Robust QoE-Driven DASH Over OFDMA Networks

Kefan Xiao, *Student Member, IEEE*, Shiwen Mao, *Fellow, IEEE*, and Jitendra K. Tugnait, *Fellow, IEEE*

***Abstract*—In this paper, the problem of effective and robust delivery of Dynamic Adaptive Streaming over HTTP (DASH) videos over an orthogonal frequency-division multiplexing access (OFDMA) network is studied. Motivated by a measurement study, we propose to explore the request interval and robust rate prediction for DASH over OFDMA. We first formulate an offline cross-layer optimization problem based on a novel quality of experience (QoE) model. Then the online reformulation is derived and proved to be asymptotically optimal. After analyzing the structure of the online problem, we propose a decomposition approach to obtain a user equipment (UE) rate adaptation problem and a BS resource allocation problem. We introduce stochastic model predictive control (SMPC) to achieve high robustness on video rate adaption and consider the request interval for more efficient resource allocation. Extensive simulations show that the proposed scheme can achieve a better QoE performance compared with other variations and a benchmark algorithm, which is mainly due to its lower rebuffering ratio and more stable bitrate choices.**

***Index Terms*—Dynamic Adaptive Streaming over HTTP (DASH), rate adaptation, resource allocation, Quality of Experience (QoE), Orthogonal Frequency Division Multiple Access (OFDMA).**

I. Introduction

RECENT years have witnessed the tremendous increase in mobile video traffic. Video has now dominated the mobile data traffic for over 60 percent in 2016, and is expected to account for over 75 percent in 2021 [2]. At the same time, the rapidly growth in both the overall mobile traffic (which has increased 18-fold since 2011) and the number of mobile devices (429 millions were added in 2016) have made mobile video streaming a great challenge. In addition, the instability nature of wireless links makes the situation even worse. There is a compelling need to achieve high efficiency and robustness of video delivery over wireless networks, while guaranteeing users' Quality of Experience (QoE). This problem should be studied from both wireless infrastructure and user aspects.

Video streaming has drawn great attention for decades. The early works are mainly based on the User Datagram Protocol (UDP), which can provide timely transmission compared with Transportation Control Protocol (TCP) that is designed for reliability rather than timeliness. However, the deployment of UDP based algorithms is challenging due to the incompatibility with firewalls and other types of middleboxes. On the other hand, TCP is supported by most middleboxes for its reliability and security. The congestion control algorithms facilitate timely transmission as well. Therefore, Hypertext Transfer Protocol (HTTP) based video streaming is now the mainstream technique. In particular, Dynamic video streaming over HTTP (DASH), which can adapt to the variation of network conditions, is recognized as a promising technique to enhance the user QoE. Many commercial video services, e.g., YouTube and Netflix, are based on DASH, which have accounted for more than half of the total Internet traffic in North America in 2016 [3].

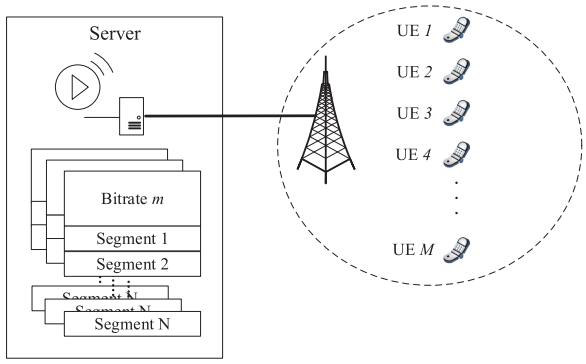


Fig. 1. DASH mobile video streaming system architecture.

Among different generations of mobile transmission techniques, the 3GPP-Long Term Evolution (LTE) and Wi-Fi (802.11 standards) are the two most popular. The Orthogonal frequency-division multiplexing (OFDM) is the common method of both standards that encode and transmit digital data on multiple subcarriers of a broadband channel. OFDM is expected to continue serving the next generation wireless networks for its ability on tackling narrowband interference and frequency-selective fading with a low complexity. OFDM also introduces great flexibility by allowing dynamically assigning subcarriers, time, and power to each user in order to accommodate customized QoS requirements, such as transmission rate [4], [5] or power efficiency [6].

We consider the scenario that a video user equipment (UE) receives a DASH video through an OFDM network, as shown in Fig. 1. The two techniques are in different layers in the protocol stack and have different design goals. However, for video streaming, both of them should be designed for the ultimate goal of guaranteeing user QoE. In this paper, we study the cross-layer,

Manuscript received December 22, 2018; revised June 5, 2019; accepted July 11, 2019. Date of publication July 31, 2019; date of current version January 24, 2020. This work was supported in part by the US NSF under Grants ECCS-1923717 and CNS-1822055, and in part by the Wireless Engineering Research and Education Center (WEREC) at Auburn University. This work was presented at IEEE GLOBECOM, Washington, DC, USA, December 2016. The Associate Editor coordinating the review of this manuscript and approving it for publication was Dr. S. Mehrotra. *(Corresponding author: Shiwen Mao.)*

The authors are with the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36839-5201 USA (e-mail: kzx0002@auburn.edu; smao@ieee.org; jktugnait@auburn.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2019.2929929

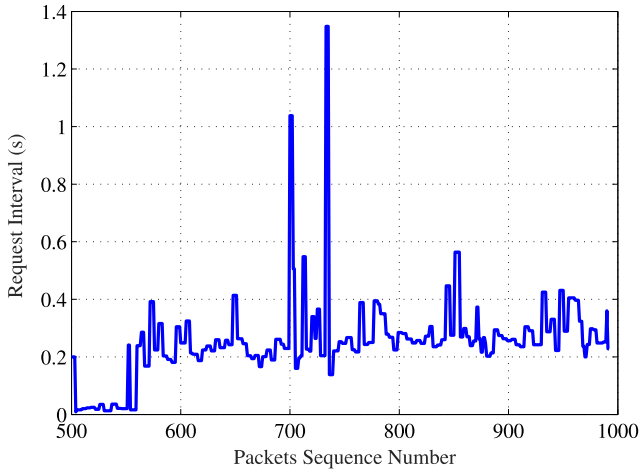


Fig. 2. The variations of request intervals obtained in our measurement study over an LTE link.

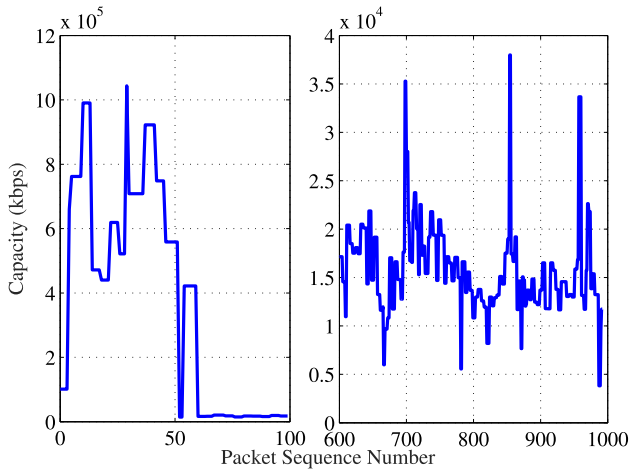


Fig. 3. The variations in the average capacity of an LTE video link.

joint design of DASH and OFDM. In DASH, the video consumers take control of choosing different data rates for future video segments, which are coded (or stored) at the video server. The corresponding information is sent to the consumers at the beginning of video transmission. Resource allocation in OFDMA networks, on the other hand, is executed at the cellular base station, which takes into account various factors such as channel state information (CSI), power budget, and user QoE requirement and determines the optimal resource allocation for multiple video UEs.

A. Motivation

To provide useful insights into the problem of DASH based video streaming over OFDMA networks, we conducted a measurement study using dash.js [7] in Broun Hall, Auburn University, Auburn, AL, USA over Version LTE. The video source is placed on a remote server and coded into 10 different bit rates ranging from 254 kbps to 14931 kbps. We collected the video playback trace information over LTE in 1 hour daytime and 1 hour night time. Define *request interval* as the period between when the request for the next segment is sent and when the first byte of the requested video segment arrives at the user. In this experiment, we measured the request interval of the DASH video session, as well as the link capacity that can be acquired by measuring the download speed of each video segment.

As shown in Fig. 2, the request interval could be as large as 1.3 s, while 80 percent of the intervals are in the range from 0.2 to 0.4 s. However, most of the joint design of DASH and scheduling algorithms do not fully consider this relatively large period of time and still allocate resources to the video users, even though not much transmissions are needed during this interval. Motivated by this observation, we propose a new formulation of the resource allocation problem at the BS by exploiting the request intervals (see Section III).

In addition, we measured the capacity of the LTE link during playback. Usually the capacity will be influenced by factors such as fading and shadowing, which will all trigger the capacity variation at small timescales. In DASH, many existing rate adaption schemes are based on the average downloading capacity over one segment, which is assumed to be slowly varying and relatively stable. In Fig. 3, we present the average capacity of one segment, which exhibits quite large variations over time. Such large variations could make the existing rate adaptation schemes ineffective. Motivated by this observation, we propose a model predictive control based optimization of the rate adaption at the video user side to achieve robustness (also see Section III).

B. Contributions and Organization

We address the cross-layer QoE-driven optimization problem of DASH over OFDMA networks in this paper. The contribution of this work is three-fold:

1) We develop a new QoE model and formulate an *offline* optimization problem jointly considering the new QoE model as well as the specific resource allocation model in OFDMA networks.
2) We then derive an *online* optimization problem to approximate the offline problem. We analyze the online problem and decompose it into a BS resource allocation problem and an UE rate adaptation problem, for each of which effective solution algorithms are developed. More important, we also prove the asymptotic optimality of the online algorithm.
3) Extensive simulations are conducted to validate the performance of the proposed algorithms. The results show that the proposed algorithms can achieve a 40% less rebuffering ratio than the state-of-art MPC based algorithm.

The remainder of this paper is organized as follow. The System model is presented in Section II. In Section III, we first formulated a globe optimization, offline problem and then transform it to an online optimization problem. The analysis and problem decomposition are presented in Section IV. In Section IV, we further explicitly take the request interval of the users into consideration for more efficient resource allocation. The rate-adaption process is analyzed and then a robust scheme is proposed. Our simulation study and results are presented in Section V. The related works are discussed in Section VI and we conclude this paper in Section VII.

II. System Model

A. Network Model

We consider a wireless video streaming network as shown in Fig. 1, consisting of video servers, the wireline network, a cellular base station (BS), and multiple UEs. The videos are stored in the remote servers. To reduce the delay of transmission, more and more servers are deployed closer to end UEs. Generally, the capacity bottleneck in the end-to-end transmission path is the last hop, where a BS serves multiple UEs. The transmission in the Internet before the last hop could be modeled as a *request interval time* since the network conditions such as capacity and congestion level mainly influence the propagation time before the BS.

We consider an OFDMA based cellular system with one BS serving N video streaming UEs, which are denoted as $\mathbb{N} = \{1, 2, ..., N\}$. We assume the total available bandwidth is W Hz, which consists of M subcarriers denoted as $\mathbb{M} = \{1, 2, ..., M\}$. Each subcarrier $m \in \mathbb{M}$ is assumed to have an equal bandwidth of $\kappa = W/M$ Hz. Since the bandwidth of each subcarrier is sufficiently small, they only experience flat fading. For a resource block in OFDM, the time length of each resource allocation is denoted as τ, the time step is denoted as $t = 1, 2, ...$, and each time slot contains an integer number of OFDM symbols and is an integer multiple of τ.

Denote $g_{im}(t)$ as the channel gain of UE $i \in \mathbb{N}$ on subcarrier $m \in \mathbb{M}$ at time t, and $p_{im}(t)$ accounts for the total transmission power assigned to subcarrier m for UE i at time t. Assume the total power can be assigned is P. Since multiple users can share subcarriers in each time slot, denote the non-overlapping time fraction of user i on subcarrier m in time slot t as $0 \le v_{im}(t) \le 1$. Without loss of generality, assume the duration of a time slot is unit time. We have

$$\sum_{i=1}^{N} v_{im}(t) \le 1, \quad \forall m, t. \tag{1}$$

The additive white Gaussian noise (AWGN) at the receiver has unit spectrum density. According to Shannon formula, user i's maximum rate on subcarrier m at time t is

$$r_{im}(t) = \begin{cases} v_{im}(t)\kappa \log_2\left(1 + \frac{p_{im}(t) g_{im}^2(t)}{v_{im}(t)\kappa}\right), & \text{if } v_{im}(t) > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

The BS will first estimate the CSI on each subcarrier. Then with different targets, e.g., maximizing the total data rate or maximizing the energy efficiency, the BS assigns transmit power $p_{im}(t)$ and time fraction $v_{im}(t)$ to each UE i on each subcarrier m at each time t. Naturally, the total data rate for UE i at time t is the summation of all the data rate $r_{im}(t)$ on each subcarrier, denoted as $r_i(t) = \sum_{m=1}^{M} r_{im}(t)$.

B. Streaming Video Model

The video for UE i is S_i, which has been partitioned into K_i consecutive segments, each coded with different bit rates. Assume each segment of all videos is l s. Without lost of generosity, assume all the coded bit rates R_i are in the same set $\mathbb{R}$. Naturally, the size of segment k_i in bits can be represented as $R_i[k_i] \times l$.

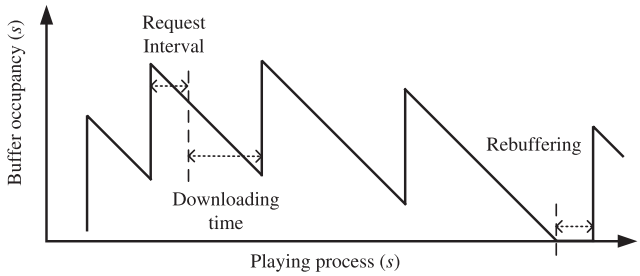


Fig. 4. Timeline of the discrete time DASH system.

During the video session, the playback buffer dynamic process at a user is illustrated in Fig. 4. When a new segment is completely downloaded, the buffer occupancy (in seconds) is immediately increased. The buffer occupancy decreases linearly with time when the segment is played out. When the buffer is empty, the playback stalls. The time till the next segment arrives is the *rebuffering time* (the interval before the 5th segment in Fig. 4). After downloading a segment, the UE estimates the average rate of the segment and sends a request back to the video server for the next segment. The period since the request is sent till the first byte of the next segment arrives at the UE, is the *request interval* (see the 2nd segment in Fig. 4). Depends on the computational complexity of the UE algorithm and the server it chooses, the request interval could be as large as several seconds, as shown in Section I.

Denote $B_i[k_i] \in [0, B_{i,\max}]$ as the playback buffer occupancy at UE i, which represents the amount of video data stored in the buffer after downloading segment k_i, as measured in playback time. Users may have different buffer sizes $B_{i,\max}$, which represents the total amount of storage measured in playback time. The UE can play each video segment only after it is fully downloaded, since the segment itself contains the playback metadata.

Let $s_i[k_i]$ represent the time slot when segment k_i is downloaded at UE i. During playback, the UE calculates the average rate $\bar{C}_i[k_i]$ of downloading segment k_i, as an important indicator of future capacity in most existing algorithms. The request interval is denoted by $\pi_i[k_i]$, which depends on the network parameters and network congestion level. During this period, this UE does not need any resource from the BS for transmission. Since this period could be considerably large, it should be considered in the resource allocation scheme.

The playback buffer process can be modeled as follows. The timeline of the operations and updates is presented in Fig. 4.

$$\begin{cases} B_i[k_i] = \min\left\{\max\left\{0, B_i[k_i - 1] - \pi_i[k_i] - \frac{R_i[k_i]l}{\bar{C}_i[k_i]}\right\} + l, B_{i,\max}\right\} \\ \bar{C}_i[k_i] = \frac{\sum_{t=s_i[k_i-1]+\pi_i[k_i]}^{s_i[k_i]} r_i(t)}{s_i[k_i] - s_i[k_i - 1]} \\ s_i[k_i] = s_i[k_i - 1] + \pi_i[k_i] + \frac{R_i[k_i]l}{\bar{C}_i[k_i]}. \end{cases} \tag{3}$$

In (3), the first equation describes the bounded behavior of buffer occupancy (i.e., finite playout buffer size and nonnegativity).

The second equation describes how to calculate the average capacity, while the third equation describes how $s_i[k_i]$ evolves over time, as illustrated in Fig. 4. The rebuffering time for one segment downloading event is $\max\{0, \pi_i[k_i] + \frac{R_i[k_i]l}{\bar{C}_i[k_i]} - B_i[k_i - 1]\}$. Rebuffering event happens when this value is larger than 0, which significantly degrades the user QoE [8].

C. Quality of Experience (QoE) Model

The QoE is an important indicator of the performance of video communication systems. How to design a QoE model to capture the user's viewing experience has been extensively studied. In the past, the QoE is mainly divided into two parts: objective measures and subjective evaluation. Objective measures, e.g., Peak Signal-to-Noise Ratio (PSNR), represent the quality of video frames, while subjective evaluation, such as Mean Opinion Score (MOS) [9], reflects user's assessment of the viewing experience [10].

In [11], the authors study how the factors, such as playback buffer and average bit rate, influence the user engagement. It is suggested that new QoE models should be developed that consider both video quality and user experience, which could be represented coarsely as user engagement. The authors in [8] suggest that the main factors of a QoE model should include buffering ratio, video starting time, average bit rate, and attributes such as type of video, Internet service provider (ISP), region, etc. In [12], the QoE model is defined as a weighted sum of several main factors. The weights could be adjusted for different scenarios. Recent studies such as [11], [13], [14] suggest that the video quality and its variation have strong correlation with the MOS. In this paper, we aim to be as general as possible to model user QoE as a weighted sum of several important factors. Each of the weighted factors should reflect one attribute that has a specific physical meaning, and can influence the experience of users. The weights, on the other hand, accounts for the importance of each factor in the overall QoE. Research has been conducted on finding the best weights to better approximate the QoE of users, and the state-of-art results could be applied directly here.

Intuitively, the larger the bit rate, the better the video quality. The relationship between video quality and bit rate could be modeled as [14]

$$q(t) = f(R[t]) = a \cdot \log(R[t]) + b. \tag{4}$$

The coefficients a, b could be adjusted according to specific video type and playback device for stored videos. On the other hand, the variance of video quality across segments influences the user experience as well. The tradeoff between the average quality and variance is a key factor to be taken into consideration.

The key factors of our QoE model is listed below for UE i along with the explanation of their physical meanings.

- *Average video quality* $m_i[K]$*:* it represents the video quality level averaged over the entire video playback period [12], defined as: $m_i[K] = \frac{1}{K}\sum_{k=1}^{K} q(R_i[k])$.
- *Variance of video quality* $\text{Var}_i[K]$*:* it accounts for the quality variation from segment to segment, given by $\text{Var}_i[K] = \frac{1}{K}\sum_{k=1}^{K}(q(R_i[k]) - m_i[K])^2$. As the average video quality, the variance also has a considerable impact on the QoE [12].
- *Rebuffering ratio* $\text{Reb}_i[K]$*:* rebuffering occurs when there is underflow at the playback buffer. The rebuffering ratio is defined as the total rebuffering time over the total video duration $L_i = K_i \times l$, as

$$\text{Reb}_i = \frac{1}{L_i}\sum_{k=1}^{K} \max\left\{0, \pi_i[k_i] + \frac{R_i[k]l}{C_i[k]} - B_i[k-1]\right\}. \tag{5}$$

  This factor affects the QoE even more significantly than variance [15].
- *Startup delay* T_i^s*:* it represents the time between user requests a video and the playback begins. Normally, a certain length of buffer occupancy need to be accumulated before playback starts [12]. It depends on the transmission rate and how the video is encoded.

As different user might focus on different factors, we use a weighted sum to ensure flexibility of this model. This weighted form follows the linear model practice when conducting the regression analysis of how multiple relevant factors influence the subject metric. Each term has a clear physical meaning, and the weights tell the story of how and at what scale each term influences the overall QoE. For segment k_i, user i's QoE is defined as

$$\begin{aligned} Q[k_i] = \; & q(R_i[k_i]) + \theta \cdot (q(R_i[k_i]) - m_i[K_i])^2 \\ & + \frac{\lambda}{K_i l} \max\left\{0, \pi_i[k_i] + \frac{R_i[k_i]l}{\bar{C}_i[k_i]} - B_i[k_i - 1]\right\}. \end{aligned} \tag{6}$$

in which $\theta < 0$ and $\lambda < 0$ are tunable weights. The reason we divide the rebuffering time by the total length of the video is to normalize the rebuffering time, since the same rebuffering time should have different impacts on user QoE when the video lengths are different. For the entire playback process, the QoE of user i can be defined as

$$Q_i = \sum_{k_i=1}^{K_i} Q[k_i] + \eta \cdot T_i^s, \tag{7}$$

where $\eta < 0$ is a tunable weight, and T_i^s is the startup delay. This model can be flexibly tuned for different users and application scenarios with different parameter sets $\{\theta, \lambda, \eta\}$. Furthermore, if the viewing process is emphasized instead of the startup phase, the overall QoE can be defined as $Q_i = \sum_{k_i=1}^{K_i} Q[k_i]$, i.e., removing the startup delay term.

To consider fairness among users, we adopt the concave utility function $U_\alpha(\cdot)$ defined as [16]

$$U_\alpha(x) = \begin{cases} \log(x), & \text{if } \alpha = 1 \\ \frac{x^{1-\alpha}}{1-\alpha}, & \text{otherwise.} \end{cases} \tag{8}$$

For instance, when $\alpha = 1$, \maximizing the utility sum ensures proportional fairness. We define the fairness QoE as $Q^{fair}(k_i) = U_\alpha\left(Q(k_i)\right)$ and the QoE through the playing process $Q_i = \sum_{k_i=1}^{K_i} Q^{fair}(k_i)$.

III. Problem Formulation

To achieve the goal of QoE maximization, the challenge is that the BS and video UEs are operating at different timescales. At each time t, the BS adapts to the variation of network states, e.g., updated CSI. On the other hand, each video user adjusts the bit rate of the next video segment after downloading the present one. In this section, we first formulate an offline problem and then provide an online transformation with reduced complexity. The decomposition of the online problem is then presented and the decomposed problems will be solved in Section IV.

A. The Offline Optimization Problem

The offline optimization problem is formulated based on the assumption that we know all the information about the video processes. The optimization variables related to the BS are wireless resources: $v_{im}(t)$ and $p_{im}(t)$ for UE i on subcarrier m. Video rate adaption is executed each time when the UE finishes downloading a segment. The data-rate $R_i[k_i]$ can only be chosen from a given set $\mathbb{R}_i$ defined by the video encoder (or, server). The *offline* QoE maximization problem, denoted by **Prob-Offline**, is formulated as follows.

$$\max_{\{p_{im}(t),v_{im}(t),R_i[k_i]\}} \quad \Lambda = \sum_{i=1}^{N}\sum_{k_i=1}^{K_i} Q^{fair}[k_i] \tag{9}$$

$$\text{s.t.:} \quad \sum_{i=1}^{N} v_{im}(t) \le 1,\ \forall m,t \tag{10}$$

$$\sum_{i=1}^{N}\sum_{m=1}^{M} p_{im}(t) \le P,\ \forall t \tag{11}$$

$$R_i[k_i] \in \mathbb{R}_i,\ \forall i \tag{12}$$

$$B_i[k_i] \in [0, B_{i,\max}],\ \forall i, k_i. \tag{13}$$

In terms of $p_{im}(t)$ and $v_{im}(t)$, problem Prob-Offline is convex. For rate adaption, it is an integer programming problem with a limited solution set (for example, most Youtube videos offer 3 to 6 resolutions). The optimal value could be solved with dynamic programming (DP).

To deal with variables $R_i[k_i]$, we relax the constraint to allow them to take any values (rather then chosen from a give set $\mathbb{R}_i$), to obtain an approximation problem. Notice that the video rate and quality can be mapped with function $q_i(\cdot)$. Therefore, the quality q_i also belongs to a set of scalar values Θ_i (as $R_i[k_i]$ is chosen from $\mathbb{R}_i$). Furthermore, we can select the optimal quality and then derive the corresponding rate using $q^{-1}(\cdot)$ and round it to the closest rate in set $\mathbb{R}_i$. Here, we abuse the notation of q_i, but the reader can easily differentiate the $q(\cdot)$ and q_i as the quality function and the scalar value of quality, respectively. As $q_i(\cdot)$ is monotonically increasing, we have $q_i \in [q_{i,\min}, q_{i,\max}]$ in which $q_{i,\min}$ and $q_{i,\max}$ can be calculated by substituting the minimum and maximum rates into $q_i(\cdot)$, respectively. The mapping is unique. The approximation problem can be written as

$$\max_{\{p_{im}(t),v_{im}(t),q_i[k_i]\}} \quad \Lambda^{\text{appr}} = \sum_{i=1}^{N}\sum_{k_i=1}^{K_i} U_\alpha\Bigg(q_i[k_i] + \theta\cdot(q_i[k_i]-m_i[K_i])^2 + \frac{\lambda}{K_i l}\max\left\{0, \pi_i[k_i] + \frac{q^{-1}(q_i[k_i])l}{\bar{C}_i[k_i]} - B_i[k_i-1]\right\}\Bigg) \tag{14}$$

$$\text{s.t.:}\ \ q_{i,\min} \le q_i[k_i] \le q_{i,\max} (10)\sim(13). \tag{15}$$

Theorem 1: The approximation problem defined in (14) is a convex optimization problem.

The proof is given in Appendix A. With Theorem 1, we can solve this problem with the Karush-Kuhn-Tucker (KKT) conditions to acquire the optimal quality and then round the rate down to the closest feasible rate. However, solving this problem is based on the knowledge of the entire (and future) playback process and network states), e.g., the mean video quality $m_i[K_i]$ over the entire playback process K_i. It is an offline problem which may not be practical in many cases.

B. Online Optimization Formulation

Following Prob-Offline, an online version **Prob-Online** is derived in this section. Notice that the only term that involves future information is the overall mean quality $m_i[K_i]$. We first propose the formation of Prob-Online with an approximation to this term and then prove the asymptotic convergence property of Prob-Online to Prob-Offline.

For a single user, we define the online QoE as follows.

$$Q_i^{\text{online}}[k_i] = U_\alpha(q(R_i[k_i]) + \theta\cdot(q(R_i[k_i]) - \hat{m}_i[k_i-1])^2 + \lambda\cdot \text{Reb}_i[k_i]). \tag{16}$$

in which $\hat{m}_i[k_i]$ is updated as

$$\hat{m}_i[k_i] = \hat{m}_i[k_i-1] + \frac{\zeta_i}{k_i+\zeta_i}(q(R_i^*[k_i]) - \hat{m}_i[k_i-1]), \tag{17}$$

where $q(R_i^*[k_i])$ is the video quality corresponding to the optimal rate $R_i^*[k_i]$, and ζ_i is a tunable parameter for different users. This way, we rewrite the problem from over the entire time window $\mathbb{T}$ to the problem that can be solved at each time slot using past and present information.

Lemma 1: The $\hat{m}_i[k]$ in (17) approximates the average of the optimal quality with k goes to infinity.

The proof is given in Appendix B. Lemma 1 is on the convergence property of $\hat{m}_i[k]$, which we use to replace the term of overall average quality $m_i[K_i]$. It lays the foundation for the proof of the convergence of the online algorithm.

Lemma 2: Prob-Online is a convex optimization problem.

The proof for lemma 2 is similar to theorem 1 and is omitted for brevity. Based on Lemmas 1 and 2, we can take a step further to claim that the solution to the online problem converges to the offline problem solution asymptotically.

Theorem 2: The solution of Prob-Online asymptotically converges to the solution of Prob-Offline.

The proof is given in Appendix C. With Theorem 2, we can derive the solution for Prob-Online with past and present information. More important, the solution converges to the offline optimal solution with the increase of time.

With the lemmas and theorems, we can solve the online problem to acquire the optimal solution $\{p^*_{im}(t), v^*_{im}(t), R^*_i[k_i]\}$ for each time slot. However, the user rate adaption may not be synchronized, and more important, the user rate adaptation is executed at a different timescale from BS resource allocation. In addition, the number of video UEs that are actually transmitting packets is varying over time (i.e., due to the request interval). To address all these challenges, we further decompose the optimization problem into two sub-problems: a BS resource allocation problem and a video rate adaptation problem, which are solved in the next section.

IV. Solution Algorithms and Analysis

A. Prob-Online Analysis

We first rewrite the online formulation by adding the equality constraints on the capacity of each user.

$$\max_{\{p_{im}(t),v_{im}(t),R_i[k_i]\}} \Lambda^{\text{online}} = \sum_{i=1}^{N} Q_i^{\text{online}}$$
$$C_i(t) = \sum_{m=1}^{M} r_{im}(t), \forall i \in \mathbb{N}, \forall t \in \mathbb{T}$$
$$(10) \sim (13). \tag{18}$$

In the above formulation, the BS operation, i.e., resource allocation, is coupled with UEs' rate adaptation in the average capacity term $\bar{C}_i[k_i]$. At each time slot $s_i[k_i]$, the controller will try to optimize the QoE by estimating the capacity for the next segment $\bar{C}_i[k_i+1]$. On the other hand, the BS controller will also optimize the QoE by allocating resource to UEs at each time slot t, which updates $C_i(t)$. With asynchronous operations of the two parties, it is natural to decompose the problem to BS and UE subproblems, to decouple their operations.

At the UE side, rate adaption is executed based on the estimation of $\bar{C}_i[k_i+1]$ at time $s_i[k_i]$ after downloading segment k_i. Most estimation algorithms are based on the historical data of capacity. The value of $\bar{C}_i[k_i+1]$ is influenced by the resource allocation before time k_i. On the other hand, the BS allocates resources to each UE at a smaller timescale than rate adaption. Therefore, at each resource allocation time step, the video rate has already been chosen and remains fixed for a period of time. Based on this observation, we present a *primal decomposition* approach as given in [17].

B. UE Rate Adaption

Rate adaption at the UE is executed when the previous segment is downloaded. The assigned transmission rate to the UE during the downloading process is determined by the environment conditions sensed by the BS. To choose the rate for the next segment, the capacity is mainly an estimation based on the previous segment. It makes sense to optimize the QoE at each UE by rate adaption with a given capacity estimation $\bar{C}_i[k_i+1]$.

At $s_i[k_i-1]$, UE i aims to maximize its QoE by choosing $R_i[k_i]$, based on $B_i[k_i-1]$ and historical information. We formulate the UE i side rate adaptation problem as

$$\max_{R_i[k_i]} \; I_i^U(k_i) \cdot \hat{Q}_i^{\text{online}}[k_i] \tag{19}$$
$$\text{s.t.:} \; R_i[k_i] \in \mathbb{R}_i \tag{20}$$
$$B_i[k] \in [0, B_{i,\max}], \forall k \in [s_i[k_i-1], s_i[k_i]], \tag{21}$$

where $\hat{Q}_i^{\text{online}}[k_i]$ is defined as in (16) with estimated $\bar{C}_i[k_i]$. The indication function $I_i^U(k_i)$ is equal to 1 when the buffer is not full and 0 otherwise. By the time when the buffer is full, the video UE would pause `Get` for a few seconds depending on the mechanism. Rebuffering time is updated as in (3).

The average download capacity is estimated based on historical data. From the application layer, the UE can only acquire an estimation of the average capacity by dividing $R_i[k_i]l$ with the time difference between sending request and completely downloading the segment. With information from the TCP layer, we can have a better estimation by narrowing down the downloading period between the time of receiving the first packet and the last packet. However, it is still hard to accurately predict the future downloading capacity due to network dynamics in the future. We can acquire an estimation by simply using the exponential average over the previous capacity. Then solve the optimization by brute-force search in the rate space $\mathbb{R}_i$, which is small (e.g., 3 to 6 different resolutions/rates). In this paper, we call this simple scheme as **DORA** (*D*ASH over *O*FDMA Networks with *R*ate *A*daptation) [1].

Inspired by [18], which measures the cellular network capacity and proposes an estimation model based on Poisson process, we also consider the downloading capacity as a random variable. Furthermore, by optimization over several future steps with estimated future information, we can increase the robustness of rate adaption based on the theory of stochastic model predictive control (SMPC). This way, the optimization does not rely on accurate estimations of future information that are hard to acquire.

Assume that the average capacity $\bar{C}_i[k_i]$ is a random process based on distribution $\Xi(\cdot)$ with mean $\bar{C}_i[k_i-1]$. At each rate adaptation time $s_i[k_i-1]$, we will first update the average capacity $\bar{C}_i[k_i-1]$ with the measurement of this segment transmission. The exponential window moving average (EWMA) method is used to update the mean as

$$\tilde{C}_i[k_i-1] = \xi \tilde{C}_i[k_i-2] + (1-\xi)\bar{C}_i[k_i-1], \tag{22}$$

with exponential weights $\xi \in (0,1)$. Suppose the prediction horizon here is z. The z predictions over the prediction horizon are drawn with Monte Carlo sampling based on the distribution. By solving the optimization problem over the z steps ahead, we can obtain control moves as follows.

$$\vec{R}_i := \{R_i[k_i], R_i[k_i+1], ..., R_i[k_i+z-1]\}. \tag{23}$$

The first control move, $R_i[k_i]$, will be taken as the choice for the next segment. In this paper, we adopt dynamic programming

Algorithm 1: UE Video Rate Adaption Algorithm

Initialize the mean of $\Xi(\cdot)$ based on historical data;
for $k_i = 1, 2, ..., K_i$ **do**
  Calculate the average capacity $\bar{C}_i[k_i - 1]$ based on the downloading process of $k_i - 1$;
  Update the mean of $\Xi(\cdot)$ with EWMA;
  Generate z steps of estimation values $\hat{C}_i[k_i + j]$, $j = 0$, 1, ..., $z - 1$, based on the distribution $\Xi(\cdot)$;
  Initialize a vector to store the QoE maximization result for each bit rate, $\vec{Q}_{|\mathbb{R}_i|\times 1} = \vec{0}$, for the DP;
  for $j = 0, 1, ..., z - 1$ **do**
    Use the DP to update $\vec{Q}[j]$;
  end
  Derive the optimal series of choices $\vec{R}_i$ from $\vec{Q}[z - 1]$. Apply the first bit rate choice for video segment k_i;
end

(DP) to compute the overall control moves $\vec{R}_i$. The rough idea is to store the intermediate optimal QoE score of each prediction step and then update it at each step. Due to limited size of the bitrate set $\mathbb{R}_i$ and small prediction horizon, the time complexity of the DP algorithm, $O(|\mathbb{R}_i| \cdot z)$, is acceptable in this context. By explicitly considering the uncertainty of estimations, this algorithm provides us a robust rate adaption solution. The new optimization problem is formulated as

$$\max_{R_i[k_i],...,R_i[k_i+z-1]\}} \quad \sum_{j=0}^{z-1} \hat{Q}_i^{\text{online}}[k_i + j] \tag{24}$$

$$\text{s.t.: } R_i[k_j] \in \mathbb{R}_i, j \in \{k_i, ..., k_i + z - 1\} \tag{25}$$

$$B_i[k_i] \in [0, B_{i,\max}], \forall k_i, \tag{26}$$

where $\hat{m}_i[k_i + j]$ and $\hat{B}_i[k_i + j]$ are updated as in (17) and (5), respectively. The procedure is presented in Algorithm 1.

The time complexity the algorithm is $O(|\mathbb{R}_i|z)$. Theoretically, a longer z means looking further into the future, of which the states are usually less accurately estimated. Thus z should not be too large. The complexity is linear with respect to $|\mathbb{R}_i|$ or z. Users can tradeoff between performance and execution time by tuning the parameter z.

C. Base Station Side Optimization

Compared to the UE rate adaption algorithm, the optimized resource allocation at the BS is a master problem that decides the capacity of each UE i at given conditions and chosen data rate $R_i[k_i]$. The BS will allocate resources to the UEs at each time t, which is much smaller than the timescale of rate adaption (e.g., microseconds vs. seconds). Moreover, for each time t, the data rate of each UE has been selected already. Therefore, the first two terms in (16) are constant from the BS perspective. Since $\lambda < 0$, we only need to minimize the rebuffering time. The problem is now transformed to minimizing the total rebuffering time of all UEs by resource allocation at the BS. The object function is given by

$$\text{Reb}_{BS}(t) = \sum_{i=1}^{N} \max \{t - s_i[k_i - 1] - B_i[k_i - 1] - I_{BS,i}(t) \cdot l, 0\}, \text{ for } t \in (s_i[k_i - 1], s_i[k_i]], \tag{27}$$

where function $I_{BS,i}(t)$ indicates whether segment k_i has been fully downloaded, defined as

$$I_{BS,i}(t) = \begin{cases} 1, & \text{if } \sum_{\mu=k_i-1}^{t} \sum_{m=1}^{M} r_{im}(\mu) \geq R_i[k_i]l \\ 0, & \text{if } \sum_{\mu=k_i-1}^{t} \sum_{m=1}^{M} r_{im}(\mu) < R_i[k_i]l. \end{cases} \tag{28}$$

In (27), $t - s_i[k_i - 1]$ is the buffer occupancy consumed since the last time a segment is downloaded, while the third term accounts for the buffer occupancy at that time. If we can guarantee the average capacity $\bar{C}[k_i]$ be larger than the ratio $\frac{R_i[k_i]l}{B[k_i-1]}$, the target function will be minimized. This condition for each UE is hard to achieve due to limited resources at the BS. We can reformulate the BS optimization problem as

$$\max_{\{p_{im}(t), v_{im}(t)\}} \Phi(p_{im}(t), v_{im}(t)) \tag{29}$$

$$\text{s. t.: } \sum_{i=1}^{N} v_{im}(t) \leq 1, \ \forall m, t \tag{30}$$

$$\sum_{i=1}^{N} \sum_{m=1}^{M} p_{im}(t) \leq P, \ \forall t, \tag{31}$$

where

$$\Phi(p_{im}(t), v_{im}(t)) = \sum_{i=1}^{N} \left\{ \frac{I_i(t)}{\rho_i(t) + \sigma} \sum_{m=1}^{M} r_{im}(t) \right\},$$

$I_i(t)$ is an indication function on whether user i is in a request interval (and thus no resource is needed), and σ is a small scalar constant that prevents the denominator to be zero. In problem (29), $\rho_i(t)$ is the user i buffer occupancy state maintained at the BS. Each time a user requests the next video segment, it also updates its buffer level to the BS. Then the BS sets $\rho_i(t)$ to the reported buffer occupancy, i.e., $\rho_i(t) = B_i[t]$. Over the next time slots, $\rho_i[t]$ evolves as follows to emulate the playback process at user i.

$$\rho_i(t) = \begin{cases} \max\{0, \rho_i(t-1) - 1\}, & s_i[k_i - 1] \leq t < s_i[k_i] \\ \max\{0, \rho_i(t-1) - 1\} + l, & t = s_i[k_i], \end{cases} \quad k_i = 1, 2, ..., K_i, \tag{32}$$

until the the next user i feeback is received. This way, the BS maintains the buffer state of each UE at a minimum control overhead. Problem (29) is to maximize a weighted sum of the downlink rates of all UEs at each time t, while each weight is inversely proportional to the playback buffer occupancy at the corresponding UE.

Theorem 3: The problem defined in (29) is convex.

With Theorem 3, we can solve the problem with a convex optimization solver. Since we assume dynamic link states, the time complexity is $O(|\mathbb{N}| \cdot |\mathbb{M}|^2)$, where $|\mathbb{N}|$ is the number of UEs and $|\mathbb{M}|$ is the number of subcarriers. Both $|\mathbb{N}|$ and $|\mathbb{M}|$ are usually not very large. In addition, the algorithm is executed at the BS, which by assumption should have much more computation power than the UE. Furthermore, some techniques such as early stopping could be applied to achieve a trade-off between performance and execution time.

The algorithm considers both robust rate adaption and *r*equest *i*nterval is termed **DORA-RI**.

V. Simulation Study

A. Simulation Scenario and Algorithm Configuration

In this section, the performance of the proposed algorithms is evaluated with Python multi-threading simulation. For the physical layer, the total bandwidth is $W = 5$ MHz and consists of $M = 32$ subcarriers, each with $\kappa = 160$ KHz. The energy constraint of the BS is $P = 10$ W. We generate the channel gain over unit noise energy with the Rayleigh channel model for which the expectation is 5 dB. The time frame for resource allocation is $\tau = 5$ ms. We assume the BS acquires CSI with contamination of -60 dB W/Hz. Since the request interval mainly depends on the network conditions, e.g., the congestion level, it could be as large as 10 s. In this paper, we simulate the request interval as a uniformly distributed process in the range of 50 ms to 500 ms. The fairness function is chosen as the natural logarithm $U_\alpha(\cdot) = \ln(\cdot)$.

Each video lasts for 2 minutes and is partitioned into 1 s segments and coded into five levels of rates, i.e., $\mathbb{R} =$ [100 kbps, 300 kbps, 500 kbps, 900 kbps, 1500 kbps, 2000 kbps], which is consistent with the 240p, 360p, 480p, 720p,[1] and 1080p formats for general genres [19]. For the QoE model, we use constant α_i and β_i for all users. The quality model α_i and β_i are fitted with the data from [15] and assumed to be the same for all users. We set $\theta = 0.2$, $\lambda = 2.5$, and $\eta = 20$ according to the guidelines in [12]. The distribution of the download capacity is set according to a normal distribution with mean $\hat{C}_i[k_i]$ and a variance that is 10% of the mean. The prediction horizon z is set to 7.

In order to evaluate the general performance and improvement over the state-of-art techniques, the proposed algorithm DORA-RI is first compared with the original DORA proposed in [1] and a proportionally fair network resource allocation with water-filling algorithm (PFWF-RM) scheme [16], [20]. The purpose is to demonstrate the achievable improvement over these two baseline schemes.

In addition, we would like to gain insights into the the proposed algorithm. Each component of DORA-RI are isolated out to test their impact. Specifically, the proposed scheme DORA-RI is compared with two variants:

1) DORA-MPC, which uses a deterministic model predictive control (MPC) algorithm for bit rate adaption. All the other parts of this scheme are the same as DORA-RI.

[1]Note that 900 kbps and 1500 kbps are the rates for 720p.

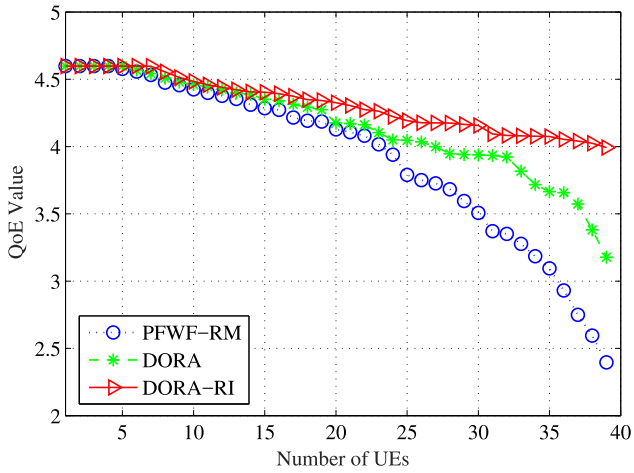


Fig. 5. Comparison of QoE values for DORA-RI, DORA, and PFWF-RM.

2) DORA-no-RI, which only considers the request interval at the BS but does not adopt SMPC. All the other parts are the same as DORA-RI.

In this comparison, we aim to examine the impact of each component of DORA-RI on its performance.

B. Simulation Results and Discussions

1) Comparison With the Baseline Schemes: Fig. 5 presents the mean QoE values achieved by DORA-RI, DORA, and PFWF-RM. The experiments are repeated 20 times for each algorithm and the average values are presented. The decreasing trend of QoE over increased number of users is intuitive. The reason lies in the fact that the average wireless resource each user can have is diminishing. In the worst case, DORA-RI can still achieve $1.7\times$ and $1.3\times$ QoE gains over PFWF-RM and DORA, respectively. It is clear that the enhancement achieved by utilizing the request interval of each user and by making more robust decisions based on the noisy future. DORA-RI achieves a much better performance than the original version DORA proposed in [1].

To better reveal the reason for the QoE gain, the average rebuffering time ratio results of the three schemes are presented in Fig. 6. Again, it is reasonable that the rebuffering ratio is increasing with increased number of UEs. When more UEs are served, each of them has a decreasing portion of the total resource. However, DORA-RI can keep the rebuffering time ratio low over the entire range, compared with the other two schemes. Especially when the resource is scarce for each user to smoothly support the transmission of video data. By saving the valuable resource form UEs that do not need them during the request interval, we can achieve much less stalling time, which in turn contributes to a high QoE score. Moreover, prediction of the future can help the algorithm to make better decisions to use a lower rate but to achieve smoother playback. Therefore, by smartly choosing smaller rates and more efficiently allocating resources, DORA-RI can achieve a $1/10$ rebuffering time ratio, which means much smoother playback and pleasant user experience.

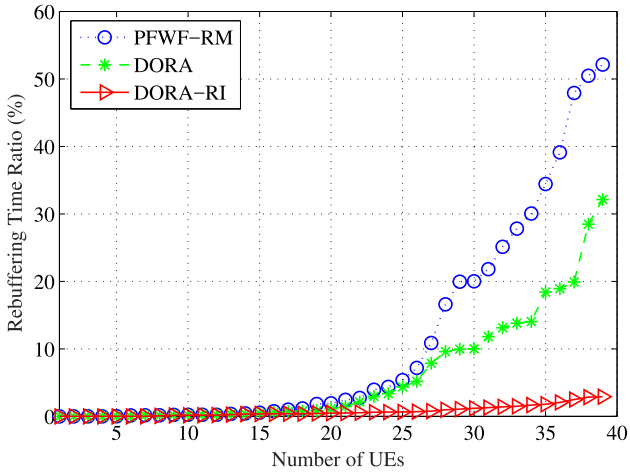


Fig. 6. Comparison of rebuffering ratio for DORA-RI, DORA and PFWF-RM.

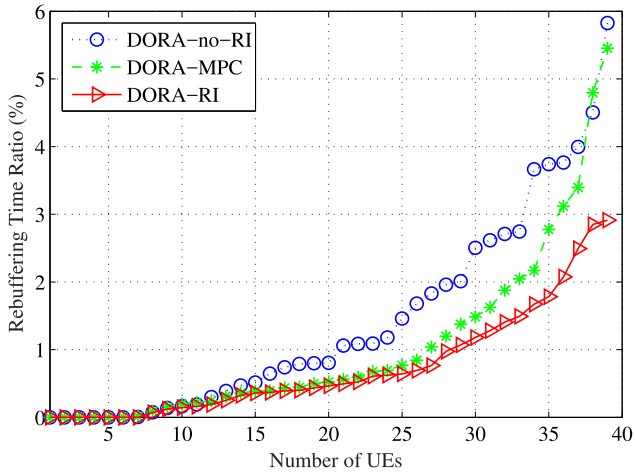


Fig. 8. Comparison of rebuffering ratio for DORA-RI, DORA-MPC and DORA-no-RI.

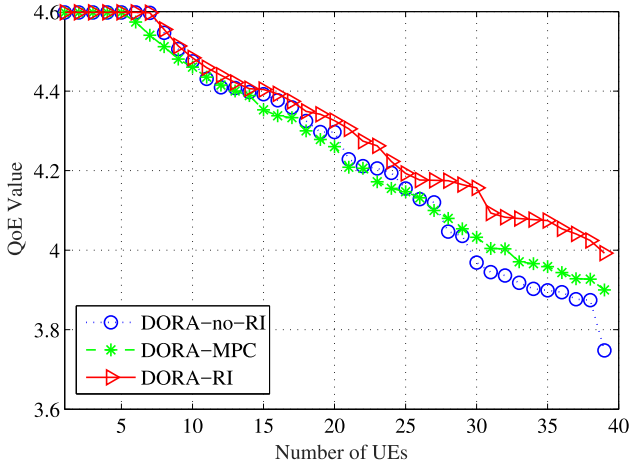


Fig. 7. Comparison of QoE score for DORA-RI, DORA-MPC and DORA-no-RI.

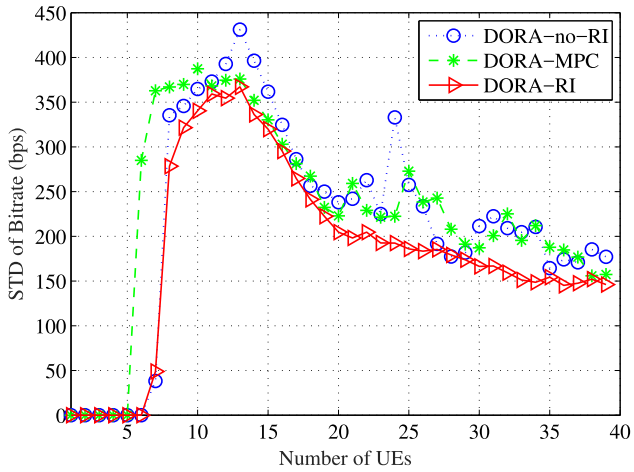


Fig. 9. Comparison of data-rate standard deviation for DORA-RI, DORA-MPC and DORA-no-RI.

2) Impact of DORA-RI Components: We compare the performance of DORA-RI with DORA-MPC, which uses MPC for rate adaption instead of SMPC, and DORA-no-RI that only optimizes rate adaption with no request interval consideration. In Fig. 7, the average QoE values achieved by each scheme are presented. DORA-RI achieves a 2% gain over DORA-MPC and an 8% gain over DORA-no-RI. Notice that the gain is under the influence of the logarithm utility function (8). The real gain in QoE values are 10% and 30% over DORA-MPC and DORA-no-RI, respectively. By considering the request interval, DORA-RI achieves a 30% gain and even a $1.8\times$ gain in the worst case. In addition, SMPC can provide a better performance over deterministic MPC by considering the stochastic dynamics of the network disturbances.

Fig. 8 illustrates how the average rebuffering time ratio is influenced by each component. It is obvious that all the three schemes outperform the two baseline schemes (i.e., DORA and PFWF-RM) comparing to Fig. 5. DORA-RI achieves a 45% smaller rebuffering time ratio than DORA-MPC in the worst case. This benefit is introduced by taking uncertainty into consideration while conducting the optimization. In addition, the consideration of request interval does enhance the rebuffering performance. However, we notice that when the resource gets really scarce, e.g., when the number of UEs becomes larger than 35, the performances of DORA-no-RI and DORA-MPC get closer. This is because the deterministic MPC lacks the ability to properly handle the more random request intervals when the network gets congested. This result actually justifies the necessity of incorporating SMPC.

In Fig. 9, the average standard deviation (STD) of bit rate over one playback process is presented. Here, we aim to understand how the bit rate varies during the process. Counter-intuitively, the STD does not always increase as the number of UEs is increased. We found that initially, there is sufficient resources such that each user could always choose the largest bit rate, leading to minimal STDs. When the number of UEs gets higher, the resource for each user is decreasing, which leaves them less choice on bit rates. Therefore, the STD also gets smaller. In the middle

range, however, the STD is relatively larger. This is because the resource is sufficient to alter the capacity largely but not enough to satisfy the largest bit rate. That is, it is harder to accurately predict future capacity in this range. We can see that DORA-RI still achieves the lowest STD of video rates among the variants.

VI. Related Work

HTTP based video streaming technologies has drawn great attention. Some interesting works are on modeling the new QoE model to support bit rate adaption, while bit rate adaptation strategy is an important problem in DASH. Many existing techniques could be generally categorized into: (i) buffer based (BB) techniques, (ii) capacity based (CB) schemes, and (iii) integrated techniques. In FESTIVE [21], the authors focused on improving the stability and fairness of multiple users that share a bottleneck link. PANDA [22] adopted a TCP congestion control like method to adjust video segment rate, but required certain overhead for probing. In [23], the authors considered rate adaptation with adjustment of a threshold. The authors in [24] proposed a regression method to predict the future capacity and a classical PID based controller for rate adaption. In [12], the authors proposed an MPC based approach for rate adaption at the user side and a fastMPC method. However, the authors used a different QoE model and did not consider the prediction method for robustness. Markov decision process (MDP) based rate adaption algorithms have been proposed in [25]–[27]. In [25], the authors applied DP to optimize the data rate adaption and showed that the DP algorithm in a shot horizon could be better than a stochastic decision just on the last network capacity. In [26], the authors proposed a solution, termed mDASH, that was based on MDP for DASH data rate adaption. Decision of the next segment data rate was made by taking factors into consideration, such as buffered video time, and history bandwidth and data rate. The work in [27] was focused on the vehicular environment. For MDP based algorithms, the high computational complexity is an obstacle for realtime applications. However, techniques such as Microsoft SmoothStreaming [28] have been shown to perform poorly in wireless networks [29]. Most rate adaption methods can only passively adapt to the variation in throughput.

The integration of video streaming and wireless network scheduling is a promising direction to maximize the performance of videos in the wireless environment. The authors in [30] tackled video transmissions in MU-MIMO networks and [31] investigated the adaptive video streaming and scheduling problem. Both works utilized Lyapunov optimization to achieve good performance with assumption of helper(s) as a centralized entity that can acquire user information. The authors in [32] proposed a joint optimization framework of scheduling and rate-adaption with a proof of optimality. However, this work did not explicitly consider the resource allocation details (e.g., power, subcarrier, and time assignment) and the QoE model used in this work was more based on a mathematical formation. Lin [33] proposed a cross layer method for scalable video streaming over OFDMA networks. The tradeoff between efficiency and fairness was addressed. In this work, the authors used the traditional PSNR as indication of video quality instead of the modern QoE models, which incorporate more important factors that influence user QoE. The authors of [34] tackled the problem of heterogeneous network streaming on the client side. The network cost constraints were considered and the goal was to achieve high quality streaming and low energy cost on the UE. In [35], on the other hand, the authors proposed a method that utilize both unicast and multicast concurrently to reduce the energy consumption of UEs when streaming video over a cellular network. These works are focused mainly on the UE side to achieve a tradeoff between video quality and energy efficiency.

Resource allocation in OFDM networks has been extensively studied in the past decades [4]–[6], [36]. Refer to [16] for a comprehensive survey. Two main topics, i.e., energy efficiency optimization and system throughput optimization, were generally addressed. The original problem is a mixture of integer programming (subcarrier assignment) and linear programming (power allocation), which is NP-hard. Therefore, suboptimal solutions were developed by relaxation of time sharing in one time frame [5], [36]. In [5], the authors proved that the suboptimal solution will converge to the optimum asymptotically assuming ergodic noise processes.

VII. Conclusion

We investigated the problem of resource allocation for delivering multiple videos using DASH over the downlink of an OFDMA network. We first presented an offline formulation based on a novel QoE model. We then derived a more practical online formulation, and developed a distributed solution algorithm, which consisted of an resource allocation algorithm at the BS side and a rate adaptation algorithm at the user side. The proposed scheme was validated with simulations and was shown to outperform several baseline schemes and variants with considerable gains. For future work, if subjective data is available, it would be interesting to develop machine learning techniques to learn the weights in the QoE model.

Appendix A
Proof for Theorem 1

Proof: By redefining the QoE maximization problem as in (14), the QoE function can be written as

$$Q^{\text{appr}} = q_i[k_i] + \theta(q_i[k_i] - m_i[K_i])^2 + \frac{\lambda}{K_i l}\max\left\{0, \pi_i[k_i] + \frac{q^{-1}(q_i[k_i])l}{\bar{C}_i[k_i]} - B_i[k_i - 1]\right\}, \tag{33}$$

where $q_i[k_i]$ is a linear function. The quadratic part is a convex function over $q_i[k_i]$. The $\max$ term is convex over $q_i[k_i]$ and convex over $p_{im}(t)$ and $v_{im}(t)$. Since parameters θ and λ are both negative, Q^{appr} is concave for all the variables. The target function $\Lambda^{\text{appr}}(N)$ is the summation of Q^{appr} over time and users, which means it is concave as well. The constraints are all convex sets. Therefore, this is a convex problem. ■

APPENDIX B
PROOF FOR LEMMA 1

Proof: This Lemma could be presented as follows.

$$\lim_{K\to\infty}\left(\frac{1}{K}\sum_{k_i=1}^{K} q_i(R^*[k_i]) - \hat{m}_i[K]\right) = 0 \tag{34}$$

For brevity, we denote $q_i(R^*[k_i])$ by $q_i^*[k_i]$. Rewrite (17) and take summation from 1 to K. We have

$$\sum_{k_i=1}^{K}\left(\frac{k_i+\zeta}{\zeta}\right)(\hat{m}_i[k_i] - \hat{m}_i[k_i-1]) = \sum_{k_i=1}^{K}(q_i^*[k_i] - \hat{m}_i[k_i-1]). \tag{35}$$

Expand the left hand side summation to obtain

$$\frac{1}{\zeta}\left(K\hat{m}_i[K] - \sum_{k_i=1}^{K}\hat{m}_i[k_i-1]\right) - (\hat{m}_i[K] - \hat{m}_i[1]) = \sum_{k_i=1}^{K}(q_i^*[k_i] - \hat{m}_i[K] + \hat{m}_i[K] - \hat{m}_i[k_i-1]).$$

To obtain the form (34), we first divide it by K and then take limit on both sides over K. It follows that

$$\lim_{K\to\infty}\frac{K\hat{m}_i[K] - \sum_{k_i=1}^{K}\hat{m}_i[k_i-1]}{\zeta K} - \lim_{K\to\infty}\frac{\hat{m}_i[K] - \hat{m}_i[1]}{K} = \lim_{K\to\infty}\sum_{k_i=1}^{K}(q_i^*[k_i] - \hat{m}_i[K] + \hat{m}_i[K] - \hat{m}_i[k_i-1]).$$

It's obvious that the second term goes to 0 due to finite $\hat{m}_i[K]$ and $\hat{m}_i[1]$. By splitting the right-hand-side into two parts and rearranging terms, we have:

$$\frac{1-\zeta}{\zeta}\lim_{K\to\infty}\left(K\hat{m}_i[K] - \sum_{k_i=1}^{K}\hat{m}_i[k_i-1]\right) = \lim_{K\to\infty}\frac{1}{K}\sum_{k_i=1}(q_i^*[k_i] - \hat{m}_i[K]).$$

The left-hand-side is the limit we want to show to be 0. Note that (17) can be viewed as a stochastic approximation update equation for $\hat{m}_i$. Therefore, the convergence of the right-hand-side can be proven by applying Theorem 1.1 of Chapter 6 in [37]. ■

APPENDIX C
PROOF FOR THEOREM 2

Proof: The convergence proof could be transformed to show that the Prob-Online objective value converges to the Prob-Offline objective value. That is, we aim to prove

$$\lim_{t\to\infty}\Lambda(\mathbf{S}^*) - \Lambda(\tilde{\mathbf{S}}) = 0, \tag{36}$$

where $\mathbf{S}^* = \{\mathbf{q}^*(t), \mathbf{v}^*(t), \mathbf{p}^*(t)\}$ is the online solution and $\tilde{\mathbf{S}} = \{\tilde{\mathbf{q}}(t), \tilde{\mathbf{v}}(t), \tilde{\mathbf{p}}(t)\}$ is the offline solution.

Since Prob-Online is convex, and based on Lemma 2, we can derive the KKT condition as follows.

$$\begin{cases} \mathbb{I}_N\left(\Lambda(\mathbf{S_i}^*)\mathbf{S} + \alpha_i^*(t) - \gamma_{i,\min}^*(t) + \gamma_{i,\max}^*(t)\right) \\ \quad + \beta_{i,m}^*(t) = 0 \\ \alpha_i^*(t)(v_{im}^* - 1) = 0 \\ \beta_{i,m}^*(p_{im}^* - 1) = 0 \\ \gamma_{i,\min}^*(q_{i,\min} - q_i^*) = 0 \\ \gamma_{i,\max}^*(q_i^* - q_{i,\max}) = 0 \\ \alpha_i^*(t), \beta_{im}^*(t), \gamma_{i,\min}^*(t), \gamma_{i,\max}^*(t) \geq 0, \\ \quad \forall i \in \mathbb{N}, \forall m \in \mathbb{M}, \end{cases} \tag{37}$$

where $\mathbb{I}_N$ is the indicator that the variables belong to different users. Lagrange multipliers $\alpha_i^*(t)$, $\beta_{i,m}^*(t)$, $\gamma_{i,\min}^*(t)$, and $\gamma_{i,\max}^*(t)$ are the dual points that the KKT conditions are satisfied and the optimal value is achieved.

We next construct the help function $\mathbf{H}(\tilde{\mathbf{S}})$ that is differentiable and concave. The definition is

$$\mathbf{H}(\tilde{\mathbf{S}}) = \Lambda(\tilde{\mathbf{S}}) - \alpha_i^*(t)(v_{im}^* - 1) - \beta_{i,m}^*(p_{im}^* - 1) + \gamma_{i,\min}^*(q_i^* - q_{i,\min}) - \gamma_{i,\max}^*(q_i^* - q_{i,\max}). \tag{38}$$

We have $\mathbf{H}(\tilde{\mathbf{S}}) \geq \Lambda(\tilde{\mathbf{S}})$ due to the constraints in problem (14). In addition, we can acquire the following inequality by the concavity and differentiability properties of $\mathbf{H}(\tilde{\mathbf{S}})$.

$$\mathbf{H}(\tilde{\mathbf{S}}) \leq \mathbf{H}(\mathbf{S}^*) + \mathbf{H}(\mathbf{S}^*)\mathbf{S}(\mathbf{S}^* - \tilde{\mathbf{S}}). \tag{39}$$

From (38) and the KKT conditions (37), we obtain the following inequality.

$$\Lambda(\tilde{\mathbf{S}}) \leq \Lambda(\mathbf{S}^*) + 2\theta\sum_{i=1}^{N}\sum_{t=0}^{T}(q_i^*(t) - \hat{m}_i) * (q_i^*(t) - \tilde{q}_i(t).$$

In Lemma 1, we have proved that $\lim_{K\to\infty}\frac{1}{K}\sum_{k_i=1}(q_i^*[k_i] - \hat{m}_i[K])$. Here we abuse the notations of K_i and T, which both represent the total playback time. Then we have

$$\lim_{T\to\infty}\frac{1}{T}\sum_{k_i=1}\Lambda(\tilde{\mathbf{S}}) \leq \frac{1}{T}\lim_{T\to\infty}\Lambda(\mathbf{S}^*). \tag{40}$$

Since $\tilde{\mathbf{S}}$ is the optimal offline solution, $\Lambda(\tilde{\mathbf{S}})$ is the optimal value for Prob-Offline. It follows that

$$\Lambda(\tilde{\mathbf{S}}) \geq \Lambda(\tilde{\mathbf{S}}). \tag{41}$$

Combining (40) and (41), we conclude that the theorem holds true. ■

APPENDIX D
PROOF FOR THEOREM 3

Proof: First of all, we prove the concavity of the rate function, which can be rewritten as $r_{im}(t) = v(t)\kappa\log_2(1 + \frac{p(t)g^2(t)}{v(t)\kappa})$ for brevity. For two feasible solutions (v_2, p_2) and (v_3, p_3), suppose there exists a feasible solution $(v_1, p_1) = \epsilon(v_2, p_2) + (1-\epsilon)(v_3, p_3)$, where $0 \leq \epsilon \leq 1$. By the definition of concavity, we need to show that $r_{im}(v_1, p_1) \geq \epsilon r_{im}(v_2, p_2) + (1-\epsilon)r_{im}(v_3, p_3)$. There are three possible cases for v_2 and v_3 that needs to be analyzed.

- *Case I:* when $\{v_2 > 0, v_3 > 0\}$. Then we have $v_1 > 0$. Because $\log_2(1 + p_1 g_{im}^2)$ is a concave function of p_1, its *perspective* $v_1 \nu \log_2(1 + \frac{p_1 g_{im}^2}{v_1 \nu})$ is concave given that $v_1 > 0$ [38]. Thus, concavity holds true for this case.
- *Case II:* when $\{v_2 > 0, v_3 = 0\}$. Then we have $v_1 = \epsilon v_2$ and $p_1 = \epsilon p_2 + (1-\epsilon)p_3$. It follows that

$$r_{i,m}(v_1, p_1) = \epsilon v_2 \kappa \log_2 \left(1 + \frac{(\epsilon p_2 + (1-\epsilon)p_3) g_{im}^2}{\epsilon v_2 \kappa}\right) \geq \epsilon v_2 \log_2 \left(1 + \frac{p_2 g_{im}^2}{v_2 \kappa}\right).$$

- *Case III:* when $\{v_2 = 0, v_3 = 0\}$. This is a trivial case that the equality holds for $v_1 = 0$.

With the concavity of $r_{im}(t)$, it is convenient to show that the $\sum_{s=1}^{S} r_{im}(t)$ is concave since it is a nonnegative combination of concave functions $r_{im}(t)$. The concavity of the cost function $\Phi(p_{im}(t), v_{im}(t))$ can be argued for the same reason.

In addition, the utility constraints define a convex set Υ. By the definition of utility function, it's convenient to show that for any $r_1, r_2 \in \Upsilon$, $\epsilon r_1 + (1-\epsilon) r_2 \in \Upsilon$ for any ϵ with $0 \leq \epsilon \leq 1$. Both constraints $\sum_{i=1}^{N}(t) v_{im}(t) \leq 1$ and $\sum_{i=1}^{N}(t) \sum_{m=1}^{M} p_{im}(t) \leq P$ also define convex sets. They are the *intersection* of *half-spaces*. Intersection preserves the convexity of the sets. Thus the constraints are convex.

We conclude that Problem (29) is convex. ■

REFERENCES

[1] K. Xiao, S. Mao, and J. Tugnait, "QoE-driven resource allocation for DASH over OFDMA networks," in *Proc. IEEE IEEE Global Commun. Conf.*, Washington, DC, USA, Dec. 2016, pp. 1–6.

[2] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update, 2016–2021," Cisco, San Jose, CA, USA, White Paper, 2018. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html

[3] Sandvine, "Global Internet phenomena report," Sandvine, San Jose, CA, USA, White Paper, 2017. [Online]. Available: https://www.sandvine.com/trends/global-internet-phenomena/

[4] Z. Mao and X. Wang, "Efficient optimal and suboptimal radio resource allocation in OFDMA system," *IEEE Trans. Wireless Commun.*, vol. 7, no. 2, pp. 440–445, Feb. 2008.

[5] X. Wang and G. B. Giannakis, "Resource allocation for wireless multiuser OFDM networks," *IEEE Trans. Inf. Theory*, vol. 57, no. 7, pp. 4359–4372, Jul. 2011.

[6] Z. Ren, S. Chen, B. Hu, and W. Ma, "Energy-efficient resource allocation in downlink OFDM wireless systems with proportional rate constraints," *IEEE Trans. Veh. Technol.*, vol. 63, no. 5, pp. 2139–2150, Mar. 2014.

[7] DASH Industry Forum, "The dash.js project." [Online]. Available: https://github.com/Dash-Industry-Forum/dash.js/wiki

[8] A. Balachandran *et al.*, "Developing a predictive model of quality of experience for internet video," in *Proc. ACM SIGCOMM Conf. SIGCOMM*, Hong Kong, China, Aug. 2013, pp. 339–350.

[9] A. Khan, L. Sun, and E. Ifeachor, "Content clustering based video quality prediction model for MPEG4 video streaming over wireless networks," in *Proc. IEEE Int, Conf, Commun.*, Dresden, Germany, Jun. 2009, pp. 1–5.

[10] J. You, U. Reiter, M. M. Hannuksela, M. Gabbouj, and A. Perkis, "Perceptual-based quality assessment for audio–visual services: A survey," *Signal Process.: Image Commun.*, vol. 25, no. 7, pp. 482–501, 2010.

[11] F. Dobrian *et al.*, "Understanding the impact of video quality on user engagement," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, pp. 362–373, Aug. 2011.

[12] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over HTTP," in *Proc. Proc. 2015 ACM Conf. Special Interest Group Data Commun.*, London, U.K., Aug. 2015, pp. 325–338.

[13] J. Jiang *et al.*, "CFA: A practical prediction system for video QoE optimization," in *Proc. 13th USENIX Symp. Netw. Syst. Des. Implementation*, Santa Clara, CA, USA, Mar. 2016, pp. 137–150.

[14] C. Chen, X. Zhu, G. de Veciana, A. C. Bovik, and R. W. Heath, "Rate adaptation and admission control for video transmission with subjective quality constraints," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 1, pp. 22–36, Feb. 2015.

[15] C. Chen *et al.*, "A dynamic system model of time-varying subjective quality of video streams over HTTP," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Vancouver, Canada, May 2013, pp. 3602–3606.

[16] F. Shams, G. Bacci, and M. Luise, "A survey on resource allocation techniques in OFDMA networks," *Elsevier Comput. Netw.*, vol. 65, no. 2, pp. 129–150, Jun. 2014.

[17] D. P. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1439–1451, Aug. 2006.

[18] K. Winstein, A. Sivaraman, and H. Balakrishnan, "Stochastic forecasts achieve high throughput and low delay over cellular networks," in *Proc. 10th USENIX Symp. Netw. Syst. Des. Implementation*, Lombard, IL, USA, Apr. 2013, pp. 459–471.

[19] S. Lederer, C. Müller, and C. Timmerer, "Dynamic adaptive streaming over HTTP dataset," in *Proc. ACM Multimedia*, Nara, Japan, Oct./Nov. 2012, pp. 89–94.

[20] H. J. Kushner and P. A. Whiting, "Convergence of proportional-fair sharing algorithms under general conditions," *IEEE Trans. Wireless Commun.*, vol. 3, no. 4, pp. 1250–1259, Jul. 2004.

[21] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with festive," *IEEE/ACM Trans. Netw.*, vol. 22, no. 1, pp. 326–340, Jan. 2014.

[22] Z. Li *et al.*, "Probe and adapt: Rate adaptation for HTTP video streaming at scale," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 4, pp. 719–733, Apr. 2014.

[23] 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 *Proc. ACM Conf. SIGCOMM*, Chicago, IL, USA, Aug. 2014, pp. 187–198.

[24] G. Tian and Y. Liu, "Towards agile and smooth video adaptation in dynamic HTTP streaming," in *Proc. 8th Int. Conf. Emerg. Netw. Experiments Technol.*, Nice, France, Dec. 2012, pp. 109–120.

[25] S. Hu, L. Sun, C. Gui, E. Jammeh, and I.-H. Mkwawa, "Content-aware adaptation scheme for QoE optimized DASH applications," in *Proc. IEEE Global Commun. Conf.*, Austin, TX, USA, Dec. 2014, pp. 1336–1341.

[26] C. Zhou, C.-W. Lin, and Z. Guo, "mDASH: A Markov decision-based rate adaptation approach for dynamic HTTP streaming," *IEEE Trans. Multimedia*, vol. 18, no. 4, pp. 738–751, Apr. 2016.

[27] A. Bokani, M. Hassan, S. Kanhere, and X. Zhu, "Optimizing http-based adaptive streaming in vehicular environment using Markov decision process," *IEEE Trans. Multimedia*, vol. 17, no. 12, pp. 2297–2309, Dec. 2015.

[28] Microsoft, "Smooth streaming protocol." (2017). [Online]. Available: https://msdn.microsoft.com/en-us/library/ff469518.aspx

[29] S. Akhshabi, A. C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP," in *Proc. 2nd Annu. ACM Conf. Multimedia Syst.*, San Jose, CA, USA, Feb. 2011, pp. 157–168.

[30] D. Bethanabhotla, G. Caire, and M. J. Neely, "Adaptive video streaming in MU-MIMO networks," 2014, *arXiv:1401.6476*. [Online]. Available: https://arxiv.org/abs/1401.6476.

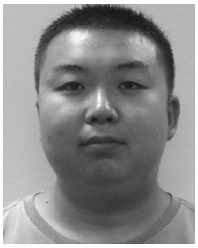
[31] D. Bethanabhotla, G. Caire, and M. J. Neely, "Adaptive video streaming for wireless networks with multiple users and helpers," *IEEE Trans. Commun.*, vol. 63, no. 1, pp. 268–285, Jan. 2015.

[32] V. Joseph and G. de Veciana, "NOVA: QoE-driven optimization of DASH-based video delivery in networks," in *Proc. IEEE INFOCOM—IEEE Conf. Comput. Commun.*, Toronto, Canada, Apr./May 2014, pp. 82–90.

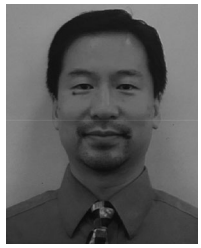
[33] K. Lin and S. Dumitrescu, "Cross-layer resource allocation for scalable video over OFDMA wireless networks: Trade-off between quality fairness and efficiency," *IEEE Trans. Multimedia*, vol. 19, no. 7, pp. 1654–1669, Jul. 2017.

[34] Y. Go, O. C. Kwon, and H. Song, "An energy-efficient HTTP adaptive video streaming with networking cost constraint over heterogeneous wireless networks," *IEEE Trans. Multimedia*, vol. 17, no. 9, pp. 1646–1657, Sep. 2015.

[35] S. Almowuena, M. M. Rahman, C.-H. Hsu, A. A. Hassan, and M. Hefeeda, "Energy-aware and bandwidth-efficient hybrid video streaming over mobile networks," *IEEE Trans. Multimedia*, vol. 18, no. 1, pp. 102–115, Jan. 2016.
[36] J. Huang, V. G. Subramanian, R. Agrawal, and R. A. Berry, "Downlink scheduling and resource allocation for OFDM systems," *IEEE Trans. Wireless Commun.*, vol. 8, no. 1, pp. 288–296, Jan. 2009.
[37] J. Harold, G. Kushner, and G. Yin, *Stochastic Approximation and Recursive Algorithm and Applications*, 2nd ed. New York, NY, USA: Springer-Verlag, 2003.
[38] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.



Kefan Xiao (S'14) received the B.E. degree from Xian Jiaotong University, Xi'an, China, in 2011, and M.S. degree from Shanghai Jiaotong University, Shanghai, China, in 2014, both in electronic engineering. He received the Ph.D. degree in electrical and computer engineering from Auburn University, Auburn, AL, USA, in December 2018. He is currently a Software Engineer with Google Inc. His research interests include TCP congestion control and video streaming.



Shiwen Mao (S'99–M'04–SM'09–F'19) received the Ph.D. degree in electrical and computer engineering from Polytechnic University, Brooklyn, NY, USA. Currently, he is the Samuel Ginn Distinguished Professor with the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL, USA, and the Director of the Wireless Engineering Research and Education Center (WEREC) at Auburn University.

His research interests include wireless networks, multimedia communications, and smart grid. Dr. Mao is a member of the ACM. He was co-recipient of the IEEE ComSoc MMTC Best Journal Paper Award in 2019 and the MMTC Best Conference Paper Award in 2018, IEEE SECON 2017 Best Demo Award, Best Paper Awards from IEEE GLOBECOM 2016, IEEE GLOBECOM 2015, IEEE Wireless Communications and Networking Conference 2015, and IEEE International Conference on Communications 2013, and the 2004 IEEE Communications Society Leonard G. Abraham Prize in the Field of Communications Systems. He is a Distinguished Speaker of the IEEE Vehicular Technology Society. He is on the Editorial Board of IEEE Open Journal of the Communications Society,IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE INTERNET OF THINGS JOURNAL, IEEE MULTIMEDIA, IEEE NETWORKING LETTERS, and *ACM GetMobile*, among others.

Jitendra K. Tugnait (M'79–SM'93–F'94) received the B.Sc.(Hons.) degree in electronics and electrical communication engineering from Punjab Engineering College, Chandigarh, India, in 1971, the M.S. and the E.E. degrees from Syracuse University, Syracuse, NY, USA, in 1973 and 1974, respectively, and the Ph.D. degree from the University of Illinois, Urbana-Champaign, Champaign, IL, USA, in 1978, all in electrical engineering.

From 1978 to 1982, he was an Assistant Professor of Electrical and Computer Engineering with the University of Iowa, Iowa City, IA, USA. He was with the Long Range Research Division of the Exxon Production Research Company, Houston, TX, USA, from June 1982 to September 1989. He joined the Department of Electrical & Computer Engineering, Auburn University, Auburn, AL, in September 1989 as a Professor. He currently holds the title of James B. Davis Professor. His current research interests are in statistical signal processing, wireless communications, and multiple target tracking.

Dr. Tugnait was an Associate Editor for the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE SIGNAL PROCESSING LETTERS, and the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and was the Senior Area Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING. He is currently a Senior Editor for IEEE WIRELESS COMMUNICATIONS LETTERS.