Index-Aware Reinforcement Learning for Adaptive Video Streaming at the Wireless Edge

Guojun Xiong[1], Xudong Qin[2], Bin Li[2], Rahul Singh[3], Jian Li[1]
[1]SUNY-Binghamton University, [2]Pennsylvania State University, [3]Indian Institute of Science

ABSTRACT

We study adaptive video streaming for multiple users in wireless access edge networks with unreliable channels. The key challenge is to jointly optimize the video bitrate adaptation and resource allocation such that the users' cumulative quality of experience is maximized. This problem is a finite-horizon restless multi-armed multi-action bandit problem and is provably hard to solve. To overcome this challenge, we propose a computationally appealing index policy entitled `Quality Index Policy`, which is well-defined without the Whittle indexability condition and is provably asymptotically optimal without the global attractor condition. These two conditions are widely needed in the design of most existing index policies, which are difficult to establish in general. Since the wireless access edge network environment is highly dynamic with system parameters unknown and time-varying, we further develop an index-aware reinforcement learning (RL) algorithm dubbed `QA-UCB`. We show that `QA-UCB` achieves a sub-linear regret with a low-complexity since it fully exploits the structure of the `Quality Index Policy` for making decisions. Extensive simulations using real-world traces demonstrate significant gains of proposed policies over conventional approaches. We note that the proposed framework for designing index policy and index-aware RL algorithm is of independent interest and could be useful for other large-scale multi-user problems.

CCS CONCEPTS

• **Networks** → **Network performance analysis**; • **Computing methodologies** → **Sequential decision making**.

KEYWORDS

Reinforcement learning, restless bandits, index policy, wireless edge networks, video streaming.

ACM Reference Format:
Guojun Xiong[1], Xudong Qin[2], Bin Li[2], Rahul Singh[3], Jian Li[1]. 2022. Index-Aware Reinforcement Learning for Adaptive Video Streaming at the Wireless Edge. In *The Twenty-third International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (MobiHoc'22), October 17–20, 2022, Seoul, Republic of Korea.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3492866.3549726

MobiHoc'22, October 17–20, 2022, Seoul, Republic of Korea
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9165-8/22/10…$15.00
https://doi.org/10.1145/3492866.3549726

1 INTRODUCTION

Video streaming has dominated the traffic carried by wireless access edge networks in recent years. It has already accounted for 59% of the whole Internet traffic in 2016 and is expected to reach 79% in 2022 [32]. This trend poses significant challenges to meet the stringent constraints on the required quality of service provided by the network so as to satisfy the quality of experience (QoE) of users. Exacerbating this challenge is the fact that many emerging video streaming services such as gaming and smart gyms often involve interactions between *multiple users*, which may compete for the *limited and highly dynamic* wireless edge network resources.

In this paper, we are interested in designing adaptive streaming algorithms which ensure high QoE for multiple users that are streaming video files in wireless access edge networks. Specifically, each user receives video chunks that can be transmitted over *unreliable wireless channels* at *varying bitrates*, and *competes for the limited wireless network resources* for video delivery. This is a challenging problem that lacks effective solutions in the literature. Most of the existing results are based on the widely used *Dynamic Adaptive Streaming over HTTP* (DASH) [29, 31], and only study a *single-user* setting [22, 30, 42]. However, when multiple users compete for wireless network resources, the performance of these policies could degrade dramatically [19, 33]. Existing works for scheduling video files to multiple users often consider an infinite-horizon setting [5, 12]; however, the time horizon of transmitting video contents is rarely infinite, especially for emerging applications with short contents on platforms like TikTok, Twitch, YouTube, etc. These issues are further pronounced in wireless access edge networks where the links connecting the wireless access point to the users are unreliable.

To address these challenges, we pose the problem of adaptive video streaming for multiple users in wireless access edge networks with unreliable channels so as to maximize the cumulative QoE as a Markov decision process (MDP) [27] in Section 3. This MDP turns out to be a *finite-horizon restless multi-armed multi-action bandit* (`RMA2B`) problem [13, 35, 39, 43] since we can choose bitrates amongst multiple levels (i.e., multiple actions) for video streaming in wireless access edge networks. Though we can solve this `RMA2B` by using generic algorithms for MDPs such as the value iteration [27], this approach suffers from the curse of dimensionality, and also does not provide any insight into the solution, since it completely ignores the rich structure present in the underlying MDP. Much effort has been devoted to developing low-complexity and near-optimal solutions for such MDPs, which are largely inspired by the celebrated *Whittle index policy* [38]. However, Whittle-like policies are *only* well defined when the underlying MDP is *indexabile*. Establishing the Whittle indexability of restless multi-armed bandit (RMAB)[1] or RMA2B problems is intractable in many scenarios, especially when the probability transition kernel of the MDP is convoluted [24]. As a result, except for a few special cases, the Whittle indices of many practical problems remain unknown. Finally, most of the existing index policies [13, 35, 37, 45] are provably asymptotically optimal under *a global attractor condition*[2], which is often hard to establish and is only verified numerically [35, 45].

In this paper, we circumvent these limitations by designing new index policies for multi-user adaptive video streaming in wireless access edge networks in Section 4. We first obtain a relaxed problem which can be equivalently formulated as a linear programming (LP) problem using occupancy measures [2]. Then we propose an index policy entitled `Quality Index Policy`, where each user is associated with a so-called "quality index", which is merely based on the occupancy measures solved from the LP. As a result, our index policy is computationally efficient. Unlike the Whittle-like policies, our index policy is well-defined without the requirement of indexability. In contrast with [13, 35, 37, 45], our proof of asymptotic optimality holds regardless of the global attractor condition since we consider a finite-horizon setting. We note that our proposed framework of designing index policies is very general, and can be applied to other MDP problems where multiple users compete for limited network resources with multiple actions.

Since the wireless access edge network environment is highly dynamic, system parameters such as wireless channel conditions and user video playback buffer are typically unknown and time-varying, we further explore the possibility of designing a lightweight machine learning aided algorithm to address these issues in Section 5. Though directly applying popular reinforcement learning (RL) algorithms such as UCRL2 [15] or Thompson Sampling [11] may resolve these issues, the computational complexity and regret of resulting solutions grow exponentially with the number of users and state spaces, making such solutions too slow to be of any practical use. To address these challenges, we propose a RL based algorithm dubbed *Quality Index Aware UCB* (QA-UCB) that leverages the inherent structure of our problem. We show that QA-UCB achieves an optimal sub-linear regret with a low-complexity since it not only leverages the approach of *optimism-in-the-face-of-uncertainty* [3, 15] to balance exploration and exploitation, but more importantly, it learns to leverage the near-optimal `Quality Index Policy` for making decisions. As a result, QA-UCB is easy to implement in large-scale systems. To the best of our knowledge, our work is the first to develop an index-aware RL policy in the context of adaptive video streaming for multiple users in wireless access edge networks.

Finally, our extensive simulations using real video and network traces in Section 6 demonstrate that our proposed polices produce significant performance gain over conventional approaches.

2 RELATED WORK

Video Streaming in Wireless Networks has been extensively studied in different scenarios, where most problems were formulated as constrained optimization problems, e.g., [21, 30, 42], which were solved based on gradient algorithm, Lagrangian methods, game theory, etc. MDP, a systematic stochastic optimization approach has also been adopted to model video streaming [7, 10, 44]. However, these results cannot be applied to the multi-user setting due to the curse of dimensionality. [6, 19, 33] modeled the (multi-user) video streaming problem as a MDP and leveraged off-the-shelf (deep) RL algorithms (e.g., Q-learning and Actor-critic algorithm) directly. As a result, these methods are often computationally expensive since they contend directly with an extremely high dimensional state-action space. Further, these methods have no finite-time performance analysis of the proposed RL algorithms. To the best of our knowledge, none of the above works provided an index based policy for adaptive video streaming for multiple users in wireless access edge network. Such an index based approach naturally lends itself to a lightweight RL framework that can fully exploit the structure of index policy so as to reduce the high computational complexity. This index-aware RL algorithm and the regret analysis further distinguish our work from existing results.

Restless Bandits and Reinforcement Learning. RMAB is a general model for sequential decision making problems but is PSPACE hard [26]. To this end, Whittle [38] proposed a heuristic policy for the infinite-horizon RMAB where the decision maker can only choose two actions for each arm. [13, 35, 45] generalized this to the setting with multiple actions/channels. However, they are either (i) still limited to proving (partial) indexability; or (ii) considering an infinite-horizon setting with the proposed policies only guaranteed to be asymptotically optimal [37] under a difficult-to-verify global attractor condition. In contrast, we consider a finite horizon, and hence existing techniques cannot be directly applied. Finite-horizon RMAB or RMA2B have been less studied. To the best of our knowledge, [14, 43] are the closest to ours; however, [14] studied RMAB with binary actions and [43] focused on homogeneous users with the same underlying environments while we consider a heterogeneous multi-user model with multiple actions. Finally, all above works assumed that the true system parameters are known.

There are also works examining RMAB from a learning perspective, e.g., [25] and references therein; however, these methods did not exploit the special structure available in the problem and content directly with an extremely high dimensional state-action space. For example, the color-UCRL2 [25] suffers from the exponential computational complexity since it needs to solve Bellman equations on a state-space with size growing exponentially with the number of arms. Recently, [4, 9, 39–41] developed RL-based algorithms to explore the problem structure through index policies. However, [4, 9] lacked finite-time performance analysis and the proposed multi-timescale stochastic approximation algorithms often suffer from slow convergence; [40] considered an infinite-horizon setting using Whittle index policy; and [39, 41] was based on a simulator for exploration, which cannot be directly applied here since it is difficult to build a perfect simulator in complex and dynamic wireless access edge network environments.

3 MODEL AND PROBLEM FORMULATION

3.1 Video Streaming Model

We consider a wireless access edge network as shown in Figure 1, that consists of a DASH server, an access point (AP) and multiple

[1] In the original form of RMAB, each arm can be either pulled (active) or not pulled (passive). RMA2B generalizes RMAB to the case that each arm can take multiple actions, which adds an additional layer of uncertainty and complexity.

[2] [37] showed that Whittle index policy fails to be asymptotically optimal if the global attractor condition is not satisfied.

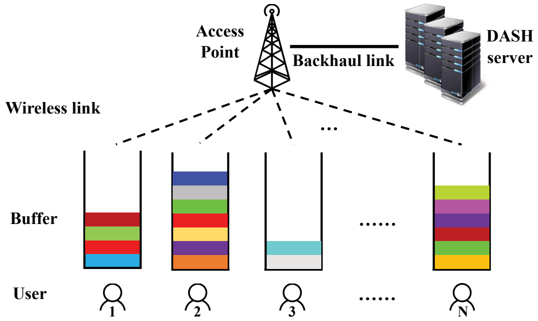

Figure 1: A multi-user adaptive video streaming model in wireless access edge networks with unreliable channels.

users. The AP connects to the DASH server using the wired reliable backhaul link, and serves N users denoted as $\mathcal{N} = \{1, \cdots, N\}$ through *unreliable wireless channels*. The DASH server stores F video files denoted as $\mathcal{F} = \{1, 2, \ldots, F\}$, each of which is divided into a set of consecutive *video segments* or *chunks*, consisting of L seconds of video. Each video chunk is further encoded at different bitrates. Let $\mathcal{R}$ be the finite set of all available bitrate levels. The AP requests video chunks of various bitrates from the DASH server, and serves the N users through unreliable wireless channels that are *constrained in network resources* used for chunk transmissions.

The operating time is divided into multiple units with each unit called a "frame", which is indexed by $t \in \mathcal{T} = \{1, \cdots, T\}$. Each frame consists of multiple physical layer transmission slots since the timescale at which the chunks are requested differs 1-3 orders of magnitude from the timescale at which the physical layer transmissions are scheduled [5, 19, 33]. Without loss of generality, we assume that the duration of each frame is L seconds, equal to the duration of a whole video chunk. We develop algorithms that adaptively choose the bitrates of video chunks. It is important to identify a proper QoE model that measures the satisfaction of a user, which in turn is judged by its long-term engagement [1]. While the QoE might be user-specific, it is widely believed that QoE is largely influenced by the following three key factors [18, 42]: *the video quality*, *the quality variation* and *the frequency of rebuffering events*.

Each user stores its video chunks in its own playout buffer before playing it. It plays one entire video chunk in each frame t. Let $B_n(t)$ be the *buffer occupancy* (measured in seconds) of user n in frame t, i.e., the play time of the video chunk left in the buffer. Since each chunk contains L seconds of video, we denote the set of all possible buffer occupancy as $\mathcal{B} := \{0, L, 2L, \cdots, B_{\max}L\}, \forall n \in \mathcal{N}$.[3] The capacity of the buffer is equal to $B_{\max}L$, and depends upon the storage limitations of users, as well as the service provider's policy. In case the buffer is empty and there is no video chunk to play, there is a video interruption. A *rebuffering* event occurs when the buffer empties, and the user has to wait until the next chunk is delivered to it. Thus, rebuffering events should be avoided so as to guarantee a stall-free playback.

At the beginning of each frame t, the AP chooses the set of users $\mathcal{N}(t) \subseteq \mathcal{N}$ to serve. It also chooses the chunk bitrate $R_n(t) \in \mathcal{R} \triangleq (0, R_{\max}]$ for user $n \in \mathcal{N}(t)$, where $R_{\max} > 0$ is the maximum available bitrate. The relation between the video bitrate and the video quality, as a function of the bitrate, experienced by user n is described by the function $q_n(\cdot) : \mathcal{R} \to \mathbb{R}_+, \forall n$. Thus, if $R_n(t)$ denotes the bitrate of user n during frame t, then the perceived video quality is equal to $q_n(R_n(t))$. We assume that $q_n(\cdot), n \in \mathcal{N}$ are non-decreasing [33, 42]. This means that a higher bitrate yields a higher video quality. Note that we allow for $q_n(\cdot)$ to be non-convex, thus for example it could be sigmoid.

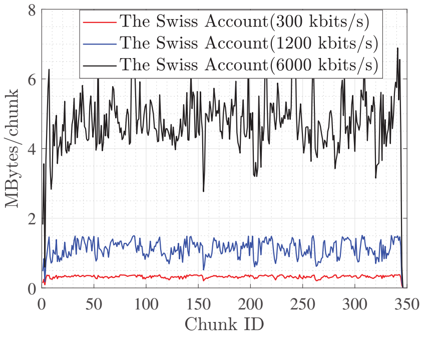

Figure 2: The total bits of a video chunk with different bitrates in real video traces [20]. See Section 6 for more details.

The AP then fetches video chunks of these bitrates from the DASH server, and transmits them to users over unreliable wireless channels. The transmission of video chunks consumes *network resources*[4]. The total amount of network resources available for video transmission is constrained by the budget W. Let $W_n(t)$ be the amount of resource that is used for video transmission of user n in frame t. Let $D_n(t)$ denote the number of video chunks with bitrate of $R_n(t)$ that can be successfully delivered by allocating $W_n(t)$ amount of network resource, where $0 < D_n(t) \leq B_{\max} + 1 - B_n(t)/L$. We model $D_n(t)$ as a random variable with probability distribution $\mathbb{P}(\cdot|R_n(t), W_n(t))$ to reflect the randomness of wireless fading and video content. Denote $C(W_n(t))$ as the throughput of the wireless link between the AP and user n when video chunks are transmitted at resource level $W_n(t)$ in frame t, and let $Q(R_n(t))$ be the total bits of a video chunk with bitrate $R_n(t)$. $C(W_n(t))$ is a random variable to capture the wireless fading effect given $W_n(t)$ amount of allocated network resource. Similarly, $Q(R_n(t))$ is a random variable to model the randomness of video content, as observed in Figure 2, where the total bits of a video chunk with three different bitrates vary across chunks in real video traces [20]. Both $C(W_n(t))$ and $Q(R_n(t))$ affect the distribution of $D_n(t)$, i.e.,

$$\mathbb{P}(D_n(t) = d|R_n(t), W_n(t)) = \mathbb{P}\left(\sum_{i=1}^{d+1} Q^{(i)}(R_n(t)) > C(W_n(t)) \geq \sum_{i=1}^{d} Q^{(i)}(R_n(t))\right), \quad (1)$$

where $Q^{(i)}(R_n(t))$ denotes the i-th realization of $Q(R_n(t))$. By convention $W_n(t) = 0$ means no resource is allocated to user n and hence $\mathbb{P}(D_n(t) = 0|R_n(t), 0) = 1$ and $\mathbb{P}(D_n(t) = d|R_n(t), 0) = 0, \forall d \in [1, B_{\max} + 1 - B_n(t)/L]$.

[3] Our results hold for user-dependent $B_{n,\max}, \forall n$, at the cost of complicated notations.

[4] Multiple types of network resources fit into our model. For example, (i) transmission power is used to combat the randomness of wireless channels, which is usually controlled at the physical layer. (ii) downlink bandwidth is allocated by the AP into disjoint sub-bands to each user, which determines the channel throughput. (iii) time can be allocated to serve users one-by-one and each user consumes a certain amount of time slots for transmission.

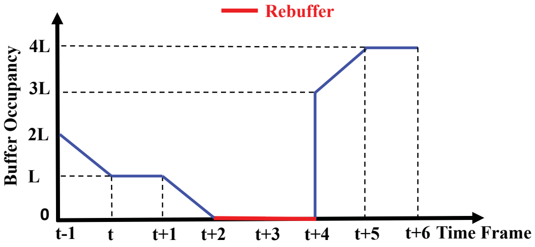

Figure 3: An illustration of buffer dynamics of user n.

To achieve high efficiency of adaptive video streaming over wireless edge network, it is expected that the quality of the video chunks transmitted to users to be as high as possible. Moreover, the video quality should also be kept as "smooth" as possible by avoiding frequently switching the quality. Based on the discussion above, we let the instantaneous QoE of user n at frame t be given as follows:

$$\begin{aligned}\text{QoE}_n(t) = D_n(t) \cdot q_n(R_n(t)) - \alpha_n \mathbb{1}_{\{B_n(t)=0\}} \\ - \beta_n \left|q_n(R_n(t)) - q_n(\Gamma_n(t))\right| \mathbb{1}_{\{D_n(t)>0\}}, \end{aligned} \tag{2}$$

where $\Gamma_n(t)$ tracks the bitrate of last successfully received chunk for user n before frame t, α_n and β_n are non-negative parameters that capture the importance of rebuffering event and quality variation of user n in the QoE evaluation. The QoE defined in (2) is a random variable and the value depends on $D_n(t)$. Note that quality variation only occurs when newly successfully downloaded chunks in current frame have different quality compared with the last successfully received chunks. Since the network resource is limited, the above QoE is achieved under the following constraint

$$\sum_{n \in \mathcal{N}(t)} W_n(t) \leq W, \quad \forall t \in \mathcal{T}. \tag{3}$$

3.2 MDP-based Problem Formulation

We pose the problem of adaptively choosing bitrates and allocating resource to maximize the cumulative user-perceived QoE as a finite-horizon MDP. Specifically, each user n is modeled as a MDP $(\mathcal{S}_n, \mathcal{A}_n, \mathbb{P}_n, r_n, \mathbf{s}_0, T)$. We begin by describing the state-space $\mathcal{S}_n$, action space $\mathcal{A}_n$, transition kernel $\mathbb{P}_n : \mathcal{S}_n \times \mathcal{A}_n \times \mathcal{S}_n \mapsto \mathbb{R}$, reward function $r_n : \mathcal{S}_n \times \mathcal{A}_n \mapsto \mathbb{R}$ of this MDP of user n. For simplicity, we assume that all N users have the same initial distribution $\mathbf{s}_0$, and the time horizon $T < \infty$.

State. Denote $\mathrm{S}_n(t) := (B_n(t), \Gamma_n(t)) \in \mathcal{S}_n$ as the state of user n at frame t, where $B_n(t) \in \mathcal{B}$ is the buffer occupancy at frame t, $\Gamma_n(t) \in \mathcal{R}$ is the bitrate of the video chunk which was last successfully received before frame t, which we call "quality tracker". We let $\mathcal{S}_n \equiv \mathcal{S} = \mathcal{B} \times \mathcal{R}$, and also $\mathcal{S}(t) = (\mathrm{S}_1(t), \cdots, \mathrm{S}_N(t))$.

Action. The AP makes the following two decisions for each user n at the beginning of frame t: (i) the chunk bitrate $R_n(t)$, and (ii) the resource $W_n(t)$ allocated to deliver it. Denote this by $\mathbf{A}_n(t) := (R_n(t), W_n(t)) \in \mathcal{A}_n$, where $R_n(t) \in \mathcal{R}$, $W_n(t) \in \mathcal{W} \triangleq [0, W]$, and $\mathcal{A}_n \equiv \mathcal{A} = \mathcal{R} \times \mathcal{W}$. Also let $\mathcal{A}(t) = (\mathbf{A}_1(t), \cdots, \mathbf{A}_N(t))$.

An adaptive video streaming policy π maps the states of all users $\mathcal{S}(t)$ to video streaming decisions $\mathcal{A}(t)$, i.e., $\mathcal{A}(t) = \pi(\mathcal{S}(t))$, $\forall t \in \mathcal{T}$.

Transition Kernel. When user n is served and a rebuffering event does not occur in frame t, i.e., $B_n(t) = B > 0$, $\forall n \in \mathcal{N}$, user n plays one video chunk in frame t. If $D_n(t) = d$ chunks are successfully delivered to user n with bitrate $R_n(t)$ by allocating $W_n(t)$ amount of resource, the buffer occupancy of user n becomes $B + (d-1)L$, and the quality tracker is updated to be the quality of the successfully received chunks. This occurs with probability $\mathbb{P}(D_n(t) = d | R_n(t), W_n(t))$ as defined in (1). More precisely, we have

$$\begin{aligned}\mathbb{P}\Big(S_n(t+1) = (B + (d-1)L, R_n(t)) | S_n(t) = (B, \Gamma), \\ A_n(t) = (R_n(t), W_n(t))\Big) = \mathbb{P}(D_n(t) = d | R_n(t), W_n(t)), \end{aligned} \tag{4}$$

for $\forall d \in [1, B_{\max} + 1 - B_n(t)/L]$. Otherwise, the buffer occupancy is decreased by L seconds, and the quality tracker is unchanged,

$$\begin{aligned}\mathbb{P}\Big(S_n(t+1) = ((B - L)_+, \Gamma) | S_n(t) = (B, \Gamma), \\ A_n(t) = (R_n(t), W_n(t))\Big) = \mathbb{P}(D_n(t) = 0 | R_n(t), W_n(t)). \end{aligned} \tag{5}$$

Similarly, when user n is served and a rebuffering event occurs in frame t, i.e., $B_n(t) = 0$, user n has no video chunk to play in frame t. As a result, its buffer occupancy becomes dL (rather than $(d-1)L$ in (4)) when $D_n(t) = d$ chunks are successfully delivered,

$$\begin{aligned}\mathbb{P}\Big(S_n(t+1) = (dL, R_n(t)) | S_n(t) = (0, \Gamma), \\ A_n(t) = (R_n(t), W_n(t))\Big) = \mathbb{P}(D_n(t) = d | R_n(t), W_n(t)), \end{aligned} \tag{6}$$

for $\forall d \in [1, B_{\max} + 1 - B_n(t)/L]$. Otherwise, the buffer occupancy maintains to be zero,

$$\begin{aligned}\mathbb{P}\Big(S_n(t+1) = (0, \Gamma) | S_n(t) = (0, \Gamma), \\ A_n(t) = (R_n(t), W_n(t))\Big) = \mathbb{P}(D_n(t) = 0 | R_n(t), W_n(t)). \end{aligned} \tag{7}$$

When user n is not served in frame t, the state transition probability of user n satisfies

$$\begin{aligned}\mathbb{P}\Big(S_n(t+1) = ((B - L)_+, \Gamma) | S_n(t) = (B, \Gamma), \\ A_n(t) = (R_n(t), 0)\Big) = 1, \forall B \in \mathcal{B}. \end{aligned} \tag{8}$$

In the following, we provide an example as shown in Figure 3 to illustrate the buffer dynamics of user n as defined in (4)-(8).

Example 1. *Suppose that the buffer occupancy of user n is $B_n(t-1) = 2L$ at the beginning of frame $t-1$, and it is served but receives no video chunk in this frame. Thus its buffer occupancy becomes $B_n(t) = L$ at the beginning of frame t. This event occurs with probability defined in (5). In frame t, user n plays one chunk and successfully receives one chunk, which occurs with probability defined in (4). To this end, $B_n(t+1) = L$. Suppose that user n is not served in frame $t+1$ and hence we have $B_n(t+2) = 0$ since it still plays one chunk. This event occurs with probability defined in (8). As a result, a rebuffering occurs in frame $t+2$ and user n cannot play a video. Suppose that user n is served in this frame but no video chunk is successfully delivered, i.e., $B_n(t+3) = 0$, which occurs with probability defined in (7). Thus a rebuffering event occurs again in frame $t+3$ and user n cannot play a video chunk. Now suppose user n is served in this frame with $3L$ video chunks successfully delivered. Hence we have $B_n(t+4) = 3L$, which occurs with probability defined in (6). In frame $t+4$, user n plays one chunk and successfully receives two chunks, and hence $B_n(t+5) = 4L$. This occurs with probability defined in (4). Finally, similar event occurs in frame $t+5$ as in frame t.*

Reward. The instantaneous reward/QoE received by user n in frame t is equal to (2). In particular, we write it explicitly as a function of state and action as $\text{QoE}_n(S_n(t), A_n(t))$.

Adaptive Video Streaming Problem. Our objective is to design a policy π that maximizes the expected cumulative rewards of all N users, while ensuring that the network resource utilization is below its capacity, i.e., we have to solve the following problem:

$$\begin{aligned} \max_{\pi} \quad & \mathbb{E}_{\pi}\left(\sum_{n=1}^{N}\sum_{t=1}^{T}\text{QoE}_n(S_n(t), A_n(t))\right) \\ \text{s.t.} \quad & \sum_{n=1}^{N} W_n(t) \le W, \quad \forall t \in \mathcal{T}, \end{aligned} \tag{9}$$

where the subscript denotes the fact that expectation is taken with respect to the measure induced by the policy π. We refer to (9) as the "original problem". Though in theory this could be solved using dynamic programming [27], the complexity is $O(|\mathcal{S}|^N|\mathcal{A}|^N)$, and hence suffers from the curse of dimensionality. We overcome this difficulty by developing an index-based policy that is computationally appealing and provably optimal.

4 INDEX POLICY DESIGN AND ANALYSIS

We now propose a class of index policies for the QoE maximization problem (9). Our proposed solution utilizes a so-called "relaxed problem", whose solution provides an upper bound for the original problem (9). This relaxed problem can be posed as a linear programming (LP). We solve this LP and obtain an optimal occupation measure, which is then used to develop an index policy entitled `Quality Index Policy` for the QoE maximization problem (9). We show that our proposed index policy is asymptotically optimal when both the number of users and the resource constraint go to infinity with their ratio holding constant.

4.1 The Relaxed Problem

We relax the instantaneous constraints that are required to hold in each frame t in the original problem (9) so that now they need to hold only on average. This gives us the following *relaxed problem*

$$\begin{aligned} \max_{\pi} \quad & \mathbb{E}_{\pi}\left(\sum_{n=1}^{N}\sum_{t=1}^{T}\text{QoE}_n(S_n(t), A_n(t))\right) \\ \text{s.t.} \quad & \mathbb{E}_{\pi}\left(\sum_{n=1}^{N} W_n(t)\right) \le W, \quad \forall t \in \mathcal{T}. \end{aligned} \tag{10}$$

Then we immediately have the following result

Lemma 1. *The optimal value achieved by the relaxed problem in* (10) *is an upper bound of that of the original problem* (9).

Proof. The proof is straightforward since the constraint in (10) expands the feasible region of (9). □

It is well known (see e.g. [2]) that the relaxed problem (10) can be reduced to a LP in which the decision variables are the occupation measures of the controlled process. We begin with some definitions.

Definition 1. *(Occupancy measure) [2]. The occupancy measure μ of a policy π in a finite-horizon MDP is defined as the expected number of visits to a state-action pair (s, a) in each frame t. Formally,*

$$\mu = \Big\{\mu_n(s, a; t) = \mathbb{P}(S_n(t) = s, A_n(t) = a) : \forall n \in \mathcal{N}, t \in \mathcal{T}\Big\}. \tag{11}$$

It can be easily checked that $\sum_{(s,a)} \mu_n(s, a, t) = 1$ and $0 \le \mu_n(s, a, t) \le 1$, $\forall n \in \mathcal{N}$, $s \in \mathcal{S}$, $a \in \mathcal{A}$, $t \in \mathcal{T}$. Hence the occupancy measure $\mu_n, \forall n$ is a probability measure. Define the expected value of $\text{QoE}_n(S_n(t), A_n(t))$ as $r_n(s, a, t)$. Using this definition, the relaxed problem (10) can be reformulated as a LP [2]:

Proposition 1. *The relaxed problem* (10) *is equivalent to the following LP*

$$\max_{\mu} \sum_{n=1}^{N}\sum_{t=1}^{T}\sum_{(s,a)} \mu_n(s, a; t) r_n(s, a; t) \tag{12}$$

$$\text{s.t.} \sum_{n=1}^{N}\sum_{(s,a)} w\mu_n(s, a; t) \le W, \ \forall t \in \mathcal{T}, \tag{13}$$

$$\sum_{a\in\mathcal{A}} \mu_n(s, a; t) = \sum_{(s',a')} \mu_n(s', a'; t-1) P_n(s', a', s), \tag{14}$$

$$\sum_{a\in\mathcal{A}} \mu_n(s, a, 1) = \mathbf{s}_0(s), \quad \forall s \in \mathcal{S}, n \in \mathcal{N}, \tag{15}$$

where (13) is a restatement of the constraint in (10)*, and w is associated with the action a; (14) indicates the transition of the occupancy measure from time frame $t-1$ to time frame t; and (15) represents the initial condition at frame* 1*.*

Let $\mu^\star = \big\{\mu_n^\star(s, a; t) : n \in \mathcal{N}, t \in \mathcal{T}\big\}$ be an optimal solution to the above LP. We now construct a Markovian randomized policy $\xi^\star = \{\xi_n^\star(t) : n \in \mathcal{N}, t \in \mathcal{T}\}$ from $\mu^\star$ as follows: if the state $S_n(t)$ in frame t is equal to s, then $\xi_n^\star(t)$ chooses the action a with a probability equal to

$$\xi_n^\star(s, a; t) := \frac{\mu_n^\star(s, a; t)}{\sum_{a\in\mathcal{A}} \mu_n^\star(s, a; t)}. \tag{16}$$

If the denominator of (16) is equal to zero, i.e., state s for user n is not reachable at time t, user n can be simply made "unserved", i.e., $\xi_n^\star(s, 0; t) = 1$ and $\xi_n^\star(s, a; t) = 0, \forall a \in \mathcal{A} \setminus \{0\}$. Note that $\xi^\star$ is non-stationary since it is time-dependent and is Markovian since it makes decisions on the basis of current state only.

4.2 The Quality Index Policy

Our proposed policy attaches the following index $\mathcal{I}_n(s; t)$ to each user n in frame t,

$$\mathcal{I}_n(s; t) := \sum_{a\in\mathcal{A}\setminus\{0\}} \xi_n^\star(s, a; t), \quad \forall n \in \mathcal{N}, \tag{17}$$

where $\xi_n^\star(s, a; t)$ is defined in (16). We call this index *the quality index* since the probability $\xi_n^\star(s, a; t)$ is related with the *quality* of action a in state s in frame t towards the QoE maximization, and is determined by the occupation measure derived by solving the LP (12)-(15). Let $\mathcal{I}(t) := \{\mathcal{I}_n(s; t) : n \in \mathcal{N}\}$ denote the set of quality indices associated with all N users in frame t.

However, the optimal indices for the relaxed problem (10) is not always feasible for the original problem (9), since in the latter, at most W network resources can be consumed by all users at a time. To resolve this issue, our `Quality Index Policy` prioritizes the users according to a decreasing order of their quality indices, and

Algorithm 1 `Quality Index Policy`

```
1: Initialize state $S_n(1)$ for user $n, \forall n$ and $\mathcal{N}(t) = 0, \forall t$.
2: Construct the LP in (12)-(15) and solve the occupancy measure $\mu^\star$;
3: Compute $\xi_n^\star(s, a, t), \forall s, a, t$ according to (16);
4: Construct the quality index set $\mathcal{I}(t) := \{\mathcal{I}_n(s;t) : n \in \mathcal{N}\}$ as in (17), and sort $\mathcal{I}_n(s;t)$ in a decreasing order;
5: while $\sum_{n \in \mathcal{N}(t)} W_n(t) \leq W$ do
6:     Serve users with quality indices in a decreasing order (Step 4) and randomly select a feasible activation action according to the probability $\xi_n^\star(s, a; t)$ in (16).
7: end while
```

then serves the maximum number of users as long as the network resource constraint $\sum_{n \in \mathcal{N}(t)} W_n(t) \leq W$ is satisfied, where $\mathcal{N}(t)$ is the subset of users that the AP transmits video chunks to in frame t. The remaining users, i.e. those in the set $\mathcal{N} \setminus \mathcal{N}(t)$, are not served in frame t. Specifically, for each served user, its action is randomly selected according to the probability $\xi_n^\star(s, a; t)$ in (16). This policy is summarized in Algorithm 1, and denoted as $\pi^\star = \{\pi_n^\star, n \in \mathcal{N}\}$.

Remark 1. `Quality Index Policy` *is computationally tractable since it requires us to solve a LP with $N|\mathcal{S}||\mathcal{A}|T$ decision variables. Note that the design of our index policy can be applied to general multi-agent MDP with constraints, and not just the QoE maximization problem for multi-user video streaming over wireless edges. In that sense, the applicability of our proposed framework is of independent interest, and beyond the current problem. Finally, our proposed index policy is well-defined even when the problem is not indexable [38].*

4.3 Asymptotic Optimality

We now show that our `Quality Index Policy` is asymptotically optimal in the same asymptotic regime as that in Whittle [38] and others [35, 37, 45]. For abuse of notation, let the number of users be ηN and the resource constraint be ηW in the asymptotic regime with $\eta \to \infty$. In other words, we consider N different classes of users with each class containing η users. Let $\text{QoE}^\pi(\eta W, \eta N)$ denote the expected QoE of the original problem (9) under an arbitrary policy π for such a system. Denote the optimal policy for the original problem (9) as $\pi^{opt} := \{\pi_n^{opt}, \forall n \in \mathcal{N}\}$.

Theorem 1. *The* `Quality Index Policy` *(Algorithm 1) is asymptotically optimal, i.e.,*

$$\lim_{\eta \to \infty} \frac{1}{\eta} \Big(QoE^{\pi^\star}(\eta W, \eta N) - QoE^{\pi^{opt}}(\eta W, \eta N) \Big) = 0. \tag{18}$$

Remark 2. *Theorem 1 indicates that as the number of per-class users goes to infinity, the average gap between the performance achieved by our* `Quality Index Policy` *$\pi^\star$ and the optimal policy π^{opt} tends to be zero.*

Proof Sketch: For any policy $\pi^\star$ derived from Algorithm 1, the left hand side of (18) is non-positive. Hence we need to show that for $\pi^\star$. Let $B_n(s;t)$ be the number of class n users in state s at time t and $D_n(s, a; t)$ be the number of class n users in state s at time t that are being served with action $a \in \mathcal{A} \setminus \{0\}$. By induction, we show that $B_n(s;t)/\eta \to P_n(s;t)$ and $D_n(s, a; t)/\eta \to P_n(s;t)\xi_n^\star(s, a; t)$, respectively, as $\eta \to \infty$ almost surely. This leads to the fact that $\lim_{\eta \to \infty} \frac{1}{\eta} \text{QoE}^{\pi^\star}(\eta W, \eta N) = \sum_{n=1}^N \sum_{t=1}^T \sum_{(s,a)} \mu_n^\star(s, a; t) r_n(s, a; t)$, which is an upper bound of $\lim_{\eta \to \infty} \frac{1}{\eta} \text{QoE}^{\pi^{opt}}(\eta W, \eta N)$.

5 REINFORCEMENT LEARNING SOLUTIONS

The computation of the `Quality Index Policy` requires the knowledge of transition probabilities and reward functions associated with the user-level MDPs that were discussed in Section 3.2. Assuming that these parameters are known is unrealistic since the wireless access edge network environment is often highly dynamic with these parameters varying over time. Hence, we now design learning algorithms that combine the optimism principle on top of the `Quality Index Policy`. We denote the resulting learning rule as *Quality Index Aware UCB* (`QA-UCB`). We prove that `QA-UCB` achieves an optimal sub-linear regret. Moreover, the multiplicative "pre-factor" that goes with the time-horizon dependent function in the regret, is quite low since it fully leverages the structure of the `Quality Index Policy` for making decisions.

5.1 The Learning Algorithm: `QA-UCB`

Algorithm Overview. We adapt the upper confidence bound (UCB) strategy [3] to our problem and call our RL algorithm as the `QA-UCB` policy, which is summarized in Algorithm 2. Specifically, `QA-UCB` decomposes the total operating time into episodes, and each episode is composed of H consecutive frames. Let K be the total number of episodes until time T. We denote the k-th episode by $\mathcal{H}_k$ and let τ_k denote the time when it starts. Thus, $T = KH$. Each episode consists of two phases: *planning and policy execution.*

At the planning phase of each episode (lines 2-4 in Algorithm 2), `QA-UCB` constructs a confidence ball that contains a set of plausible MDPs [15] for each user $n \in \mathcal{N}$. In order to obtain an optimistic estimate of the true MDP parameters, `QA-UCB` solves an *optimistic planning* problem where the MDP parameters can be chosen from the constructed confidence ball. This problem turns out to be a LP where the decision variables are the occupancy measures corresponding to the process associated with N users. The planning problem, which is referred to as *extended LP* in Algorithm 2 is described below. `QA-UCB` then defines the corresponding `Quality Index Policy` using the solutions to the extended LP.

At the policy execution phase of each episode (line 5 in Algorithm 2), `QA-UCB` executes the constructed `Quality Index Policy`. The key contribution and novelty of our proposed RL algorithm is to leverage our proposed `Quality Index Policy` for making decisions, rather than directly contending with an extremely large state-action space to balance between exploration and exploitation. These together contribute to the sub-linear regret of `QA-UCB` with a low-complexity, which is discussed in detail later.

Optimistic Planning. `QA-UCB` maintains two counts for each user n. Let $C_n^{k-1}(s, a)$ be the number of visits to state-action pairs (s, a) until τ_k, and $C_n^{k-1}(s, a, s')$ be the number of transitions from s to s' under action a. At τ_k, `QA-UCB` updates the respective counts as $C_n^k(s, a) = C_n^{k-1}(s, a) + \sum_{h=1}^H \mathbb{1}(S_n^k(h) = s, A_n^k(h) = a)$, and $C_n^k(s, a, s') = C_n^{k-1}(s, a, s') + \sum_{h=1}^H \mathbb{1}(S_n^k(h+1) = s' | S_n^k(h) = s, A_n^k(h) = a), \forall (s, a) \in \mathcal{S} \times \mathcal{A}$ and $\forall (s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ for user n, where $S_n^k(h)$ is the state of user n at the h-th time frame in episode k.

Algorithm 2 QA-UCB Policy

```
Require: Initialize $C_n^1(s,a)=0$, and $\hat{P}_n^1(s'|s,a)=1/|\mathcal{S}|$
for $k=1,2,\cdots,K$ do
    Construct $\mathcal{P}_n^k(s,a)$ and $\mathcal{R}_n^k(s,a)$ according to (21) at $\tau_k$;
    Compute the optimal solution to the extended LP (23);
    Recover $\xi^{k,\star}$ according to (24) and establish the corresponding Quality Index Policy $\pi^{k,\star}$;
    Execute $\pi^{k,\star}$ in the current episode.
end for
```

At τ_k, QA-UCB estimates the true transition model and the true reward by the corresponding empirical averages as:

$$\hat{P}_n^k(s'|s,a)=\frac{C_n^{k-1}(s,a,s')}{\max\{C_n^{k-1}(s,a),1\}}, \tag{19}$$

$$\hat{r}_n^k(s,a)=\frac{\sum\limits_{\tau=1}^{k-1}\sum\limits_{h=1}^{H} r_n(s,a)\mathbb{1}(S_n^\tau(h)=s, A_n^\tau(h)=a)}{\max\{C_n^{k-1}(s,a),1\}}. \tag{20}$$

The QA-UCB further defines confidence intervals for the transition probabilities (resp. rewards) such that the true transition probabilities (resp. rewards) lie in them with high probabilities. Formally, for $\forall(s,a)\in\mathcal{S}\times\mathcal{A}$, we define

$$\begin{aligned}&\mathcal{P}_n^k(s,a):=\{\tilde{P}_n^k(s'|s,a),\forall s':|\tilde{P}_n^k(s'|s,a)-\hat{P}_n^k(s'|s,a)|\le\delta_n^k(s,a)\},\\&\mathcal{R}_n^k(s,a):=\{\tilde{r}_n^k(s,a),\forall s,a:\tilde{r}_n^k(s,a)=\hat{r}_n^k(s,a)+\delta_n^k(s,a)\},\end{aligned} \tag{21}$$

where the size of the confidence intervals $\delta_n^k(s,a)$ is built according to the Hoeffding inequality [23] for $\epsilon\in(0,1)$ as

$$\delta_n^k(s,a)=\sqrt{\frac{1}{2C_n^{k-1}(s,a)}\log\Big(\frac{4|\mathcal{S}||\mathcal{A}|N(k-1)^2H^2}{\epsilon}\Big)}. \tag{22}$$

To this end, the set of plausible MDPs associated with the confidence intervals is $\mathcal{M}_n^k=\{M_n=(\mathcal{S},\mathcal{A},\tilde{r}_n,\tilde{P}_n):\tilde{P}_n^k(\cdot|s,a)\in\mathcal{P}_n^k(s,a),\tilde{r}_n^k(s,a)\in\mathcal{R}_n^k(s,a)\}$. Then QA-UCB computes a policy $\pi^{\star,k}$ by performing optimistic planning. In other words, given the set of plausible MDPs, it selects an optimistic MDP and an optimistic policy by solving a "modified LP", which is similar to the LP (12)-(15) but with transition and reward functions replaced by $\tilde{P}_n^k(\cdot|\cdot,\cdot)$ and $\tilde{r}_n^k(\cdot,\cdot;\cdot)$, $\forall s,a,h,n,k$, respectively, in the confidence intervals (21) since the corresponding true values are not available.

The Extended LP Problem. We cannot directly solve the "modified LP" since the true transitions and rewards are unknown. To this end, we rewrite it as an extended LP problem by leveraging the *state-action-state occupancy measure* $z_n^k(s,a,s',h)$ defined as $z_n^k(s,a,s',h)=P_n(s'|s,a)\mu_n^k(s,a,h)$ to express the confidence intervals of the transition probabilities. The extended LP over $z^k:=\{z_n^k(s,a,s',h),\forall n\in\mathcal{N}\}$ is given as follows:

$$\begin{aligned}\max_{z^k}\ &\sum_{h=1}^{H}\sum_{n=1}^{N}\sum_{(s,a,s')} z_n^k(s,a,s';h)\tilde{r}_n^k(s,a;h)\\ \text{s.t.}\ &\sum_{n=1}^{N}\sum_{(s,a,s')} wz_n^k(s,a,s';h)\le W,\ \forall h\in\mathcal{H},\\ &\sum_{a,s'} z_n^k(s,a,s';h)=\sum_{s',a'} z_n^k(s',a',s,h-1),\quad\forall h\in\mathcal{H},\\ &\frac{z_n^k(s,a,s';h)}{\sum_y z_n^k(s,a,y;h)}-(\hat{P}_n^k(s'|s,a)+\delta_n^k(s,a))\le 0,\\ &-\frac{z_n^k(s,a,s';h)}{\sum_y z_n^k(s,a,y;h)}+(\hat{P}_n^k(s'|s,a)-\delta_n^k(s,a))\le 0,\\ &\sum_{a,s'} z_n^k(s,a,s';1)=\mathbf{s}_0^k(s),\quad\forall s\in\mathcal{S},\forall n\in\mathcal{N}.\end{aligned} \tag{23}$$

This LP has $O(|\mathcal{S}|^2|\mathcal{A}|HN)$ constraints and decision variables. Such an approach was also used in the context of adversarial MDPs [16, 28] and in constrained MDPs [8, 17]. Once we compute the optimal solution $z^{k,\star}$ to (23), we recover the Markovian randomized policy $\xi^{k,\star}$ as

$$\xi_n^{k,\star}(s,a;h)=\frac{\sum_{s'} z_n^{k,\star}(s,a,s';h)}{\sum_{b,s'} z_n^{k,\star}(s,b,s';h)},\quad\forall n\in\mathcal{N}. \tag{24}$$

Finally, we compute the *quality index* in (17) using the policy derived in (24), from which we construct the Quality Index Policy $\pi^{k,\star}:=\{\pi_n^{k,\star},\forall n\}$, and execute this policy in this episode. We summarize this process in Algorithm 2.

5.2 The Learning Regret

We use *regret* to evaluate the efficiency of QA-UCB policy, which is defined as the expected gap between the offline optimum, i.e., the best policy under full knowledge of all transition probabilities and reward information, and the cumulative reward obtained by QA-UCB. Specifically, denote the cumulative reward under policy π as $R(\mathbf{s}_0,T):=\sum_{t=1}^{T}\sum_{n=1}^{N} r_n(t)$, which is a random variable. Then the expected average reward under policy π satisfies $\gamma(\mathbf{s}_0):=\lim_{T\to\infty}\frac{1}{T}\mathbb{E}_\pi[R(\mathbf{s}_0,T)]$, and the optimal average reward is $\gamma^\star:=\sup_\pi\gamma(\mathbf{s}_0)$, which is independent of initial states for MDPs with finite diameter [27]. Then the regret of π is defined as $\Delta(T):=T\gamma^\star-\mathbb{E}_\pi[R(\mathbf{s}_0,T)]$. The following theorem establishes the finite-time performance of QA-UCB policy.

Theorem 2. *The regret of* QA-UCB *policy satisfies*

$$\Delta(T)=\tilde{O}\Big(\sqrt{T}\Big(\sqrt{\log T}+W\sqrt{|\mathcal{S}||\mathcal{A}|N}\Big)\Big). \tag{25}$$

Proof Sketch: The results are achieved by combing two steps: (i) *Step 1.* We bound the regret due to the inherent randomness of rewards. We show that the cumulative regret can be expressed as the sum of regrets accumulated during each episode. (ii) *Step 2.* We further classify the episodic regrets into either "failure event" (the true MDP is not in the confidence ball) or "good event" (the true MDP is within confidence ball). We show that these are upper bounded as $W\sqrt{T}$ and $\sqrt{T\log T}$, respectively.

Remark 3. QA-UCB *achieves an $\tilde{O}(\sqrt{T})$ regret no worse than the state-of-the-art colored-UCRL2 [25] for* RMAB. *However, colored-UCRL2 suffers from an exponential implementation complexity since it derives a policy by solving Bellman equations on a state-action space with size growing exponentially with the number of users in each episode. In contrast, our* QA-UCB *derives a policy by leveraging the proposed lightweight* Quality Index Policy *along with solving an LP, whose complexity grows linearly with the state-action space. In addition,*

Video Information	Resolution/Bitrate	Mean Chunk Size	Variance
The Swiss Account, Sport, Length: 57:34, Chunk length: 10s	480×360 300 kbits/s	0.3144MB	0.0017
	1280×720 1200 kbits/s	1.2265MB	0.0349
	1920×1080 6000 kbits/s	4.3414MB	2.5067
Big Buck Bunny, Animation, Length: 09:46, Chunk length: 10s	480×360 300 kbits/s	0.3217MB	0.0020
	1280×720 1200 kbits/s	1.1166MB	0.0607
	1920×1080 6000 kbits/s	4.7303MB	0.7479

Table 1: Properties of Swiss Account video traces [20].

the explore-then-commit mechanism has recently been adopted to design low-complexity RL algorithms for RMAB*, e.g., Restless-UCB [36] and R(MA)^{2}B-UCB [39] by sampling and constructing the plausible MDPs only once via a simulator (a generative model). However, it is infeasible to build a perfect simulator in dynamic wireless access edge environments as considered in this paper. Furthermore, [36] sacrifices the regret performance to $\tilde{O}(T^{2/3})$ since it depends on the performance of an offline oracle approximator for policy execution. While R(MA)^{2}B-UCB [39] achieves $\tilde{O}(\sqrt{T})$ regret, the multiplicative "pre-factor" that goes with the time dependent function in the regret is linear in $|\mathcal{S}|$ and $|\mathcal{A}|$, while our* QA-UCB *has a much smaller "pre-factor".*

6 EXPERIMENTS

In this section, we numerically evaluate the performance of our proposed Quality Index Policy and QA-UCB.

6.1 Evaluation Setup

Video Traces. We evaluate our policies using the Swiss Account video traces [20]. In particular, we consider the sport trace in [20] with key traces characteristics summarized in Table 1. All videos are encoded into multiple chunks, with each chunk of $L = 10$ seconds. Each video consists of three bitrates: 300 kbits/s, 1200 kbits/s and 6000 kbits/s, from which we abstract the bitrate levels as $\mathcal{R} = \{1, 2, 3\}$. The total bits of a video chunk with different bitrates across chunks are presented in Figure 2. It is clear from Table 1 that the higher the bitrate, the larger mean chunk size and variance. This is consistent with our QoE model as described in Section 3.

Baselines. We compare our policies with the following baselines:

▷ *Vanilla:* A base case with served users being allocated the highest resources, and no differentiation between users.

▷ *Greedy:* Each user greedily selects the action with the largest reward for current state.

▷ *Deep Q Network (DQN):* This is a deep Q-learning policy designed for multi-user wireless video streaming in [6]. We exploit the same configurations as in [6], i.e., two hidden layers with 64 and 32 neurons in each layer with double DQN algorithm, and a ϵ-greedy policy with decayed learning rate.

▷ *Panda* [21] uses "probe and adapt" mechanism to adjust video bitrate based on estimated network bandwidth, where "probe" means that users constantly measure the network bandwidth and "adapt" indicates that users adapt their video bitrates based the "probe".

Monte Carlo Simulation. We consider 20 users, and a total resource $W = 10$ MBps with $\mathcal{W} = \{0, 0.25\text{MBps}, 0.5\text{MBps}, 1\text{MBps}\}$. We apply the Monte Carlo method to estimate the success download and state transition probabilities based on the statistics obtained from network traces [34], in which the dynamics of a user's download speed is presented in Figure 4. Based on this, we generate a wireless fading channel for each user in our simulation. Specifically, in each round, each user experiences a wireless download speed $w \in \mathcal{W}$ and $w/2$ with probability 0.7 and 0.3, respectively. With the allocated resource, each user receives video chunks with selected bitrates. We use Monte Carlo simulation with 10^6 independent trails to compute the average success download probability and then generate the state transition kernel accordingly. Finally, we set the maximum buffer size for each user as $B_{max} = 9$ and the QoE is defined as $q_n(R) = R, \alpha_n = 3, \beta_n = 1, \forall n$ and $R \in \mathcal{R}$.

6.2 Evaluation Results

Asymptotic Optimality. We first validate the asymptotic optimality of Quality Index Policy (see Theorem 1). In particular, we define the difference of average per-user QoE obtained by any policy with that obtained from the theoretical upper bound solved from the LP (12)-(15) (see Proposition 1) as the per-user optimality gap. Figure 5 shows the per-user optimality gap of Quality Index Policy with different number of users when the total number of frames is $T = 100$ and $T = 200$. We observe that as the number of users increases, the per-user optimality gap decreases significantly and closes to zero under both settings. This verifies the asympototic optimality in Theorem 1. Moreover, the per-user optimality gap also decreases with a larger frame number.

Optimality Gap. The optimality gap is defined in the similar way as the per-user optimality gap, but measures the gap between the total expected cumulative QoE. We run QA-UCB for $K = 100$ and $K = 200$ episodes with $H = 100$ frames in each episode. It is clear from Figure 6 that our Quality Index Policy performs most closely to the theoretical optimum and significantly outperforms existing algorithms. From Figure 6, it is also evident that QA-UCB performs close to Quality Index Policy as the number of episodes increases since QA-UCB leverages our proposed index policy for making decisions along with the information learned from the episodes. Furthermore, QA-UCB significantly outperforms DQN.

Average QoE. Figure 7 shows the average QoE attained by different policies. The error bars are drawn based on the 95% percentage of QoE CDFs in Figure 8. It is clear that QA-UCB outperforms all baselines. The improvement becomes more pronounced when we compare the QoE CDF in Figure 8, where QA-UCB achieves a higher QoE for a larger fraction of users. For example, QA-UCB achieves an average QoE over 5 for 95% of the time whereas the next best policy (DQN) is only about 40%. Further, we observe a steeper CDF curve of QA-UCB compared to baselines, suggesting that it guarantees fairness among users since most users have similar average QoE.

Rebuffering. As discussed in Section 3, rebuffering greatly impacts QoE experienced by users. It is clear from Figure 9 that QA-UCB ensures lower rebuffering than the other policies under consideration.

Learning Regret. The learning regrets of QA-UCB and DQN are shown in Figure 10, where we use the Monte Carlo simulation with 10, 000 independent trails. To evaluate the regret of DQN, we

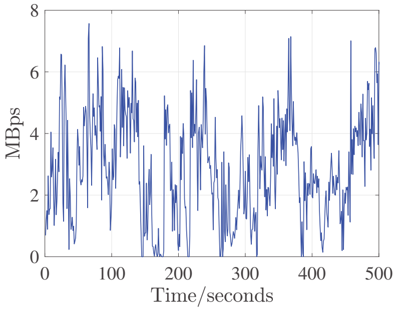

Figure 4: User's download speed with respect to time [34].

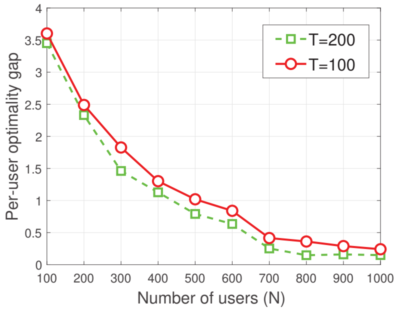

Figure 5: Asymptotic optimality of *Quality Index Policy.*

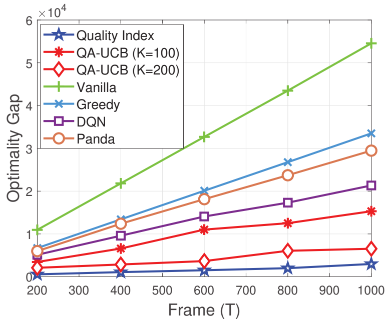

Figure 6: Comparison of the optimality gap.

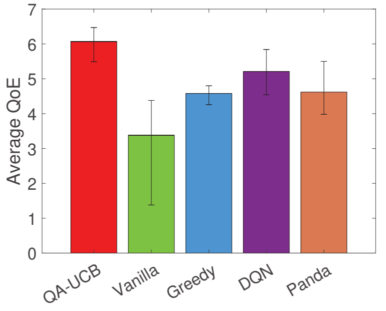

Figure 7: Average QoE.

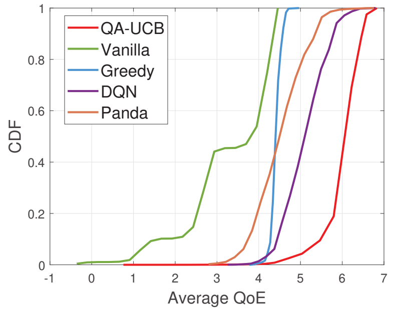

Figure 8: CDF of average QoE.

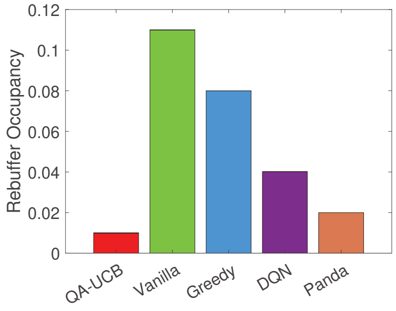

Figure 9: Rebuffering.

also modify the training horizon into episodes with each episode containing 100 frames as our QA-UCB setting. As shown in Figure 10, QA-UCB achieves a much smaller regret as compared with DQN. In particular, the accumulated regret of QA-UCB plateaus around 2×10^4 frames while that of DQN has a non-negligible increase.

Scalability. We also evaluate the scalability of our policies by increasing the number of users with a fixed number of frames $T = 250$ in Figure 11. As the number of users increases, the gap between the accumulated QoE achieved by QA-UCB and that achieved by solving LP (12)-(15) (see Proposition 1) keeps the same, while the gap becomes larger for DQN and Panda, suggesting that QA-UCB is more scalable to the network environments.

Consistency. As discussed in Section 3, the QoE of user n in (2) depends on the quality function $q_n(\cdot)$ and parameters α_n, β_n. We now show that the performance improvement of our QA-UCB over baselines is consistent across different settings. Specifically, we further consider $q_n(R) = 0.5R, \alpha_n = 1, \beta_n = 1, \forall n$ and $R \in \mathcal{R}$, under which QA-UCB still significantly outperforms other policies in shown Figure 12, similar to our observations in Figure 6.

7 CONCLUSION

We studied the problem of adaptively choosing bitrates and allocating network resources for maximizing the cumulative QoE of multiple users that are streaming videos over a shared wireless access edge network. Though it can be cast as a finite-horizon restless bandit problem, it is provably hard to solve. To circumvent this, we designed a computationally appealing `Quality Index Policy` that is provably asymptotically optimal. Since the wireless edge environment is highly dynamic with system parameters varying over time, we further proposed an index-aware RL algorithm dubbed as QA-UCB. We proved that QA-UCB achieves a sub-linear regret with a low-complexity since it fully leverages our proposed index policy for making decisions. To the best of our knowledge, this is the first work that designs a lightweight index-aware reinforcement learning policy with sub-linear regret in the context of adaptive video streaming with multiple users in wireless edge networks. We performed simulations using real-world video traces, and observed that our policies outperform conventional ones.

ACKNOWLEDGEMENTS

The work of G. Xiong and J. Li was supported in part by the NSF grants CRII-CNS-NeTS-2104880 and RINGS-2148309, and was supported in part by funds from OUSD R&E, NIST, and industry partners as specified in the Resilient & Intelligent NextG Systems (RINGS) program, as well as the DOE DE-EE0009341. The work of R. Singh was partially supported by the SERB Grant SRG/2021/002308, and PC 39010B. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

REFERENCES

[1] Zahaib Akhtar, Yun Seong Nam, Ramesh Govindan, Sanjay Rao, Jessica Chen, Ethan Katz-Bassett, Bruno Ribeiro, Jibin Zhan, and Hui Zhang. 2018. Oboe: Auto-Tuning Video ABR Algorithms to Network Conditions. In *Proc. of ACM SIGCOMM*.

[2] Eitan Altman. 1999. *Constrained Markov Decision Processes*. Vol. 7. CRC Press.

[3] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. 2002. Finite-Time Analysis of the Multiarmed Bandit Problem. *Machine Learning* 47, 2 (2002), 235–256.

[4] Konstantin Avrachenkov and Vivek S Borkar. 2020. Whittle Index Based Q-learning for Restless Bandits with Average Reward. *arXiv preprint*

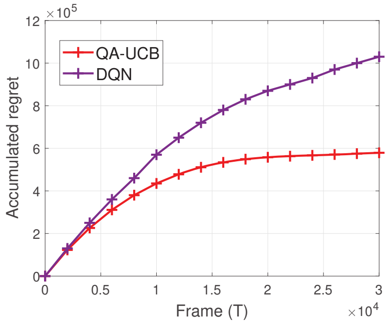

Figure 10: Learning regret.

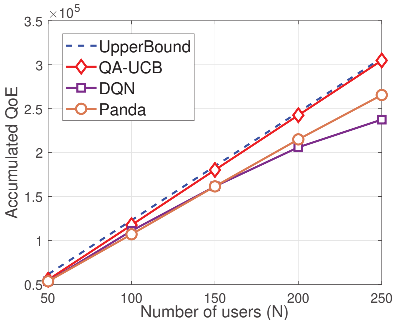

Figure 11: Scalability.

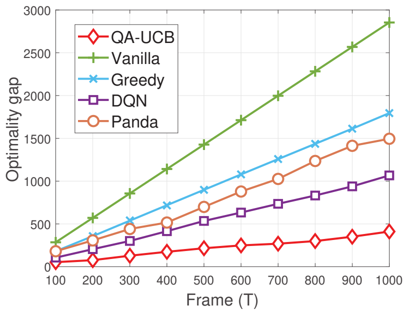

Figure 12: Consistent improvement.

arXiv:2004.14427 (2020).

[5] Dilip Bethanabhotla, Giuseppe Caire, and Michael J Neely. 2016. WiFlix: Adaptive Video Streaming in Massive MU-MIMO Wireless Networks. *IEEE Transactions on Wireless Communications* 15, 6 (2016), 4088–4103.

[6] Rajarshi Bhattacharyya, Archana Bura, Desik Rengarajan, Mason Rumuly, Srinivas Shakkottai, Dileep Kalathil, Ricky KP Mok, and Amogh Dhamdhere. 2019. Qflow: A Reinforcement Learning Approach to High QoE Video Streaming over Wireless Networks. In *Proc. of ACM MobiHoc*.

[7] Chao Chen, Robert W Heath, Alan C Bovik, and Gustavo de Veciana. 2013. A Markov Decision Model for Adaptive Scheduling of Stored Scalable Videos. *IEEE Transactions on Circuits and Systems for Video Technology* 23, 6 (2013), 1081–1095.

[8] Yonathan Efroni, Shie Mannor, and Matteo Pirotta. 2020. Exploration-Exploitation in Constrained MDPs. *arXiv preprint arXiv:2003.02189* (2020).

[9] Jing Fu, Yoni Nazarathy, Sarat Moka, and Peter G Taylor. 2019. Towards Q-Learning the Whittle Index for Restless Bandits. In *2019 Australian & New Zealand Control Conference (ANZCC)*. IEEE, 249–254.

[10] Chen Gong and Xiaodong Wang. 2013. Adaptive Transmission for Delay-Constrained Wireless Video. *IEEE Transactions on Wireless Communications* 13, 1 (2013), 49–61.

[11] Aditya Gopalan and Shie Mannor. 2015. Thompson Sampling for Learning Parameterized Markov Decision Processes. In *Proc. of COLT*.

[12] Yashuang Guo, Qinghai Yang, F Richard Yu, and Victor CM Leung. 2017. Dynamic Quality Adaptation and Bandwidth Allocation for Adaptive Streaming Over Time-Varying Wireless Networks. *IEEE Transactions on Wireless Communications* 16, 12 (2017), 8077–8091.

[13] David J Hodge and Kevin D Glazebrook. 2015. On the Asymptotic Optimality of Greedy Index Heuristics for Multi-Action Restless Bandits. *Advances in Applied Probability* 47, 3 (2015), 652–667.

[14] Weici Hu and Peter Frazier. 2017. An Asymptotically Optimal Index Policy for Finite-Horizon Restless Bandits. *arXiv preprint arXiv:1707.00205* (2017).

[15] Thomas Jaksch, Ronald Ortner, and Peter Auer. 2010. Near-Optimal Regret Bounds for Reinforcement Learning. *Journal of Machine Learning Research* 11, 4 (2010).

[16] Chi Jin, Tiancheng Jin, Haipeng Luo, Suvrit Sra, and Tiancheng Yu. 2019. Learning Adversarial MDPs with Bandit Feedback and Unknown Transition. *arXiv preprint arXiv:1912.01192* (2019).

[17] Krishna C Kalagarla, Rahul Jain, and Pierluigi Nuzzo. 2021. A Sample-Efficient Algorithm for Episodic Finite-Horizon MDP with Constraints. In *Proc. of AAAI*.

[18] Jonathan Kua, Grenville Armitage, and Philip Branch. 2017. A Survey of Rate Adaptation Techniques for Dynamic Adaptive Streaming Over HTTP. *IEEE Communications Surveys & Tutorials* 19, 3 (2017), 1842–1866.

[19] Qiao Lan, Bojie Lv, Rui Wang, Kaibin Huang, and Yi Gong. 2020. Adaptive Video Streaming for Massive MIMO Networks via Approximate MDP and Reinforcement Learning. *IEEE Transactions on Wireless Communications* 19, 9 (2020), 5716–5731.

[20] Stefan Lederer, Christopher Müller, and Christian Timmerer. 2012. Dynamic Adaptive Streaming over HTTP Dataset. In *Proc. of ACM MMSys*.

[21] Zhi Li, Xiaoqing Zhu, Joshua Gahm, Rong Pan, Hao Hu, Ali C Begen, and David Oran. 2014. Probe and Adapt: Rate Adaptation for HTTP Video Streaming at Scale. *IEEE Journal on Selected Areas in Communications* 32, 4 (2014), 719–733.

[22] Hongzi Mao, Ravi Netravali, and Mohammad Alizadeh. 2017. Neural Adaptive Video Streaming with Pensieve. In *Proc. of ACM SIGCOMM*.

[23] Andreas Maurer and Massimiliano Pontil. 2009. Empirical Bernstein Bounds and Sample Variance Penalization. *arXiv preprint arXiv:0907.3740* (2009).

[24] José Niño-Mora. 2007. Dynamic Priority Allocation via Restless Bandit Marginal Productivity Indices. *Top* 15, 2 (2007), 161–198.

[25] Ronald Ortner, Daniil Ryabko, Peter Auer, and Rémi Munos. 2012. Regret Bounds for Restless Markov Bandits. In *Proc. of Algorithmic Learning Theory*.

[26] Christos H Papadimitriou and John N Tsitsiklis. 1994. The Complexity of Optimal Queueing Network Control. In *Proc. of IEEE Conference on Structure in Complexity Theory*.

[27] Martin L Puterman. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons.

[28] Aviv Rosenberg and Yishay Mansour. 2019. Online Convex Optimization in Adversarial Markov Decision Processes. In *Proc. of ICML*.

[29] Iraj Sodagar. 2011. The MPEG-DASH Standard for Multimedia Streaming Over the Internet. *IEEE Multimedia* 18, 4 (2011), 62–67.

[30] Kevin Spiteri, Rahul Urgaonkar, and Ramesh K Sitaraman. 2020. BOLA: Near-Optimal Bitrate Adaptation for Online Videos. *IEEE/ACM Transactions on Networking* 28, 4 (2020), 1698–1711.

[31] Thomas Stockhammer. 2011. Dynamic Adaptive Streaming Over HTTP–Standards and Design Principles. In *Proc. of ACM MMSys*.

[32] Cisco Systems. 2019. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017–2022 White Paper. *[Online.] Available: https://s3.amazonaws.com/media.mediapost.com/uploads/CiscoForecast.pdf* (2019).

[33] Kexin Tang, Nuowen Kan, Junni Zou, Chenglin Li, Xiao Fu, Mingyi Hong, and Hongkai Xiong. 2021. Multi-user Adaptive Video Delivery over Wireless Networks: A Physical Layer Resource-Aware Deep Reinforcement Learning Approach. *IEEE Transactions on Circuits and Systems for Video Technology* 31, 2 (2021), 798–815.

[34] J. van der Hooft, S. Petrangeli, T. Wauters, R. Huysegems, P. R. Alface, T. Bostoen, and F. De Turck. 2016. HTTP/2-Based Adaptive Streaming of HEVC Video Over 4G/LTE Networks. *IEEE Communications Letters* 20, 11 (2016), 2177–2180.

[35] Ina Maria Verloop. 2016. Asymptotically Optimal Priority Policies for Indexable and Nonindexable Restless Bandits. *The Annals of Applied Probability* 26, 4 (2016), 1947–1995.

[36] Siwei Wang, Longbo Huang, and John Lui. 2020. Restless-UCB, an Efficient and Low-complexity Algorithm for Online Restless Bandits. In *Proc. of NeurIPS*.

[37] Richard R Weber and Gideon Weiss. 1990. On An Index Policy for Restless Bandits. *Journal of Applied Probability* (1990), 637–648.

[38] Peter Whittle. 1988. Restless Bandits: Activity Allocation in A Changing World. *Journal of Applied Probability* (1988), 287–298.

[39] Guojun Xiong, Jian Li, and Rahul Singh. 2022. Reinforcement Learning Augmented Asymptotically Optimal Index Policies for Finite-Horizon Restless Bandits. In *Proc. of AAAI 2022*.

[40] Guojun Xiong, Shufan Wang, Jian Li, and Rahul Singh. 2022. Model-free Reinforcement Learning for Content Caching at the Wireless Edge via Restless Bandits. *arXiv preprint arXiv:2202.13187* (2022).

[41] Guojun Xiong, Shufan Wang, Gang Yan, and Jian Li. 2022. Reinforcement Learning for Dynamic Dimensioning of Cloud Caches: A Restless Bandit Approach. In *Proc. of IEEE INFOCOM*.

[42] Xiaoqi Yin, Abhishek Jindal, Vyas Sekar, and Bruno Sinopoli. 2015. A Control-Theoretic Approach for Dynamic Adaptive Video Streaming Over HTTP. In *Proc. of ACM SIGCOMM*.

[43] Gabriel Zayas-Cabán, Stefanus Jasin, and Guihua Wang. 2019. An Asymptotically Optimal Heuristic for General Nonstationary Finite-Horizon Restless Multi-Armed, Multi-Action Bandits. *Advances in Applied Probability* 51, 3 (2019), 745–772.

[44] Chao Zhou, Chia-Wen Lin, and Zongming Guo. 2016. mDASH: A Markov Decision-based Rate Adaptation Approach for Dynamic HTTP Streaming. *IEEE Transactions on Multimedia* 18, 4 (2016), 738–751.

[45] Yihan Zou, Kwang Taik Kim, Xiaojun Lin, and Mung Chiang. 2021. Minimizing Age-of-Information in Heterogeneous Multi-Channel Systems: A New Partial-Index Approach. In *Proc. of ACM MobiHoc*.