, USA, Aug. 2014, p. 187–198.

[16] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over HTTP," in *Proceedings of ACM SIGCOMM 2015*, New York, NY, USA, Aug 2015, p. 325–338.

[17] J. Luo, F. R. Yu, Q. Chen, and L. Tang, "Adaptive video streaming with edge caching and video transcoding over software-defined mobile networks: A deep reinforcement learning approach," *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 1577–1592, 2020.

[18] Y. Guo, F. R. Yu, J. An, K. Yang, C. Yu, and V. C. M. Leung, "Adaptive bitrate streaming in wireless networks with transcoding at network edge using deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 3879–3892, 2020.

[19] H. Yeo, C. J. Chong, Y. Jung, J. Ye, and D. Han, "Nemo: Enabling neural-enhanced video streaming on commodity mobile devices," in *Proceedings of ACM MobiCom 2020*, New York, NY, USA, Sep. 2020.

[20] Y. Zhang, P. Zhao, K. Bian, Y. Liu, L. Song, and X. Li, "DRL360: 360-degree video streaming with deep reinforcement learning," in *IEEE INFOCOM 2019*, April 2019, pp. 1252–1260.

[21] F. Raphel and S. M. Sameer, "A speed adaptive joint subcarrier and power allocation technique for downlink ofdma video transmission over doubly selective channels," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 1879–1887, 2020.

[22] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over HTTP," in *Proceedings of SIGCOMM 2015*, vol. 45, no. 4, New York, NY, USA, Aug. 2015, p. 325–338.

[23] A. Mehrabi, M. Siekkinen, and A. Ylä-Jääski, "Edge computing assisted adaptive mobile video streaming," *IEEE Transactions on Mobile Computing*, vol. 18, no. 4, pp. 787–800, 2019.

[24] Speedtest. Speedtet market report: Canada average mobile download speed based on q2-q3 data in 2019. Accessed on Sep. 9, 2020. [Online]. Available: https://www.speedtest.net/reports/canada/

[25] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Proceedings of NeurIPS 2019*, Vancouver, Canada, Dec. 2019, pp. 8026–8037.

[26] A. Elgabli, M. Felemban, and V. Aggarwal, "Groupcast: Preference-aware cooperative video streaming with scalable video coding," *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 1138–1150, 2019.

[27] B. Xiong, Y. Kalantidis, D. Ghadiyaram, and K. Grauman, "Less is more: Learning highlight detection from video duration," in *Proceedings of CVPR 2019*, Long Beach, CA, USA, June 2019.

[28] M. Sun, A. Farhadi, and S. Seitz, "Ranking domain-specific highlights by analyzing edited videos," in *Proceedings of ECCV 2014*, Zurich, Switzerland, Sep. 2014, pp. 787–802.

APPENDIX A
PROOF OF THEOREM 1

Proof. (a) We first prove that there always exists an optimal solution $(\boldsymbol{r}, \boldsymbol{B}, \boldsymbol{o}, \boldsymbol{\tau}^s, \boldsymbol{\tau}^f, \hat{\tau}^s_{i,j}, \hat{\tau}^f_{i,j})$ of Problem (P1) that satisfies the following inequality

$$\hat{\tau}^s_{i,j} \geq \hat{\tau}^f_{i,j-1}, \forall j \in \mathcal{J}. \tag{23}$$

We assume that there exists a feasible solution $\phi = \left(\boldsymbol{r}, \boldsymbol{B}, \boldsymbol{o}, \boldsymbol{\tau}^s, \boldsymbol{\tau}^f, \{\hat{\tau}^s_{i,j}\}_{j\in\mathcal{J}}, \{\hat{\tau}^f_{i,j}\}_{j\in\mathcal{J}}\right)$, where $\hat{\tau}^s_{i,j'} < \hat{\tau}^f_{i,j'-1}$. We denote

$$\begin{aligned}\phi^{'} = \Big(&\boldsymbol{r}, \boldsymbol{B}, \boldsymbol{o}, \boldsymbol{\tau}^s, \boldsymbol{\tau}^f, \\ &\{\hat{\tau}^s_{i,1}, ..., \hat{\tau}^s_{i,j-2}, \hat{\tau}^s_{i,j}, \hat{\tau}^s_{i,j-1}, \hat{\tau}^s_{i,j+1}, ..., \hat{\tau}^s_{i,|\mathcal{I}|}\}, \\ &\{\hat{\tau}^f_{i,1}, ..., \hat{\tau}^f_{i,j-2}, \hat{\tau}^f_{i,j}, \hat{\tau}^f_{i,j-1}, \hat{\tau}^f_{i,j+1}, ..., \hat{\tau}^f_{i,|\mathcal{I}|}\}\Big).\end{aligned} \tag{24}$$

Substituting $\phi^{'}$ into (3) and (12a-d,f-i), we have $\phi^{'}$ also satisfies (23) and achieves the same optimal overall QoE. Repeating the above process, for any set of optimal solutions, we can always find an optimal solution that satisfies (23).

(b) We second prove that there always exists an optimal solution

$$\left(\boldsymbol{r}, \boldsymbol{B}, \boldsymbol{o}, \boldsymbol{\tau}^s, \boldsymbol{\tau}^f, \{\hat{\tau}^s_{i,j}\}_{j\in\mathcal{J}}, \{\hat{\tau}^f_{i,j}\}_{j\in\mathcal{J}}\right) \tag{25}$$

of Problem (P1) that satisfies the inequality (13b). Taking the summation of (12e) over j, we have $\sum_{j'=1,...,j} \sigma \sum_{t\in[\hat{\tau}^s_{i,j}, \hat{\tau}^f_{i,j}]} o_{i,t} \geq \sum_{j'=1,...,j} \mu(r_{i,j})\beta$. Substituting (23) into the above inequality yields $\sum_{j'=1,...,j} \mu(r_{i,j'})\beta \leq \sigma \sum_{t=1,...,\hat{\tau}^f_{i,j}} o_{i,t}$. This, together with (1), implies $\sum_{j'=1,...,j} \mu(r_{i,j'})\beta \leq \sigma \sum_{t=1,...,\tau^s_{i,j}} o_{i,t}$.

(c) Similarly, substituting (23) and (1) into (13b) with simple algebra, we can prove by induction that there always exists an optimal solution of Problem (P2) that satisfies the inequality (12e).

Combining (b) and (c), together with the fact that objective function and constraints (12b-d,f-h) are independent of $\hat{\tau}^s$ and $\hat{\tau}^f$, we finish the proof of Theorem 1. □

APPENDIX B
PROOF OF LEMMA 1

Proof. a) We first prove that the optimal value of Problem (P2) is equal to the optimal value of the following problem:

$$\max_{\phi_2} \quad Q\left(\boldsymbol{r}, \boldsymbol{\tau}^s, \boldsymbol{\tau}^f, \boldsymbol{u}\right) \tag{26a}$$

$$\text{s.t.} \quad \beta \sum_{j'=1,...,j} \mu(r_{i,j'}) \leq \sigma \sum_{t=1,...,\tau^s_{i,j}} o_{i,t}, \forall i, j, \tag{26b}$$

$$\beta \sum_{j'=1,...,j} r_{i,j} \leq \sum_{t=1}^{\tau^f_{i,j}} B_{i,t} \log_2\left(1 + \text{SNR}_{i,t}\right), \forall i, j, \tag{26c}$$

$$\text{constraints (13b, d, f-h)(13c).} \tag{26d}$$

By simple algebra similar to Theorem 1, we have: (i) substituting (3) into the summation form of (12e) yields that there always exists an optimal solution of Problem (P2) that satisfies the inequality (26b). (ii) we can prove by induction that there always exists an optimal solution of Problem (26) that satisfies the inequality (12e). Combining (i) and (ii), we finish the proof of (a).

b) We second prove that we can always find an optimal solution of Problem (3) given the optimal solution of (P2), and vice versa. Given i and $\tau^f_{i,j} \leq t < \tau^f_{i,j+1}$, we have $\{j^{'} | \tau^f_{i,j'} \leq t\} = \{1, 2, ..., j\}$ because of the sequential downloading constraint (12f). On one hand, substituting $t = \tau^f_{i,j}$ into (15b) yields (26b). On the other hand, for any $\tau^{'} > 0$, substituting $\sum_{t=\tau^f_{i,j}+1}^{\tau^f_{i,j}+\tau^{'}} B_{i,t} \log_2\left(1 + \text{SNR}_{i,t}\right) \geq 0$ into (26b) yields $q_{i,1}(\tau^f_{i,j} + \tau^{'}) \leq q_{i,2}(\tau^f_{i,j} + \tau^{'})$. In conclusion, constraint (15b) is equivalent to (26b). Similarly, we have that constraint (15c) is equivalent to (26c).

Combining (a) and (b), we finish the proof of Lemma 1. □

APPENDIX C
PROOF OF LEMMA 2

Proof. For notation simplicity, we use $Q_1(\boldsymbol{r})$, $Q_2(\boldsymbol{r})$, and $\boldsymbol{r}_{-i}$ to denote the optimal values of Problem (P5) and Problem (P4) given r, and the bitrate decisions except user i in this proof, respectively. Moreover, we also define $Q_\gamma\left(\boldsymbol{r},\boldsymbol{\gamma},\boldsymbol{o}\right) = \hat{Q}\left(\boldsymbol{r},\boldsymbol{b},\boldsymbol{o}|\,\boldsymbol{\gamma}\right)$.

First, given (i,t), we show that for any $\boldsymbol{r} \in \{\boldsymbol{r}|\boldsymbol{r}_{-i} = \boldsymbol{r}^*_{-i}, r_i < \lfloor r_i^* \rfloor\}$, $Q_1(\boldsymbol{r}) \leq Q_1\left(\left(\lfloor r_i^* \rfloor, \boldsymbol{r}^*_{-i}\right)\right)$. It's equivalent to showing that for any $\boldsymbol{r} \in \{\boldsymbol{r}|\boldsymbol{r}_{-i} = \boldsymbol{r}^*_{-i}, r_i < \lfloor r_i^* \rfloor\}$, we can find an $\left(\left(\lfloor r_i^* \rfloor, \boldsymbol{r}^*_{-i}\right), \boldsymbol{\gamma}_1, \boldsymbol{o}_1\right)$ such that $Q_\gamma\left(\left(\lfloor r_i^* \rfloor, \boldsymbol{r}^*_{-i}\right), \boldsymbol{\gamma}_1, \boldsymbol{o}_1\right) \geq Q_1(r)$. In the feasible decision space of Problem (P5), we can always find a point $\left(\left(\lfloor r_i^* \rfloor, \boldsymbol{r}^*_{-i}\right), \boldsymbol{\gamma}_1, \boldsymbol{o}_1\right)$ in the line segment between $\left(\boldsymbol{r}, \boldsymbol{\gamma}_r^*, \boldsymbol{o}_r^*\right)$ and $\left(\boldsymbol{r}^*, \boldsymbol{\gamma}^*, \boldsymbol{o}^*\right)$. The continuousness of (P5) guarantees the existence of $\left(\left(\lfloor r_i^* \rfloor, \boldsymbol{r}^*_{-i}\right), \boldsymbol{\gamma}_1, \boldsymbol{o}_1\right)$. Due to the joint convexity of Q_γ, we have $Q_\gamma(\boldsymbol{r}, \boldsymbol{\gamma}_r^*, \boldsymbol{o}_r^*) \leq Q_\gamma\left(\left(\lfloor r_i^* \rfloor, \boldsymbol{r}^*_{-i}\right), \boldsymbol{\gamma}_1, \boldsymbol{o}_1\right) \leq Q_\gamma(\boldsymbol{r}^*, \boldsymbol{\gamma}^*, \boldsymbol{o}^*)$. Due to the equivalence of the objective function between Problem (P4) and (P5), we have $Q_\gamma(\boldsymbol{r}, \boldsymbol{\gamma}_r^*, \boldsymbol{o}_r^*) \leq Q_\gamma(\left(\lfloor r_i^* \rfloor, \boldsymbol{r}^*_{-i}\right), \boldsymbol{\gamma}_1, \boldsymbol{o}_1) \leq Q_\gamma(\boldsymbol{r}^*, \boldsymbol{\gamma}^*, \boldsymbol{o}^*)$.

Likewise, we can prove that for any $\boldsymbol{r} \in \{\boldsymbol{r}|\boldsymbol{r}_{-i} = \boldsymbol{r}^*_{-i}, r_i > \lceil r_i^* \rceil\rfloor\}$, we can find an $\left(\left(\lceil r_i^* \rceil, \boldsymbol{r}^*_{-i}\right), \boldsymbol{\gamma}_2, \boldsymbol{o}_2\right)$ such that $Q_\gamma\left(\left(\lceil r_i^* \rceil, \boldsymbol{r}^*_{-i}\right), \boldsymbol{\gamma}_2, \boldsymbol{o}_2\right) \geq Q_1(\boldsymbol{r})$. Therefore, we can conclude that the optimal $\boldsymbol{r}^o$ must be in the set $\{\boldsymbol{r}|r_i = \lfloor r_i^* \rfloor$ or $\lceil r_i^* \rceil, \forall i\}$. □

Shuoyao Wang (S'18–M'20) received the B.Eng. degree (with first class Hons.) and the Ph.D degree in information engineering from The Chinese University of Hong Kong, Hong Kong, in 2013 and 2018, respectively. From 2018 to 2020, he was an senior researcher with the Department of Risk Management, Tencent, Shenzhen, China. Since 2020, he has been with the College of Electronic and Information Engineering, Shenzhen University, Shenzhen, China, where he is currently an Assistant Professor. His research interests mainly involve machine learning and optimization theory in Multimedia Communications, Wireless Communications, and Image Processing.

Suzhi Bi (S'10–M'14–SM'19) received the B.Eng. degree in communications engineering from Zhejiang University in 2009, and the Ph.D. degree in information engineering from The Chinese University of Hong Kong in 2013. From 2013 to 2015, he was a post-doctoral research fellow with the Department of Electrical and Computer Engineering, National University of Singapore. Since 2015, he has been with the College of Electronics and Information Engineering, Shenzhen University, China, where he is currently an Associate Professor. His research interests mainly involve in the optimizations in wireless information and power transfer, mobile computing, and wireless sensing. He received 2019 IEEE ComSoc Asia-Pacific Outstanding Young Researcher Award, and Best Paper Awards of IEEE SmartGridComm 2013 and IEEE/CIC ICCC 2021. He is an Associate Editor of IEEE Wireless Communications Letters.

Ying-Jun Angela Zhang (S'00–M'05–SM'10–F'20) is a Professor at Department of Information Engineering, The Chinese University of Hong Kong. Her research interests are in optimization and learning in wireless communication systems and smart power grids. She is now an Associate Editor-in-Chief of IEEE Open Journals of the Communication Society and a Member of IEEE ComSoc Fellow Evaluation Committee. Previously, she has served as the Chair of the Executive Editorial Committee of IEEE Transactions on Wireless Communications and a Founding Chair of IEEE ComSoc Smart Grid Communications Technical Committee. She is the co-recipient of the 2011 Marconi Prize Paper Award in Wireless Communications, the 2013 SmartGridComm Best Paper Award, and the 2014 IEEE ComSoc Asia Pacific Board Outstanding Paper Award. She won the 2006 Hong Kong Young Scientist Award as the only winner from engineering science. She is a Fellow of IEEE and a Distinguished Lecturer of IEEE ComSoc.