Towards Guaranteed Video Experience: Service-aware Downlink Resource Allocation in Mobile Edge Networks

Zhisheng Yan, *Member, IEEE,* Miao Zhao, *Member, IEEE,* Cedric Westphal, *Senior Member, IEEE,* and Chang Wen Chen, *Fellow, IEEE*

***Abstract*—Video delivery has been playing an essential role in video services over edge networks. Although HTTP segment based streaming, e.g., Dynamic Adaptive Streaming over HTTP (DASH), has become the prevailing technique, it cannot provide guaranteed video playback in terms of bitrate to mobile users. In essence, HTTP streaming downloads the video segments in a best-effort fashion, i.e., passively responding to the channel dynamics. This can cause instable playback with frequent rebuffer and multi-client competition that degrades network-wide performance. In this paper, we present GESH, a network-assisted streaming framework for Guaranteed Playback-Experience Streaming over HTTP that leverages the proactive control of network resources and joint coordination among multiple clients for service-aware network resource allocation. Specifically, GESH is empowered by a new weighted proportional fair scheduling without modifying existing cellular infrastructure, a per-segment channel variation model, and a suite of algorithms to seek the optimal weights for the scheduling. Extensive evaluations show that GESH can maximally guarantee the video playback of multiple users, as well as significantly outperforming conventional HTTP streaming and current DASH systems.**

***Index Terms*—Guaranteed playback, HTTP video streaming, mobile edge networks, resource allocation.**

I. Introduction

THANKS to the explosion of mobile web access and mobile social networks, mobile video services have been boosted drastically [1]. Numerous applications providing visual content, such as video streaming, social video sharing, and mobile video surveillance, have gained broad popularity and significantly enriched the everyday life of mobile users. Typically, these video services acquire source content from video content producers, and then deliver the encoded video to mobile clients through mobile edge networks. Considering the ever-increasing compression ratio of advanced video coding techniques, video delivery over mobile edge networks is now becoming the key design issue in mobile streaming systems.

Zhisheng Yan is with Department of Computer Science, Georgia State University (email: zyan@gsu.edu).

Miao Zhao is with Department of Computing, The Hong Kong Polytechnic University (email: csmiaozhao@comp.polyu.edu.hk).

Cedric Westphal is with Huawei Innovation Center and Computer Engineering Department, University of California, Santa Cruz (email: cedric.westphal@huawei.com).

Chang Wen Chen is with School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen and Department of Computer Science and Engineering, University at Buffalo (email: chencw@cuhk.edu.cn, chencw@buffalo.edu).

HTTP segment based streaming has recently become the dominant technology for video delivery due to its compatibility with Internet's hardware and software infrastructure. The video source is encoded into small segments that can be decoded individually. Thus the mobile client can playback the video without downloading the entire video file.

In order to improve users' experience in bandwidth-varied mobile networks, dynamic adaptive streaming over HTTP (DASH) has been widely deployed for various video services. Unlike conventional HTTP segment based streaming, the DASH source is encoded into multiple bitrate versions, each of which is then divided into small segments. The client adaptively requests the most appropriate segment at each switching point based on its throughput measurement of the mobile channel. That way, the user is expected to maximally utilize mobile bandwidth and achieve satisfactory experience.

Despite the prevalence of DASH, we observe that current HTTP streaming framework cannot provide mobile users with *guaranteed video playback*, i.e., ensuring a higher video bitrate than a certain threshold, which could significantly degrade the user satisfaction. First, current video playback is unreliable since DASH bandwidth estimation is not accurate in mobile edge networks with strong channel dynamics. The DASH client using per-segment throughput based estimation frequently overestimates its bandwidth share if the shared channel has been exclusively used by itself [2], [3]. Bandwidth underestimation will also occur if the download of a segment does not saturate the entire bandwidth [2], [3]. Thus it is unlikely to achieve stable playback without rebuffer, let alone guaranteed video playback. Second, the client-driven nature of current systems incurs intractable and poor performance in the multi-client mobile edge networks. This is because the mobile client has no knowledge of the other shared streams in the same bottleneck. Even though a client might achieve satisfactory playback through aggressive bitrate requests, the bandwidth share of other competing clients would be squeezed and their playback would not be guaranteed [4], [5].

In fact, the fundamental problem behind these issues is that both traditional HTTP streaming and current DASH employ a *best-effort* strategy, wherein they can only respond to the channel dynamics passively in order to *enhance* the playback performance of individual users. There is no proactive control of radio resources or joint coordination among multiple clients in order to *guarantee* the playback of multiple users. Therefore, the goal of this research is to provide guaranteed

video playback performance to multiple users by leveraging the centralized downlink resource allocation at the radio access network (RAN). We aim at maximizing the system bandwidth efficiency while guaranteeing the playback performance of multiple HTTP streaming clients in one mobile cell. Note that we guarantee the playback experience in terms of bitrate. Although this is not equivalent to the subjective Quality of Experience (QoE), it is the most fundamental factor that contributes to the QoE. Guaranteeing the bitrate will lay the foundation for the ultimate goal of QoE guarantee.

To achieve this goal, we face several technical challenges.

- *Allocating radio resources in a service-aware manner without modifying current cellular infrastructure*: Proportional fair scheduler (PFS) has been widely deployed at the link layer of RANs [6]. How should the client-driven application-layer video service interact with the network-side PFS and guide the resource allocation based on the service requirement is an essential yet difficult task. It is necessary to feed this service information to the scheduler while keeping the core PFS algorithm unchanged.
- *Estimating the per-segment throughput of cellular channels*: Although the instant channel rate of each radio resource unit is available within the RAN (e.g., at the base station), allocating appropriate amount of radio resources in order to achieve a certain throughput during the next segment is still non-trivial. This is because cellular channels are extremely dynamic and thus it is challenging to obtain the future channel rate. Furthermore, we would not be able to know which exact resource units would be assigned to a given user before performing the PFS algorithm. Therefore, it is necessary to estimate the per-segment throughput of a user deliberately.
- *Guaranteeing the playback of multiple video users under shared bottleneck*: The service-aware downlink resource allocation is fundamentally dependent on both channel and playback conditions. Allocating more resource units to a user with smaller instant channel rate may improve the user's playback, but will leave much less resource units for other competing users with the same service requirement. The playback status such as buffer size also matters since a user with larger buffer may not need video data as eagerly as those users with smaller buffer size.

To tackle these challenges, we present *GESH*, a framework for **G**uaranteed Playback-**E**xperience **S**treaming over **H**TTP in mobile edge networks. GESH is a network-assisted streaming framework, where the client-side information is communicated to the RAN for service-aware resource allocation in a standard-compliant way. To maintain current PFS infrastructure, we propose to exploit a weighted proportional fair scheduling, where the user weights are input parameters to the core PFS algorithm. The weights can be dynamically adjusted based on multiple users' segment requests, buffer status, and channel variations, which will accordingly guide the radio resource allocation and guarantee the video playback.

To optimize the service-aware resource allocation in GESH, we start with modeling the per-segment channel variation as a Markov model by utilizing large-scale channel data. Based on the channel model, we formulate a resource allocation optimization problem that maximizes the throughput under the constraints of segment requests, buffer occupancy and bandwidth limit. We then design an optimal dynamic programming algorithm and an efficient greedy algorithm that is proven to be optimal in order to seek the optimal weights of users. Extensive simulations under various settings show that GESH can maximally guarantee the video playback of multiple users, as well as significantly outperforming conventional HTTP segment based streaming and current DASH systems.

To summarize, the contributions of this research include:

- A weighted proportional fair scheduling framework that considers both channel and video service information (Section III).
- A per-segment channel rate model and a suite of algorithms that jointly achieve the optimal resource allocation (Section III-IV).
- A demonstration of the effectiveness of the service-aware GESH via systematic simulations (Section V).

II. Related Work

A. DASH and Rate Adaptation

Recently, both MPEG [7] and 3GPP [8] have made tremendous efforts towards the standardization of DASH, which indicates the prevalent adoption of this technology in video industry. DASH standard defines the media presentation description (MPD), segment format, and implementation guideline of the system. The specific rate adaptation strategies are not part of the standard and are left to system designers.

Early works of DASH focused on the rate adaptation algorithms for individual video client [9], [10], [11] to fill in the gap of DASH standard. A survey of rate adaptation and QoE for DASH was presented in [12]. In general, the client dynamically estimates the maximum video bitrate supported by the end-to-end bandwidth based on its local measurement, e.g., using throughput and buffer status. However, all these algorithms target the single-user client-side adaptation and cannot guarantee the playback performance in multi-client edge networks. In fact, playback instability and unfairness have been commonly identified as among the weaknesses for client-side adaptation in practical experiments where two or more clients compete for the same bottleneck [4], [5], [13].

B. Video Delivery for Multiple Clients

Recently, there has been an obvious trend for the investigation of multi-client HTTP streaming. MPEG has included server and network-assisted DASH into its working draft [14]. Besides, 3GPP has pointed out the use cases of network assistance in its latest specification [15], which indicates future standardization for technical details.

To address the unfairness issue, backend server side traffic shaping has been studied. In [16], the authors stabilized the source traffic from the server for competing clients when oscillations are detected. In another traffic shaping scheme [13], the traffic of the client who enjoyed a higher quality would be firstly downgraded. These server-side schemes

demonstrate the principle of centralized control. However, they are not specifically designed for cellular edge networks. These schemes work at the application layer and impose no constraint on the physical-layer radio resources, which can lead to either underutilized or overutilized bandwidth for video playback. More importantly, such architecture requires to record server states rather than using pure stateless HTTP, resulting in scalability and reliability issues.

Little work has been carried out to design DASH over multi-client edge networks. Most related schemes [4], [17], [18], [19], [20], [21] aimed at jointly adapting the bitrate of multiple users to optimize a certain utility, subject to instant channel rate constraints. Although the requested bitrate is more reasonable, the application-layer bitrate selection is oblivious to downlink resource allocation. Consequently, the resulting throughput that is an output of downlink scheduling may still deviate from the selected bitrate significantly. More importantly, without the intelligent resource control, the client's playback performance solely relies on the channel conditions, which provides no guarantee of playback performance. Moreover, there were other schemes [22], [23] that combined the designs of rate adaptation and resource allocation in order to allow service-channel-aware playback. However, they strongly depend on the customized low-layer scheduler and require the complete replacement of the standardized proportional fair scheduler, which makes them difficult to be applied into practice. It is also extremely difficult to evaluate their realistic value. First, completely new architecture indicates a large number of customized interfaces, modules, and self-defined parameters. It is not yet clear how these settings would impact the real-world cellular networks and what would be the optimal configuration. Second, measuring the replacement cost and compare it with the playback performance is also non-trivial and needs a full-scale cross-discipline investigation across business, product, and research teams.

C. Downlink Resource Allocation

Proportional fair scheduling algorithms [6], [24] have been widely deployed to maximize the system throughput and guarantee the fairness among multiple users. In each scheduling cycle, PFS sequentially schedules a given resource unit, e.g., a resource block in LTE, to the user who enjoys the maximum ratio between instant channel rate and the average past throughput. This process repeats until all radio resources have been allocated. To accommodate heterogeneous downlink channels, a PFS that dynamically monitors and adjusts the scheduling metric was proposed in [25]. To satisfy the differentiated requirement, a parameterized downlink scheduling algorithm that seeks a flexible tradeoff between throughput and fairness was proposed in [26]. These algorithms provide the foundations of PFS downlink scheduling and resource allocation. However, they focus on general data traffic and do not consider the unique effects of video playback, which can lead to unacceptable experience for the video users.

To address the time-sensitive issue of multimedia traffic, maximum-largest weighted delay first [27] and logarithm-rule [28] scheduling algorithms were proposed with the consideration of delay constraints. They increased the priority of delay-sensitive flows when their head-of-line delays were close to a certain threshold. However, packet-level delay bound may not necessarily have direct relationship with playback performance. For example, even with some packets exceeding the delay bound, a user can still achieve desired experience as long as the player buffer is not underflowed. In [29], a joint application, MAC and physical layer design was proposed for video delivery over OFDMA networks. Recently, application driven network [30] was proposed to address differentiated application requirement by slicing the physical network into different logically isolated subnetworks using independent protocols. Nevertheless, these designs need to completely replace existing PFS infrastructure, which is relatively impractical. More importantly, these resource allocation schemes are not specifically designed for HTTP streaming and therefore the reported results may not be consistent in DASH environment.

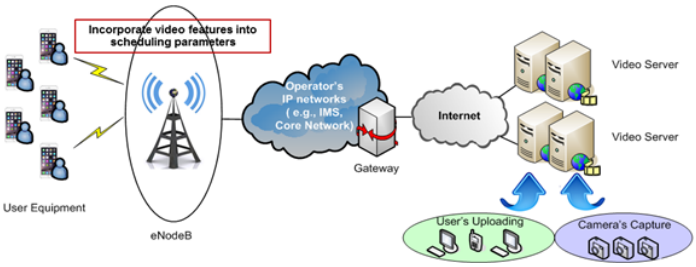


Fig. 1. The system architecture considered in this paper.

D. Summary

Indeed, these existing works provide important insights for the design of the proposed GESH framework. However, the objective in this research is fundamentally different in that GESH aims at guaranteeing the video playback proactively rather than improving the performance in a best-effort manner. First, GESH strives to exploit the channel variation by estimating the per-segment achievable throughput that is the output of the PFS rather than by directly using the instant channel rate. Coupled with the application information, GESH further adjusts the radio resource share (i.e., weights) of each user on top of the standard PFS scheduler (instead of replacing it) in order to maximally guarantee the video playback of multiple users. Moreover, GESH is transparent to the rate selection in HTTP segment based streaming. It simply intends to guarantee the requested bitrate rather than modifying the bitrate requests.

III. System Models

A. System Architecture

The system architecture under investigation is shown in Fig. 1. The video server collects video sources from video content producers for on-demand or live video services. The source sequences will be transcoded into multiple bitrate versions to provide different levels of video experience. Each version of the video is split into multiple segments with the same segment length. Note that, in the case of live videos, the segments are prepared in real-time along with the live event.

In this paper, we consider a single LTE cell. We assume all the edge entities, e.g., the base station and PFS scheduler, operate in the same way as current LTE networks. The key intelligence lies in an additional module that calculates the input parameters of PFS and thus guides the PFS resource allocation. This allows us to explore the collective knowledge of multiple streams for service-aware resource control. Therefore, the network is expected to guarantee the segment requests and the playback of different users. Such an architecture may be utilized as advanced network configuration for any other applications with relatively strict service requirements.

We consider a set of premium users subscribing to the video service that promises guaranteed video playback. Depending on the account priority, premium users may also have different levels of guaranteed playback performance. For example, some are guaranteed to enjoy smooth HD video services whereas some others have smooth SD video services. Since we propose to guarantee playback performance by downlink resource allocation, the bitrate selection logic of these premium users is to constantly request each segment at the bitrate to be guaranteed, e.g., a HD version. Note that regular users who receive video services in a best-effort fashion may coexist with premium users in practice. There is no special treatment of their resource allocation and thus no guarantee for their playback performance.

The operations of the streaming system proceed as follows. Initially, the video server broadcasts the MPD so that the downlink scheduler (within the RAN) and the clients will be aware of the available video bitrates versions. For each switching period that equals to the segment length, the premium users request a segment at the bitrate to be guaranteed based on their service priority. Unlike conventional downlink resource allocation that is oblivious to these segment requests, the network-assisted GESH system will perform service-aware resource allocation based on the requested bitrate, the low-layer channel status, and the high-layer playback information. While the bitrate request and instant channel state are available to the RAN by default in current standard, client playback information, such as buffer status, needs to be embedded in additional periodic feedback from clients. This is feasible as 3GPP has standardized the quality metrics reporting process for clients and uses HTTP POST as the reporting protocol [31]. Note that the latency incurred by this buffer status reporting process is minimal and can be ignored in model and algorithm design. In fact, only a few bytes of data are sufficient to contain the buffer occupancy information and thereby a negligible delay will be introduced in a LTE environment with a $\sim$5 Mbps upload speed. In this way, the premium users are able to enjoy the video with guaranteed playback performance while neither the current infrastructure of PFS in cellular edge networks and the standard request-response framework of HTTP streaming needs to be modified. Moreover, GESH does not impact the statelessness at the backend video servers and preserves the system scalability.

B. Weighted Proportional Fair Scheduling

In this section, we introduce the proposed weighted proportional fair scheduling framework for GESH, which enables the service-aware resource allocation without modifying existing PFS infrastructure.

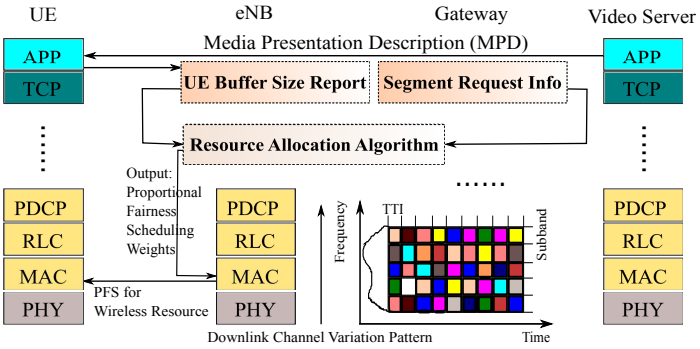


Fig. 2. The weighted proportional fair scheduling framework in GESH.

In LTE networks, time-frequency radio resources are divided into resource blocks (RBs) as shown in Fig. 2. We first consider the general PFS algorithm that computes a scheduling metric ρ of the lth RB at the upcoming Transmission Time Interval (TTI) $t+1$ for user i, i.e.,

$$\rho_{il}(t+1) = \omega_i \frac{c_{il}(t+1)}{R_i(t)} \tag{1}$$

where ω_i is the weight for each user indicating its priority, $c_{il}(t+1)$ is the instant channel rate of user i under the lth RB at TTI $t+1$, and $R_i(t)$ is the smoothed throughput of the user computed until current TTI t. The smoothed throughput for user i can be computed as,

$$R_i(t) = \alpha R_i(t-1) + (1-\alpha)\sum_{l=1}^{L} \delta_{il}(t) c_{il}(t) \tag{2}$$

where $R_i(t-1)$ is the computed throughput at TTI $t-1$, L is the total number of RB in one TTI, $\delta_{il}(t)$ (equal to 0 or 1) indicates whether or not the lth RB is assigned to user i at current TTI t, and α is the smoothing factor that balances the past throughput and the newly available channel rate. The algorithm will then assign the given RB l at TTI $t+1$ to the user achieving the maximum scheduling metric, i.e.,

$$\arg\max_i \ \rho_{il}(t+1) \tag{3}$$

It then updates $\rho_{il}(\cdot)$ continuously and repeats the RB assignment for all RBs at each TTI. It has been proved that such a PFS algorithm will asymptotically optimize the long-term log utility of throughput [32], i.e.,

$$\max \sum_i \omega_i \log R_i \tag{4}$$

and accomplish asymptotic throughput fairness, i.e.,

$$\frac{R_i}{R_h} = \frac{\omega_i}{\omega_h} \tag{5}$$

where $h \neq i$ represents a different user from i. In classic PFS, ω_i is set to be equal to ω_h, implying the same priority for all users. Therefore, the long-term throughput of all users are asymptotically equal and fair.

The proposed weighted proportional fair scheduling framework shown in Fig. 2 is built upon the above analysis. Before a

segment request, the input parameters of PFS, i.e., the weights ω_i of each user, are dynamically adjusted in GESH system in order to guide the resource allocation during this segment period. This will ultimately achieve a desired throughput that is proportional to the weight and satisfy the requested bitrate for each user. Specifically, the weights should be jointly determined by the current buffer occupancy, segment request, and channel condition, i.e.,

$$\frac{R_i}{R_h} = \frac{\omega_i}{\omega_h} = f(\mathbf{r}, \mathbf{B}, \mathbf{C}) \tag{6}$$

where $\mathbf{r}$, $\mathbf{B}$ and $\mathbf{C}$ are the vector of requested video bitrate, buffer size in terms of seconds, and per-segment channel rate for all users. Since GESH adjusts the weights to guarantee the playback for every segment request, the throughput R_i in (6) is re-defined as the throughput within one segment period.

C. Channel Modeling

Since the goal of GESH is to guarantee each segment request by proper downlink resource allocation, it is imperative to understand how the channel conditions vary from one segment period to the other. Hence, we now model the per-segment channel variation. In particular, we model the *per-segment channel rate* $C_i \in \mathbf{C}$ when user i occupies all radio resources during this segment. This way, the resource allocation algorithm can decide how many percents of all segment resources should be allocated to user i to meet its throughput demand in one segment. Such a per-segment channel rate differs for each user. The channel modeling is proceeded by a data-driven method using large-scale traces.

We build the LTE cell using a 3GPP standard-compliant LTE simulation environment [33] whose implementation and setup will be detailed in Section V. In a given LTE cell with N users, we aim at modeling the transition of per-segment channel rate for an arbitrary user. In other words, this modeling characterizes the overall channel condition of this N-user cell. In one run, we measure the channel rate of a user when she occupies all the resource blocks during a segment period T, and we collect this per-segment channel rate for all N users. Each run lasts for 600 seconds and generates $N\frac{600}{T}$ samples. We then repeat this for 100 runs, wherein the users are randomly distributed over the cell in each run and thereby the channel conditions of a given user (e.g., user #2) are different among the 100 runs. Under this given LTE environment with N users, we can eventually collect a total of $\frac{N\times 600\times 100}{T}$ samples for the channel modeling.

We propose to model the per-segment channel rate variation as a Markov model because cellular channels have been widely modeled as Markov models and such a modeling has been proven to be effective [34]. We divide the range of per-segment channel rate into M states and each state is represented by the median value of its rate range. For a particular per-segment channel rate transition (i.e., from one state to another state), we can obtain the transition probability by dividing the number of such state transitions over the total number of all transitions with the same starting state. As a result, we can obtain a transition probability matrix $\mathbf{P} \in \mathbb{R}^{M\times M}$ at $M \times M$ dimensions, which characterizes the per-segment channel variations for an arbitrary user in such a LTE cell. By repeating the above steps for other LTE cell environment, i.e., with different number of users, we can reach a set of transition probability matrices under typical LTE cell configurations. An example 3D-plot of the transition probability matrix under a 10-premium-user LTE cell is shown in Fig. 3. It is observed from the figure that the per-segment channel rate of a user is more likely to keep unchanged or transit to a neighboring channel rate.

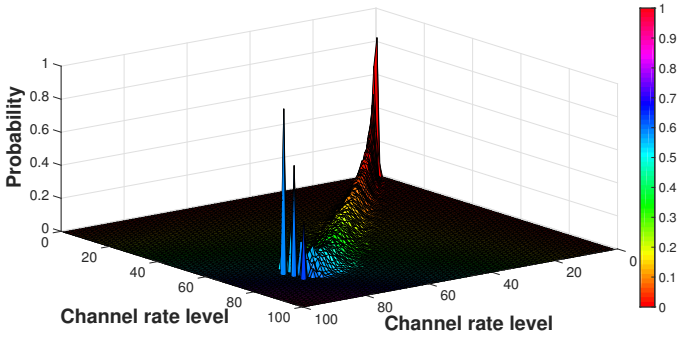


Fig. 3. Transition probability matrix of the per-segment channel rate in a 10-premium-user LTE cell.

Given the per-segment channel rate of a user $C_{i,u}$ during the current segment and the transition probability matrix, we can now easily derive the expected per-segment channel rate during the upcoming segment C_i over all the possible rate levels, i.e.,

$$C_i = \sum_v p(u,v) C_{i,v} \tag{7}$$

where u is the index of current rate state determined by $C_{i,u}$, $C_{i,v}$ is the representative median rate of a possible next state v, and $p(u,v)$ is the transition probability in the matrix $\mathbf{P}$.

IV. Service Aware Resource Allocation

A. Problem Formulation

In order to achieve the desired proportional throughput in (6) and satisfy the requested bitrate, we need to deliberately configure the weights before every segment requests. Specifically, we need to set the ratio between the weights of two users as the ratio between their estimated throughput during the upcoming segment, i.e.,

$$\frac{\omega_i}{\omega_h} = \frac{\frac{n_i}{K} R_{i,seg}}{\frac{n_h}{K} R_{h,seg}} \tag{8}$$

where n_i is the number of RBs allocated to user i in the upcoming segment, K is the total number of available RBs during one segment, $\frac{n_i}{K}$ represents the resource share of user i within the upcoming segment, $R_{i,seg} = g(C_i)$ is the estimated per-segment throughput mapped from the per-segment channel rate assuming user i takes all RBs of the entire segment, and $g(\cdot)$ is a mapping function that maps the physical-layer channel rate to the application-layer rate, accounting for the higher-layer packet overhead, and it will be defined in Section V. In a given system, K is a fixed number and the exact partition of RBs in our evaluation will be introduced in Section V. Note

that for different segment periods, the maximum channel rate of the same RB may be different.

To derive the weights, we first seek the optimal resource share in the upcoming segment, i.e., the optimal number of RBs assigned to each user n_i, by considering the following constraints. First, a sufficient amount of video data should be downloaded to guarantee the continuous playback of the requested video. In other words, the total duration of the streamed video and the buffered video should be longer than the segment length T. On the other hand, the buffered data should not exceed the maximum buffer size of the video client in order to avoid buffer overflow. Otherwise, the newly downloaded data would have to be discarded due to the lack of storage space, which would consequently cause incomplete playback or playback artifact as well as wasting network bandwidth. Furthermore, the network provider will only assign a limited bandwidth to this video service. If the total resource share of all users is larger than the dedicated capacity, it may conflict with the operator's bandwidth management policy.

Since efficient utilization of radio resources can benefit the entire streaming ecosystem, the objective of the resource allocation is to maximize the total throughput of users under the limited bandwidth. Building on the above insights, we proceed by formulating the SERVICE-AWARE RESOURCE ALLOCATION PROBLEM (SRAP)

Definition 1 (SRAP): Suppose an LTE cell has N premium users that require guaranteed video service, each user with maximum buffer size $B_{\max}$, a transition probability matrix of per-segment channel rate $\mathbf{P}$, and its current per-segment channel rate. Given videos divided into T-second segments and requested by user i at the bitrate r_i, as well as a total of K resource blocks, the problem is to determine the number of allocated RB n_i for user i such that the video throughput of the cell is maximized without client buffer over/underflow or exceeding the bandwidth limit for this video service.

Mathematically, we can formulate SRAP as,

$$\begin{aligned} \max \quad & \textstyle\sum_{i=1}^{N} \frac{n_i R_{i,seg} T}{K} \\ \text{s. t.} \quad & R_{i,seg} = g(\mathbf{I}\mathbf{P}\mathbf{C_{rep}}^{\mathrm{T}}) \\ & \frac{n_i R_{i,seg} T}{r_i K} + B_i \le B_{\max} \\ & \frac{n_i R_{i,seg} T}{r_i K} + B_i \ge T \\ & \textstyle\sum_{i=1}^{N} n_i \le K \end{aligned} \tag{9}$$

where $\mathbf{I}$ is a indicator vector that implies the state of the current per-segment channel rate $C_{i,cur}$ of user i and $\mathbf{C_{rep}}$ is a vector of the representative median rate of M states. The first constraint defines the computation of the estimated throughput of user i when taking the RBs of the entire segment. The second and third constraints indicate the buffer level should always be a proper value to avoid buffer underflow and overflow. The final constraint bounds the RB assignment under the given bandwidth.

The adaptation wisdom behind (9) is that more RBs are generally allocated to those users who currently have a smaller buffer occupancy and a worse channel condition, while excessive RB assignment that may cause buffer overflow shall also be avoided. Consequently, we can not only guarantee the playback, but also enhance the fairness among these users.

Note that the above formulation inherently considers non-premium users. For a non-premium user with best-effort non-guaranteed playback, the third constraint of buffer underflow will be set to a special case, i.e., $\frac{n_i R_{i,seg} T}{r_i K} + B_i \ge 0$. This indicates that the playback of non-premium users can be stalled (empty buffer). During the resource allocation, RBs will be assigned to premium users first and the remaining RBs will be allocated to non-premium users in a best-effort way. To adjust the premium/non-premium ratio, the number of premium/non-premium users considered in the optimization can be changed accordingly.

B. Optimal Dynamic Programming Algorithm

We now propose a dynamic programming algorithm to optimally solve the SRAP in (9). The system running this algorithm will serve as the performance bounds. We will compare different benchmarks against the performance bounds in Section V. We first consider the subproblem of SRAP with $N_s (N_s < N)$ video users and $K_s (K_s < K)$ RBs dedicated to this video service. Let $U[N_s][K_s]$ denote the maximum utility, i.e., the maximum throughput for N_s users under K_s RBs. The optimal $U[N_s][K_s]$ of the subproblem should be the summation of the optimal throughput for $N_s - 1$ users and the throughput of the N_sth user. Thereby, we have the following iterative formula.

$$\begin{aligned} U[N_s][K_s] = \max\Bigg\{ & U[N_s - 1][K_s - m] + \frac{m R_{N_s,seg} T}{K_s} \;\Bigg| \\ & \text{where } m \le K_s, n_{N_s,\min} \le m \le n_{N_s,\max} \Bigg\} \end{aligned} \tag{10}$$

where m is the number of RBs allocated to the N_sth user, $\frac{m}{K_s}$ represents the percentage of RBs allocated to the user, $R_{N_s,seg}$ is the per-segment throughput of the N_sth user when it uses up all resources in the segment, and $n_{N_s,\min}$ and $n_{N_s,\max}$ are the lower bound and upper bound of m, respectively. The two bounds are derived from the second (buffer overflow) and third (buffer underflow) constraints in (9), i.e.,

$$\begin{aligned} n_{i,\min} &= \frac{(T - B_i) r_i K}{R_{i,seg} T}, \ \forall i \\ n_{i,\max} &= \frac{(B_{\max} - B_i) r_i K}{R_{i,seg} T}, \ \forall i \end{aligned} \tag{11}$$

Note that $n_{i,\min}$ is always set to be 0 for non-premium users.

By substituting $N_s = N$ and $K_s = K$, we can iteratively solve the SRAP in (9) starting from a single user. The optimal dynamic programming algorithm is summarized in Algorithm 1. We first calculate the optimal utilities and allocations when there is only one user and the total number of RBs ranges from K to zero. We then repeat this procedure by increasing the number of users from 1 to N. That way, two 2D matrices recording the optimal utilities and allocations for different number of users and RBs can be obtained. Finally, we can go through the matrices to identify the optimal solution that generates the maximum utility. By substituting n in (8) with the optimal outputs, we can eventually configure the optimal weights of each user in the upcoming segment and achieve desired amount of guaranteed throughput in GESH.

Algorithm 1 Dynamic Programming Allocation Algorithm

Compute $R_{i,seg}$, $n_{i,\max}$, $n_{i,\min}$, $\forall i$, according to (9) and (11).
$U[0][k] \leftarrow 0$, $\forall k$
for $i = 1$ to N **do**
  for $k = K$ to 0 **do**
    $U[i][k] \leftarrow \max_m\{U[i-1][k-m]+\frac{mR_{i,seg}T}{k} \mid n_{i,\min} \leq m \leq n_{i,\max}, m \leq k\}$
    $n^*[i][k] \leftarrow \arg\max_m\{U[i-1][k-m] + \frac{mR_{i,seg}T}{k} \mid n_{i,\min} \leq m \leq n_{i,\max}, m \leq k\}$
  end for
end for
Backtrace the 2-D matrix $\mathbf{n}^*[\cdot][\cdot]$ to obtain optimal number of allocated RBs for each user that yields $U[N][K]$

Since the initialization of Algorithm 1 takes $\mathcal{O}(N)$ time and we have NK sub-instances that each consumes $\mathcal{O}(K)$ time, we can arrive at the following proposition regarding the time complexity of the algorithm.

Proposition 1: The proposed optimal dynamic programming allocation algorithm (Algorithm 1) can solve SRAP in $\mathcal{O}(NK^2)$.

C. Optimal Greedy Algorithm

As K can be a huge number, Algorithm 1 may need excessive execution time, impairing the real-time requirement of resource allocation. In this section, we propose an optimal greedy algorithm to efficiently solve for SRAP. The basic idea is that, after meeting the minimum allocation requirement of each user, the user with highest $R_{i,seg}$ will obtain its maximum possible RBs and this process repeats until all the remaining RBs are assigned.

The proposed algorithm is summarized in Algorithm 2. At the initialization stage, the users are sorted based on their estimated throughput when taking up the entire radio resources. That is, the sorted user index i satisfies $R_{1,seg} \geq \cdots \geq R_{i,seg} \geq R_{i+1,seg} \geq \cdots \geq R_{N,seg}$. Furthermore, the lower bounds and upper bounds of allocated RB are computed. We then initialize the number of allocated RBs as the lower bound and update the remaining bandwidth. The algorithm then loops from user 1 to N and at each iteration we assign as many as possible of the remaining RBs to the given user.

In terms of time complexity, the proposed algorithm takes $\mathcal{O}(N)$ to initialize the parameters and spends $\mathcal{O}(N \log N)$ to sort the user index. Due to the constant number of operations within each iteration of the algorithmic loop, we can reach the following proposition for Algorithm 2.

Proposition 2: The proposed optimal greedy allocation algorithm (Algorithm 2) can solve SRAP in $\mathcal{O}(N \log N)$.

Therefore, Algorithm 2 is much more efficient than the Algorithm 1 and is preferred in practice.

To prove the optimality of Algorithm 2, we have scrutinized the SRAP problem in (9) and arrived at two important conclusions in the following.

Lemma 1: If $\sum_{i=1}^{N} n_{i,\max} \geq K$, all optimal solutions will take up the entire radio resources exactly.

Algorithm 2 Optimal Greedy Allocation Algorithm

Sort the user index such that $R_{1,seg} \geq \cdots \geq R_{i,seg} \geq R_{i+1,seg} \geq \cdots \geq R_{N,seg}$
Compute $n_{i,\max}$, $n_{i,\min}$, $\forall i$ according to (11)
$K_{rem} \leftarrow K - \sum_{i=1}^{N} n_{i,\min}$
for $i = 1$ to N **do**
  if $K_{rem} \geq n_{i,\max} - n_{i,\min}$ **then**
    $n_i^* \leftarrow n_{i,\max}$
    $K_{rem} = K_{rem} - (n_{i,\max} - n_{i,\min})$
  else
    $n_i^* \leftarrow n_{i,\min} + K_{rem}$
    $K_{rem} = 0$
  end if
end for
Return $\mathbf{n}^*$ ▷ user index is sorted

Proof: Assume there exists an optimal solution $\mathbf{x}$ such that $\sum_{i=1}^{N} x_i < K$. Since $\sum_{i=1}^{N} n_{i,\max} \geq K$, we can find those users satisfying $x_i < n_{i,\max}$ and assign them the unused RBs. Thereby, the throughput utility of the new assignment will always be increased, which contradicts the optimality assumption of $\mathbf{x}$. ■

Lemma 2: Denote $\mathbf{n}^* = \{n_1^*, \cdots, n_N^*\}$ as the solution generated by the proposed greedy algorithm in Algorithm 2. Let j be the first index such that $n_j^* \neq n_{j,\max}$, i.e.,

$$\begin{cases} n_i^* = n_{i,\max}, \ \textit{if } i \in [1, j) \\ n_i^* \in [n_{i,\min}, n_{i,\max}), \ \textit{if } i = j \\ n_i^* = n_{i,\min}, \ \textit{if } i \in (j, N] \end{cases} \tag{12}$$

If there exists an optimal solution $\mathbf{x} = \{x_1, \cdots, x_N\}$ different from $\mathbf{n}^*$ and the smallest index d such that $x_d \neq n_d^*$, we have $x_d < n_d^*$.

Proof: This lemma can be proved by considering three different cases.

1) When $d < j$, $n_d^* = n_{d,\max}$. Since $x_d \neq n_d^*$, we have $x_d < n_d^* = n_{d,\max}$.
2) When $d = j$, we assume $x_d > n_d^*$. According to (12), we have $\sum_{i=1}^{j} n_i^* + \sum_{i=j+1}^{N} n_{i,\min} = K$ and $x_i = n_i^*$ for all $1 \leq i < j$. Therefore, we can derive $\sum_{i=1}^{j} x_i + \sum_{i=j+1}^{N} n_{i,\min} > K$. However, we have $\sum_{i=1}^{N} x_i = K$ by Lemma 1, which is a contradiction. Therefore, the assumption of $x_d > n_d^*$ is wrong. Since $x_d \neq n_d^*$ holds, we can then prove $x_d < n_d^*$.
3) When $d > j$, we assume $x_d > n_d^*$. This case is similar as Case 2 and we would have the same contradiction to Lemma 1, i.e., $\sum_{i=1}^{N} x_i > K$.

By summarizing the three cases, Lemma 2 can be proved. ■

We are now ready to prove the optimality of the proposed Algorithm 2.

Theorem 1: Algorithm 2 is optimal for SRAP in (9) and can maximize the throughput.

Proof: Given $\mathbf{n}^*$, the output of Algorithm 2, we assume there exists an optimal solution $\mathbf{x}$ and a smallest index d that differs in $\mathbf{n}^*$ and $\mathbf{x}$. According to Lemma 2, we also have $x_d < n_d^*$. Suppose we increase x_d to n_d^* and decrease as many of $\{x_{d+1}, \cdots, x_N\}$ as necessary in order to keep the total number

TABLE I
EDGE NETWORKS CONFIGURATION

Parameter	Value	Parameter	Value
Channel model	Urban/Suburban Macrocell	MIMO mode	2×1
Multi-path gains	6 paths (-3 dB∼ -18 dB)	Multi-path delay spread	6 paths (0.5 μs∼2.5 μs)
Carrier frequency	2000 MHz	Inter-site distance	1000 m
eNodeB height	25 m	UE height	1 m
DL transmit power	46 dBm	Antenna gains	eNodeB: 18 dbi, UE: 0dbi

of assigned RBs unchanged. Consequently, we can obtain a new solution $\mathbf{y} = \{y_1, \cdots, y_N\}$ with $y_i = n_i^*$ for $1 \leq i \leq d$. When $i > d$, this new solution $\mathbf{y}$ satisfies:

$$\sum_{d<i\leq N} (x_i - y_i) = y_d - x_d \tag{13}$$

By replacing $\frac{T}{K} = \beta$, the total throughput utilities achieved by $\mathbf{y}$ can be derived as,

$$\begin{aligned}
\sum_{1\leq i\leq N} \frac{y_i R_{i,seg} T}{K} &= \beta \sum_{1\leq i\leq N} y_i R_{i,seg} \\
&= \beta \Bigg[\sum_{1\leq i\leq N} x_i R_{i,seg} + (y_d - x_d) R_{d,seg} \\
&\quad - \sum_{d<i\leq N} (x_i - y_i) R_{i,seg} \Bigg] \\
&\geq \beta \Bigg[\sum_{1\leq i\leq N} x_i R_{i,seg} + R_{d,seg} \Bigg((y_d - x_d) \\
&\quad - \sum_{l<i\leq N} (x_i - y_i) \Bigg) \Bigg] \\
&= \beta \sum_{1\leq i\leq N} x_i R_{i,seg}
\end{aligned} \tag{14}$$

where the inequality follows by $R_{d,seg} \geq R_{d+1,seg} \geq \cdots \geq R_{N,seg}$.

Therefore, $U(\mathbf{y}) \geq U(\mathbf{x})$, where $U(\cdot)$ represents the throughput utility of a RB allocation solution. We now discuss the two possible cases.

1) $U(\mathbf{y}) > U(\mathbf{x})$. It indicates that $\mathbf{x}$ cannot be optimal, which contradicts the assumption that $\mathbf{x}$ is an optimal RB allocation. Hence, $\mathbf{n}^*$ is an optimal solution.
2) $U(\mathbf{y}) = U(\mathbf{x})$. It implies that we made the x_d in the optimal solution equal to the n_d^* in the proposed greedy solution without loss of the overall utility. We can proceed by treating $\mathbf{y}$ as a new optimal solution that is modified from $\mathbf{x}$. We then make y_{d+1} equal to n_{d+1}^* and carry out the entire procedure in the above. After repeating this process for $n_{d+1}^*, \cdots, n_N^*$, we will either exit through the contradiction in Case 1), or eventually end up with an optimal solution that is exactly the same as $\mathbf{n}^*$, in which case $\mathbf{n}^*$ is an optimal solution.

In summary, the theorem is proved. ■

V. EVALUATION

We have built a system-level simulation environment that includes video servers, core networks, RAN networks with the proposed GESH and multiple mobile clients, based on the architecture in Fig. 1. We use the video sequences "Big Buck Bunny", "Touchdown", "Tear of Steel", and "Johnny", and encode them into H.264/AVC videos. The segment length T is 1 second and the frame rate is 24 fps. The reported results are the average over all video sequences.

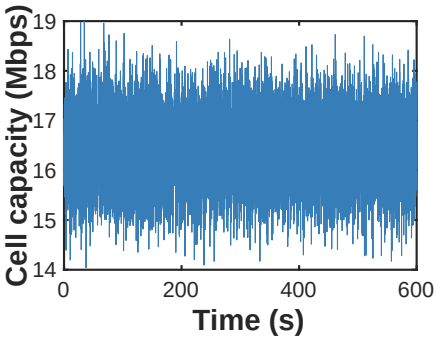


Fig. 4. Total available bandwidth of the cell versus time.

We construct a MIMO-OFDM LTE air-interface. The system bandwidth is fixed at 10 MHz and there are 10 resource units to be scheduled at each TTI. Considering that the scheduling procedure is invoked at the beginning of a TTI and each TTI lasts for 1 ms, there are a total of 10000 resource units ($K = 10000$) for a 1-second segment to be allocated. We assume the mobile users are uniformly distributed in the cell. The instant channel rate is determined by both large-scale and small-scale fading, where the propagation pathloss follows COST 231 Hata Model and the small-scale fading follows Rayleigh fading model [33]. Other detailed parameters of the LTE network are shown in Table I. Since non-premium users are a special case of premium users with $n_{i,\min} = 0$, they do not place any pressure on the resource allocation algorithms and the system capacity. Therefore, we focus the experiments on premium users only. By setting the channel rate state into $M = 150$ levels (0 to 75Mbps) and varying the number of premium users N from 10 to 30 with increments of 5, we can obtain a set of per-segment channel variation models under different cell environments using the methodology in Section III-C. We assume the premium users subscribe to the service of guaranteed bitrate at 630 Kbps and the lower layer overhead function $g(\cdot)$ is a fractional function with factor 0.9. The maximum buffer size $B_{\max}$ is set to be 12 seconds and the throughput smoothing factor α is set to be 0.7. Each of the following simulations runs for 600 seconds.

We bound the number of users at 30 because this setting can saturate the cell capacity and effectively evaluate GESH.

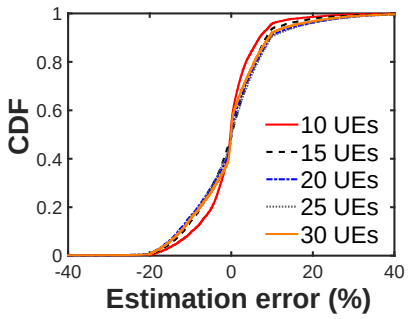


Fig. 5. CDF of channel estimation error.

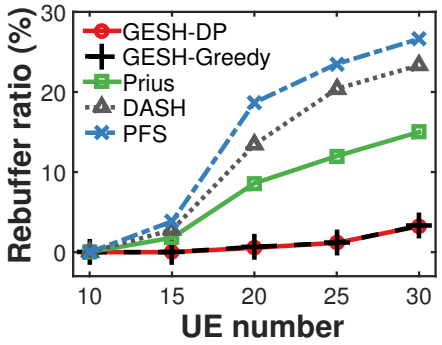


Fig. 6. Rebuffer ratio of streaming systems.

In Fig. 4, we show the total available bandwidth of the 30-UE cell (after considering the 10% overhead). It can be seen that the average total cell bandwidth is 16.41 Mbps, which is less than the total rate requirement (30×630 Kbps = 18.9 Mbps). Such a user density setting also accords with related works [19], [22] and ITU recommendation [35].

We compare the proposed optimal dynamic programming algorithm (referred as *GESH-DP*) and optimal greedy algorithm (referred as *GESH-Greedy*) to existing systems with different combinations of scheduling and streaming strategies. We first implement a conventional system using the default PFS resource allocation and HTTP streaming (referred as *PFS*). It essentially means that the input weights of users for the core PFS algorithm at the RAN are all set to be identical and the bitrate request is fixed. Furthermore, we also build a typical DASH system using the default PFS method and the basic throughput-based adaptation (referred as *DASH*), where the client selects the highest bitrate that can be supported by the smoothed per-segment throughput. Moreover, we evaluate an advanced DASH system [4] using PFS scheduling and a multi-client channel-aware adaptation (referred as *Prius*). For DASH and Prius, the source video is transcoded into 10 bitrate versions, ranging from 64 Kbps to 2000 Kbps.

A. Channel Modeling Results

We first evaluate the proposed per-segment channel variation model. In particular, we present the accuracy of the Markov model by measuring the estimation error between the estimated per-segment channel rate and the actual per-segment channel rate (obtained from periodic feedback in RAN). We show the cumulative distribution function (CDF) of the overall estimation error for all users under all requests in Fig. 5.

We learn from the results that around 83% of the estimations arrive at an estimation error less than 10%. Such an accuracy is promising for edge networks and outperforms the existing estimation methods in DASH, e.g., around 80% cases present a estimation error less than 10% using a HSDPA dataset [36]. This is attributed to the large-scale channel data (millions of data samples for one single environment) for modeling and the effectiveness of applying Markov model in cellular edge networks [34]. Therefore, the per-segment channel variation and the per-segment throughput can be approximated accurately, which further improves the efficacy of service-aware resource allocation optimization.

B. Playback Rebuffer Results

In this section, we evaluate the playback rebuffer events of GESH and the reference systems. We inspect the buffer evolution trace of all clients at each TTI. If the buffer occupancy is zero at a specific TTI, then a rebuffer event occurs. By diving the number of rebuffer events by the total number of inspection, we can derive the rebuffer ratio of a system.

We show the results under different cell environments, i.e., different number of users, in Fig. 6. It can be seen from the figure that GESH system (either GESH-DP or GESH-Greedy) outperforms the reference systems significantly when the UE number is larger than 10. At a mobile cell with less than 10 users, the system bandwidth is more than sufficient to support all users and eventually there is almost no rebuffer event for all systems. As the UE number increases, the rebuffer ratio of GESH-DP and GESH-Greedy still keeps at a relatively stable level. Even when the UE number reaches 30, the rebuffer ratio is only around 3%. This is because GESH proactively and dynamically adjusts the resource allocation based on the channel conditions and buffer occupancy. If a user is suffering a bad channel and a low buffer level, GESH will assign this user a relatively high weight that is required for the guaranteed bitrate, which allows her to obtain more RBs and to quickly boost the buffer level. In the reference systems using proportional fair scheduling, however, the resource allocation weights are equal and fixed for all users. Consequently, users with better channel conditions will generally achieve a higher throughput regardless of their buffer level. Thus those users suffering the temporary channel degradation would drain their player buffer and experience playback rebuffer.

We can also observe that the rebuffer ratio of these systems increases as the UE number increases. This is attributed to the more intense competition for system bandwidth within the cell. When the UE number reaches 30 (cell capacity saturated), the rebuffer ratio of GESH clearly starts to increase. If the UE number further increases, the GESH performance would continue to degrade since the system simply cannot support a rate demand significantly larger than its capacity.

C. Playback Smoothness Results

We proceed to evaluate the temporal variation of playback quality in mobile video streaming systems, i.e., how smooth is the video playback. We employ a metric, called playback

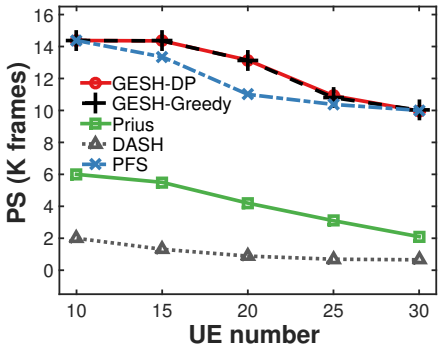


Fig. 7. Playback smoothness of streaming systems.

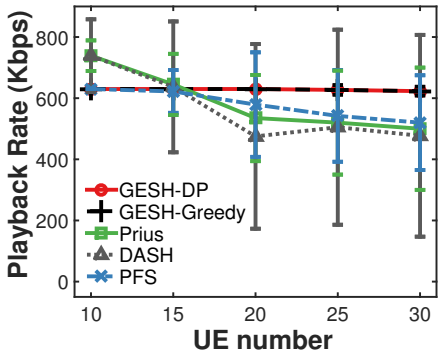


Fig. 8. Average playback rate (error bar is standard deviation) for systems.

smoothness (PS) [9], which is defined as the expected length of one playback round (in number of frames) without level change or rebuffer. Mathematically, we express PS as follows.

$$PS = \sqrt{\sum_{z=1}^{Z}(n_z^2)/Z} \tag{15}$$

where the continuous playback of one bitrate level or continuous rebuffer is defined as one round. There are a total of Z rounds and each round consists of n_z frames.

We report the UE-averaged PS under different number of users in Fig. 7. As shown in the figure, both GESH-DP and GESH-Greedy achieve a very smooth playback with a large number of frames continuously playing. This stems from the similar reason explained above. Thanks to the service-aware resource allocation and such network assistance, GESH can proactively rather than passively respond to the channel dynamics and thus keep a stable buffer. Furthermore, given that the streamed video is fixed at a bitrate in GESH, the playback smoothness is very high. On the other hand, DASH and Prius dynamically change the bitrate requests based on the channel variation, which results in an extremely instable playback with frequent bitrate variations. Prius has a slightly better performance since it aims for a fair rate adaptation for multiple users, which limits the rate variation to some extent. Although the playback of traditional PFS usually stays at a certain level for a long time, it actually spends a large portion of the time on rebuffer events. In other words, the playback of PFS is stable in a negative sense.

D. Playback Rate Results

We now evaluate the actual video playback rate averaged over multiple users and video sequences, and the standard deviation of the average playback rate. When a user is rebuffering, the playback rate is considered as zero.

It can be seen from Fig. 8 that the proposed GESH systems satisfy the 630 Kbps requirement for premier users in all cases without many fluctuations, except when the user number is 30, the playback rate is marginally below the threshold. This results from aforementioned benefits of GESH, especially the goal to guarantee the target rate. On the other hand, best-effort driven reference systems cannot always support the playback rate requirement due to their failure to provide service-aware network resource allocation. Furthermore, the reference systems show significant fluctuations in the playback rate. This is because of the rate adaptation in DASH and Prius, and the frequent buffer underflow in PFS.

E. Example Performance of Specific Users

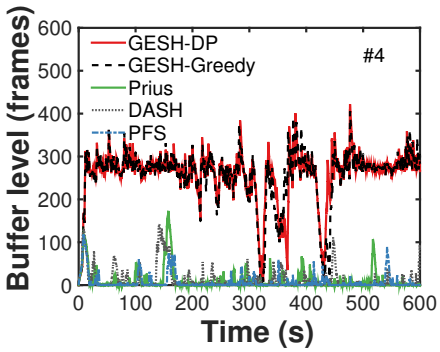
We have evaluated the playback performance of the entire mobile streaming system. In this section, we present the example performance of certain specific users in order to illustrate the effectiveness of GESH. We show the buffer evolution and the corresponding evolution of the input weights to the resource allocation of two users in Fig. 9. In such a 30-UE environment, user #4 has a dynamic and generally worse channel whereas user #8 enjoys a stable channel condition.

We can observe that user #4 in GESH systems (both GESH-DP and GESH-Greedy) achieves a considerably better performance than that in Prius, DASH or PFS. Especially, when the channel condition fluctuates and the buffer level decreases suddenly for three times during the 300th second to the 450th second (Fig. 9a), the weights of GESH-DP and GESH-Greedy always quickly boost up at the corresponding moments (Fig. 9c) in order to assign more RBs to user #4 and thus increase its buffer level. In contrast, Prius, DASH and PFS without service-aware resource allocation apply an equal weight for all users. Therefore, user #4 cannot obtain resource priority when the channel rate is mitigated, which leads to the frequent rebuffer shown in Fig. 9a.

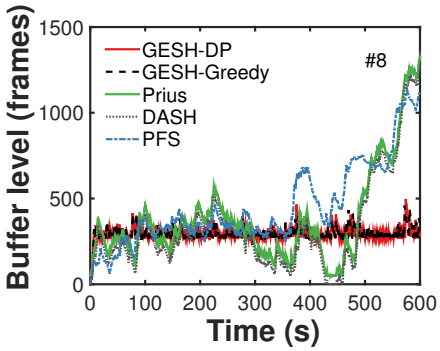
User # 8 who enjoys a good channel condition accomplishes a satisfactory playback in general. The only playback rebuffer occurs in the DASH system. This is because the bitrate is dynamically adjusted based on the estimated throughput. Due to the throughput overestimation problem in the highly dynamic cellular channel, the player would suffer buffer underflow. Prius does not observe this effect since it generally applies a more conservative rate adaptation for user fairness.

F. Execution Time Results

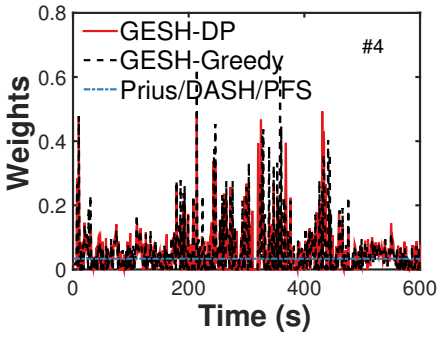
We have shown that GESH-DP and GESH-Greedy achieve a similarly satisfactory performance in different cell environment. In this section, we compare the execution time complexity of the two systems in order to validate the theoretical analysis in Proposition 1 and 2.



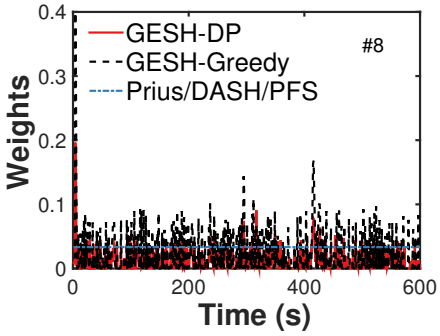
(a)



(b)



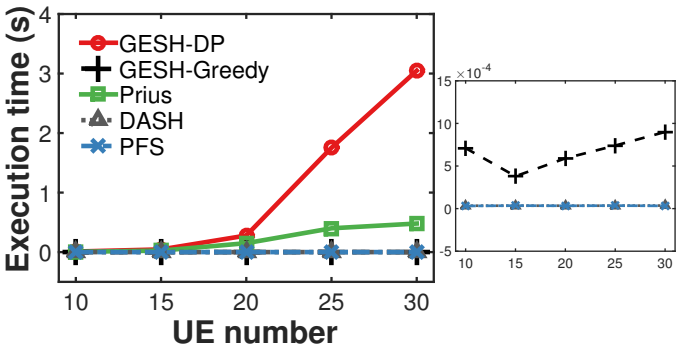
(c)



(d)

Fig. 9. Evolutions for the bad-channel user's (#4) a) buffer level and c) allocation weights, as well as the good-channel user's (#8) b) buffer level and d) allocation weights.

We measure the actual elapsed time of different algorithms for deriving the weights using a Intel Core i7 machine with 2.1 GHz CPU and 8 GB RAM. It can be seen that the execution time of GESH-DP increases significantly with the number of users in the cell. In the end, the more-than-3-second execution time in 30-UE case would make the algorithm unlikely to apply in real-time video streaming systems, which requires the algorithm to be executed at least within the segment period (i.e., 1 second in our experiments). Fortunately, the proposed GESH-Greedy algorithm achieves a stable and negligible time complexity, ranging from 380 μs to 900 μs. This demonstrates the advantages of time complexity for GESH-Greedy. Considering more powerful computing in actual schedulers, real-time performance can be achieved easily. Therefore, we can conclude that GESH-Greedy is preferred to be implemented in the real-world mobile streaming systems.



(a) (b)

Fig. 10. Execution time of a) resource allocation algorithms; b) zoomed in view for GESH-Greedy, DASH, PFS.

TABLE II
REBUFFER RATIO OF 30 UE FOR DIFFERENT VIDEO CONTENT

	Bunny	Touchdown	Tear	Johnny
GESH-DP	3.02%	4.12%	3.71%	2.15%
GESH-Greedy	3.06%	4.21%	3.74%	2.15%
Prius	14.98%	18.95%	16.62%	9.45%
DASH	21.34%	27.54%	26.46%	18.02%
PFS	22.64%	30.26%	27.68%	17.98%

G. Impacts of Video Content

We now proceed to evaluate the impacts of video content on playback performance. We present the example performance of rebuffer ratio under the 30-UE case. As shown in Table II, "Touchdown" and "Tear" have a generally higher rebuffer ratio, compared to the other two video sequences under all reference systems. This is because, as a sports video and a sci-fi movie, respectively, "Touchdown" and "Tear" have more motions than the other two videos which are a cartoon and news video, respectively. More motions imply more fluctuations in the temporal characteristics of the video and that more video segments may have an actual bitrate higher than the average bitrate used by the streaming algorithms. In this case, it is more likely for the streaming algorithms to underestimate the bitrate requirement and allocate insufficient network resources, which leads to rebuffer and performance degradation. On the other hand, if the video content is relatively stable, such as the news video "Johnny", it is easier for the streaming systems to satisfy the required playback rate.

VI. Discussion

Channel Models. To enhance the accuracy of the per-segment channel model, one may introduce more channel states and address a tradeoff between the accuracy and the computation of offline modeling. Besides, designing a online updating process of the model parameters might enable a quick capture of the channel variation. However, the proposed per-segment channel models has already shown robust performance under different user environments. We can also easily obtain the channel models under other number of users by adopting a similar modeling methodology.

Network Assistance. In this paper, we have shown the performance gains of network assistance in terms of service-aware downlink resource allocation. It is expected that other types of network assistance can further improve the video streaming performance over edge networks. For example, network-side rate adaptation on top of the proposed weighted PFS scheme can guide the traditional client-driven bitrate selection. Furthermore, moving the Markov channel models that are trained within the RAN to the clients can provide an more accurate scheme for bandwidth estimation in traditional client-driven DASH. Some full-scale investigation will be needed to justify the feasibility of these network-assisted designs.

Guaranteed Playback. Note that when the number of users in a cell is considerably large, the system bandwidth may be insufficient to guarantee the playback of all users. GESH would then attempt to guarantee the playback of as many users as possible. It is important to reiterate that the key innovation of GESH is to proactively guarantee the playback performance under a reasonable traffic load, which significantly outperforms existing best-effort systems that passively react to the channel dynamics.

Business Incentives. There are strong incentives for different parties involved to adopt the GESH framework. Users who require better and smoother video experience are now able to enjoy such an experience via content providers through subscription, membership, direct payment, etc. Similarly, content providers can receive financial benefits from premier users and attract more diverse users via quality-differentiated services. Finally, network providers can also obtain significant profit by reaching an agreement with content providers and providing them with better network resources than regular best-effort Internet traffic. Recently, the network neutrality law has been repealed in the U.S., which removes the legal barrier for services such as GESH using network differentiation. A real-world deal for service-aware resource allocation has been completed between Netflix and Comcast [37].

VII. Conclusion

In this paper, we have presented a new investigation of guaranteed video playback in mobile edge networks. The proposed GESH framework utilizes the network-assisted service-aware resource allocation to proactively guarantee rather than passively improving the playback performance in the clients. GESH adopts the weighted proportional fair scheduling framework without modifying current cellular infrastructure. At the same time, the input weights to the core PFS algorithm can be dynamically optimized based on the segment request, buffer status, and channel conditions. The optimization framework is empowered by a set of optimal algorithms. We conclude from the extensive evaluations that the proposed greedy algorithm can efficiently seek the optimal resource allocation with negligible time complexity and can achieve a guaranteed video playback which significantly outperforms the conventional HTTP streaming and DASH systems.

We would like to emphasize that the proposed GESH framework can be applied into virtually any HTTP-segment-based video delivery applications, e.g., traditional video on-demand delivery or the relatively new multi-camera streaming. Future work shall be focused on including human perception driven or crowdsourced QoE model into the mobile streaming systems, which will connect the system metrics to real-world user experience directly. Furthermore, detailed investigation on the impact of content dynamics on streaming performance can be conducted as future work. If content related parameters such as encoding method, content type, temporal/spatial complexity can be identified as critical factors, content-aware scheduling can be designed on top of GESH to accommodate the content dynamics in addition to the channel dynamics.

References

[1] Cisco, "Cisco visual networking index, 2014-2019," http://goo.gl/v8pMqV.

[2] Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A. Begen, and D. Oran, "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.

[3] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with festive," in *ACM International Conference on Emerging Networking Experiments and Technologies (CoNEXT)*, Nice, France, Sep. 2012, pp. 97–108.

[4] Z. Yan, J. Xue, and C. W. Chen, "Prius: hybrid edge cloud and client adaptation for HTTP adaptive streaming in cellular networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, pp. 209–222, Jan. 2017.

[5] S. Akhshabi, L. Anantakrishnan, A. C. Begen, and C. Dovrolis, "What happens when HTTP adaptive streaming players compete for bandwidth?" in *ACM Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, Toronto, Canada, Jun. 2012, pp. 9–14.

[6] R. Kwan, C. Leung, and J. Zhang, "Proportional fair multiuser scheduling in LTE," *IEEE Signal Process. Lett.*, vol. 16, pp. 461–464, Jun. 2009.

[7] I. Sodagar, "The MPEG-DASH standard for multimedia streaming over the Internet," *IEEE Multimedia*, vol. 18, pp. 62–67, Apr. 2011.

[8] T. Stockhammer, "Dynamic adaptive streaming over HTTP: standards and design principles," in *Proc. of the second annual ACM conference on Multimedia systems (MMSys'11)*, San Jose, USA, Feb. 2011, pp. 133–144.

[9] S. Xiang, L. Cai, and J. Pan, "Adaptive scalable video streaming in wireless networks," in *ACM conference on Multimedia systems (MMSys)*, Chapel Hill, USA, Feb. 2012, pp. 167–172.

[10] C. Müller, S. Lederer, and C. Timmerer, "An evaluation of dynamic adaptive streaming over HTTP in vehicular environments," in *ACM 4th Workshop on Mobile Video (MoVid)*, Chapel Hill, NC, Feb. 2012, pp. 37–42.

[11] C. Liu, I. Bouazizi, and M. Gabbouj, "Rate adaptation for adaptive HTTP streaming," in *ACM conference on Multimedia systems (MMSys)*, Chapel Hill, USA, Feb. 2012, pp. 169–174.

[12] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hobfeld, and P. Tran-Gia, "A survey on quality of experience of HTTP adaptive streaming," *IEEE Commun. Surveys Tuts.*, vol. 17, pp. 469–492, Firstquarter 2015.

[13] R. Houdaille and S. Gouache, "Shaping HTTP adaptive streams for a better user experience," in *ACM conference on Multimedia systems (MMSys)*, Chapel Hill, USA, Feb. 2012, pp. 1–9.

[14] A. Giladi and T. Stockhammer, "Descriptions of core experiments on DASH amendment," MPEG N14619, Jul. 2014, Available: http://goo.gl/NJjnAz.

[15] *Improved Support for Dynamic Adaptive Streaming over HTTP in 3GPP*, Std. 3GPP TR 26.938, 2014, Available: http://www.3gpp.org/DynaReport/26938.htm.

[16] S. Akhshabi, L. Anantakrishnan, C. Dovrolis, and A. Begen, "Server-based traffic shaping for stabilizing oscillating adaptive streaming players," in *ACM Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, Oslo, Norway, Feb. 2013, pp. 19–24.

[17] Z. Yan, J. Xue, and C. W. Chen, "QoE continuum driven HTTP adaptive streaming over multi-client wireless networks," in *IEEE International Conference on Multimedia and Expo (ICME)*, Chengdu, China, Jul. 2014, pp. 1–6.

[18] D. De Vleeschauwer, H. Viswanathan, A. Beck, S. Benno, G. Li, and R. Miller, "Optimization of HTTP adaptive streaming over mobile cellular networks," in *IEEE INFOCOM*, Turin, Italy, Apr. 2013, pp. 898–997.

[19] A. Essaili, D. Schroeder, E. Steinbach, D. Staehle, and M. Shehada, "QoE-based traffic and resource management for adaptive HTTP video delivery in LTE," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, pp. 988–1001, Jun. 2015.

[20] Z. Yan, C. Westphal, X. Wang, and C. W. Chen, "Service provisioning and profit maximization in network-assisted adaptive HTTP streaming," in *IEEE International Conference on Image Processing (ICIP)*, Sep. 2015, pp. 2786–2790.

[21] Z. Yan, C. W. Chen, and B. Liu, "Admission control for wireless adaptive HTTP streaming: An evidence theory based approach," in *ACM International Conference on Multimedia*, Orlando, USA, Nov. 2014, pp. 893–896.

[22] J. Chen, R. Mahindra, M. A. Khojastepour, S. Rangarajan, and M. Chiang, "A scheduling framework for adaptive video delivery over cellular networks," in *ACM International Conference on Mobile Computing & Networking (MobiCom)*, Miami, USA, Sep. 2013, pp. 389–400.

[23] M. Zhao, X. Gong, J. Liang, W. Wang, X. Que, and S. Cheng, "QoE-driven cross-layer optimization for wireless dynamic adaptive streaming of scalable videos over HTTP," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, pp. 451–465, Mar. 2015.

[24] T. Bu, L. Li, and R. Ramjee, "Generalized proportional fair scheduling in third generation wireless data networks," in *Proc. IEEE International Conference on Computer Communications (INFOCOM)*, Barcelona, Spain, Apr. 2006, pp. 1–12.

[25] C. Westphal, "Monitoring proportional fairness in CDMA2000® high data rate networks," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM)*, Dec. 2004, pp. 3866–3871.

[26] 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, pp. 288–296, Jan. 2009.

[27] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, P. Whiting, and R. Vijayakumar, "Providing quality of service over a shared wireless link," *IEEE Commun. Mag.*, vol. 39, pp. 150–154, Feb. 2001.

[28] B. Sadiq, J. B. Seung, and G. D. Veciana, "Delay-optimal opportunistic scheduling and approximations: The log rule," *IEEE/ACM Trans. Netw.*, vol. 19, pp. 405–418, Apr. 2011.

[29] L. He and G. Liu, "Quality-driven cross-layer design for H.264/AVC video transmission over OFDMA system," *IEEE Trans. Wireless Commun.*, vol. 13, pp. 6768–6782, Nov. 2014.

[30] Y. Wang, D. Lin, C. Li, J. Zhang, P. Liu, C. Hu, and G. Zhang, "Application driven network: providing on-demand services for applications," in *Proc. ACM SIGCOMM Conference*, Aug. 2016, pp. 617–618.

[31] *Progressive download and dynamic adaptive streaming over HTTP*, Std. 3GPP TS 26.247 V12.1.0, 2013, Available: http://goo.gl/4EJbvd.

[32] F. Kelly, "Charging and rate control for elastic traffic," *European Transactions on Telecommunications*, vol. 8, pp. 33–37, Jun. 1997.

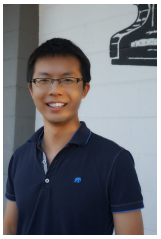
[33] *Spatial channel model for Multiple Input Multiple Output (MIMO) simulations*, 3GPP Std. TR 25.996 V12.0.0, 2012.

[34] H. S. Wang and N. Moayeri, "Finite-state markov channel-a useful model for radio communication channels," *IEEE Trans. Veh. Technol.*, vol. 44, no. 1, pp. 163–583, 1995.

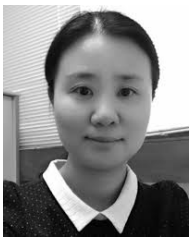
[35] *Compatibility studies in relation to Resolution 224*, ITU-R M.2241, 2011.

[36] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over HTTP," in *ACM SIGCOMM*, London, UK, Aug. 2015, pp. 325–338.

[37] Netflix to Pay Comcast for Smoother Streaming, 2014, Available: https://goo.gl/yixGd6



Zhisheng Yan received his B.S. and M.S. degrees from Shandong University and University of Science and Technology of China in 2010 and 2013, respectively, and his Ph.D. degree in Computer Science and Engineering from State University of New York at Buffalo. He is an Assistant Professor at Department of Computer Science, Georgia State University. His research interests lie in the broad area of Internet of Things. Currently, his research is focused on content delivery systems, multimedia processing and computer vision, and mobile and distributed sensing.



Miao Zhao (M'10) received the B.E. and M.E. degrees from the Department of Electronic and Information Engineering, Huazhong University of Science and Technology, Wuhan, China, and the Ph.D. degree from Electrical and Computer Engineering Department, State University of New York, Stony Brook, NY, USA.

She has been an Assistant Professor with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, since 2016. Her current research interests include data mining, recommender systems, and multimodal deep learning for various applications.

Cedric Westphal is a Principal Research Architect with Huawei Innovations working on future network architectures, both for wired and wireless networks. He also has been an adjunct assistant professor with the University of California, Santa Cruz since 2009. He received a MSEE in 1995 from Ecole Centrale Paris, and a MS (1995) and Ph.D. (2000) in EE from the University of California, Los Angeles. Cedric Westphal has authored and co-authored over eighty journal and conference papers, including several best paper awards; and been awarded over thirty patents. He was area editor for the ACM/IEEE Transactions on Networking (2009-2013), an assistant editor for (Elsevier) Computer Networks journal, and a guest editor for Ad Hoc Networks journal, JSAC, etc. He has served as a reviewer for the NSF, GENI, the EU FP7, and other funding agencies; he has co-chaired the program committee of several conferences, including IEEE ICC (NGN symposium), IEEE NFV-SDN, and he was the general chair for IEEE Infocom 2016.

Chang Wen Chen (F'04) received his BS from University of Science and Technology of China in 1983, MSEE from University of Southern California in 1986, and Ph.D. from University of Illinois at Urbana-Champaign in 1992. He is currently a Dean and Professor of School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen and an Empire Innovation Professor of Computer Science and Engineering at the University at Buffalo, State University of New York. He was Allen Henry Endow Chair Professor at the Florida Institute of Technology from July 2003 to December 2007. He was on the faculty of Electrical and Computer Engineering at the University of Rochester from 1992 to 1996 and on the faculty of Electrical and Computer Engineering at the University of Missouri-Columbia from 1996 to 2003.

He has served as the Editor-in-Chief for IEEE Trans. Multimedia from 2014 to 2016. He has also served as the Editor-in-Chief for IEEE Trans. Circuits and Systems for Video Technology from 2006 to 2009. He has been an Editor for several other major IEEE Transactions and Journals, including the Proceedings of IEEE, IEEE Journal of Selected Areas in Communications, and IEEE Journal on Emerging and Selected Topics in Circuits and Systems. He has served as Conference Chair for several major IEEE, ACM and SPIE conferences related to multimedia, video communications and signal processing. His research is supported by NSF, DARPA, Air Force, NASA, Whitaker Foundation, Microsoft, Intel, Kodak, Huawei, and Technicolor.