Dynamic Resource Allocation for Streaming Services using Deep Reinforcement Learning: User-Centric Approach to Mobile Usage

Ha Eun Song, Yu Min Park, Choong Seon Hong
Department of Computer Science and Engineering
Kyung Hee university
Yongin, 446-701, Republic of Korea
{sh8455, yumin0906, cshong}@khu.ac.kr

***Abstract*—In this work, we propose a novel method to dynamically adjust bandwidth, power, and quality while considering the user's residual data availability in a video streaming environment utilizing cellular data. The method is designed to ensure the best video quality without exceeding the user's data usage limit through deep reinforcement learning. The main goal is to maximize the user's quality of experience (QoE) through efficient resource allocation. To validate the proposed method, we conducted comparative experiments with existing algorithms that do not consider data usage and a fixed resource allocation method. The experimental results show that the proposed method increases QoE by 55.01% compared to the methods that do not consider data usage and by 444.21% compared to the fixed resource allocation method, providing users with a superior streaming experience.**

***Index Terms*—Adaptive bitrate video streaming, computing, power allocation, bandwidth allocation and mobile data usage.**

I. Introduction

Multimedia communication services (e.g., Skype, FaceTime) and video streaming platforms (e.g., Hulu, YouTube, Netflix) have become an integral part of everyday life in the modern world. Mobile video accounts for more than half of all mobile data traffic worldwide, and this share is expected to continue to grow. While the introduction of newer wireless technologies, such as 4G LTE-Advanced, has dramatically expanded the bandwidth available to users, the emergence of new video formats such as ultra-high definition (UHD), high dynamic range (HDR), light fields, and new services such as virtual reality are further increasing the demand for bandwidth on networks. These new formats and services require high-quality video streaming to meet user expectations, but delivering this while maintaining the quality of experience (QoE) is a critical issue due to bandwidth and time constraints of real-time wireless transmission [1], [2].

Video streaming over wireless channels is a complex problem requiring high video quality and low transmission delay under limited communication resources and a rapidly changing environment. In most streaming applications, the video is a stored sequence encoded at a high bit rate, which must be adapted to the wireless channel. The resource utilization patterns of video streaming users are typically irregular. They are further complicated by stringent delay requirements, limited transmission power and bandwidth, and mutual interference [3]. Existing studies have mainly focused on fast download speeds or high-quality streaming. but these approaches need to consider the user's network situation and data allowance fully. In particular, mobile users often have a limited monthly data allowance when using cellular data, and ignoring these constraints can degrade the user experience. Therefore, this research explores optimizing download speed and quality dynamically, and hence the use of power and bandwidth. While the download speed of a chunk fluctuates depending on the user's network condition and the quality of the chunk, the playback time is constant regardless of the quality of the chunk, and we exploit these characteristics to distribute resources efficiently.

The goal of our algorithm is to improve the user's streaming experience while efficiently utilizing the network's resources. To achieve this, we monitor the user's mobile data usage in real-time and dynamically allocate power, bandwidth, and quality accordingly. By doing so, this work presents a comprehensive approach to improve the efficiency and user experience of mobile streaming simultaneously.

The remainder of this paper is organized as follows: Section II briefly reviews the previous work on which this study is based. Section III details the system model presented in this study and the associated optimization problem. In Section IV, we present our proposal centered on a Markov decision process (MDP). In Section V, we evaluate and analyze the performance of the proposed methodology. Finally, Section VI concludes with the main conclusions of this study.

This work was supported by Institute of Information and communications Technology Planning and Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2019-0-01287-005, Evolvable Deep Learning Model Generation Platform for Edge Computing), in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2023-00207816), and in party by Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korea Government (MSIT) (Artificial Intelligence Innovation Hub) under Grant 2021-0-02068-003,*Dr. CS Hong is the corresponding author.

979-8-3503-3094-6/24/$31.00 ©2024 IEEE

II. Related Works

Resource Allocation for Video Streaming. In [4], an integrated approach of caching, computing, and power allocation strategies was taken to optimize adaptive bitrate(ABR) video streaming, and mixed integer nonlinear programming(MINLP) techniques were used to simplify the optimization problem.

In [5], the dynamic allocation of bandwidth and transmit power using unmanned aerial vehicles (UAVs) as relay platforms in hotspot areas was solved using Lyapunov optimization techniques.

In [6], developed a strategy to improve the quality of experience (QoE) of mobile users through buffer management and bandwidth allocation strategies in a limited resource and communication environment and effectively solved the complex optimization problem using Lyapunov optimization and fundamental decomposition techniques.

Adaptive Bitrate Allocation for Video Streaming. In [7], they study to provide an optimal video streaming experience to viewers in MEC networks considering limited storage, computing power, and spectrum resources. In this study, they approach the problem of selecting different bitrate versions of a video segment as a non-convex optimization and mixed combinational problem and introduce a transcoding technique for this purpose.

In [8], a novel approach for efficient forward error correction(FEC) in real-time communication is presented. To overcome the limitations of traditional adaptive forward error correction(AFEC) algorithms, they introduce a QoE-oriented adaptive bitrate-FEC co-control algorithm called ABRF, which predicts the loss patterns in the network and utilizes a QoE model to simultaneously make suitable bitrate and FEC decisions for real-time video streaming.

III. System Model

Consider an orthogonal frequency division multiplexing (OFDM)-based communication system with a fixed OTT edge server and N users, indexed by $\mathcal{N} = \{1, 2, \ldots, N\}$, as shown in Figure 1. We assume that each user has different mobile usage and a fixed monthly allowance. Each user receives resource optimization at the same time slot $t \in \{1, 2, \ldots, T\}$, and the video is subdivided into $\mathcal{L} = \{1, 2, \ldots, L\}$ chunks. The user's location is denoted by x_t^i, y_t^i at time slot t, and the power, bandwidth, and quality are indexed byP_t^i, B_t^i, and Q_l^i ($\forall\ i \in \mathcal{N}, \forall\ t \in \mathcal{T}, \forall\ l \in \mathcal{L}$) respectively. Users use video streaming services according to their allocated resources. The server receives delayed information from the user's buffer and reallocates resources based on that information to optimize the user's streaming experience.

A. Buffer Model

One of the critical factors for optimizing the user's experience in video streaming is buffer management. According to the paper [9], The user starts receiving a chunk at time t_r, finishes downloading that chunk at t_p, and starts streaming. This results in an initial buffering time of $(t_p - t_r)$ seconds. The buffer will remain stable if the average data throughput equals the playback rate.

The server provides a set $Q_l^i = \{240p, 360p, 480p, 720p, 1080p, 1440p\}$ of different bitrate options for each video chunk. We assume the user can download the next chunk only after completely downloading chunk l at time t at a certain bit rate R_i. If the current buffer level is high, the user can select and download the highest quality chunk within the data usage limit; however, if the buffer level is low, the user must download chunks at a bit rate below the expected throughput.

We define the length of each chunk, i.e., the playback time, as L. The download time of the ith chunk is represented by D_i, which is calculated as the $chunksize(bits)/R_i$. According to the paper [9], the buffer OFF period after downloading the ith chunk is expressed as follows:

$$\delta_i = \max(\max(b_i - D_i, 0) + L - b_{\text{max}}, 0) \tag{1}$$

The buffer OFF period is when the buffer is empty during streaming. where b_{max} is the total size of the buffer, b_i is the buffer size when the ith chunk starts downloading at a certain bit rate R_i. The rebuffering time after downloading the ith chunk is expressed as follows:

$$\gamma_i = \max(D_i - b_i, 0) \tag{2}$$

The rebuffering time is required for the buffer to be refilled during streaming. The size of the following buffer can be expressed as follows:

$$b_{\text{(i+1)}} = \max(\max(b_i - D_i, 0) + L - \delta_i, 0) \tag{3}$$

B. QoE Metric

A key metric for measuring user satisfaction in video streaming services is QoE. Users consume streaming services based on their allocated resources, which results in a specific QoE. This allows service providers to optimize the user's experience and explore ways to improve the quality of their service. As streaming occurs in chunks, the QoE consists of the following elements **The quality of the video, the rate of change in quality, the length of the buffer, the rate of change in the buffer, the amount of data the user has left**. Putting these factors together, the defined value for each user's QoE is given by the expression [10]:

$$Quality = q_{\text{(t)}} \tag{4}$$

$$Quality\ Diff = q_{\text{(t+1)}} - q_{\text{(t)}} \tag{5}$$

$$Buffer = b_{\text{(t)}} \quad \text{if}(b_{\text{(t)}} > 0) \tag{6}$$

$$Buffer\ Diff = b_{\text{(t+1)}} - b_{\text{(t)}} \tag{7}$$

$$Data\ Availavility = P(a_{\text{(t)}}) \tag{8}$$

This is important for users with limited data packages because excessive data usage can incur additional costs or service interruptions. The penalty $P(x)$ helps to balance the quality of

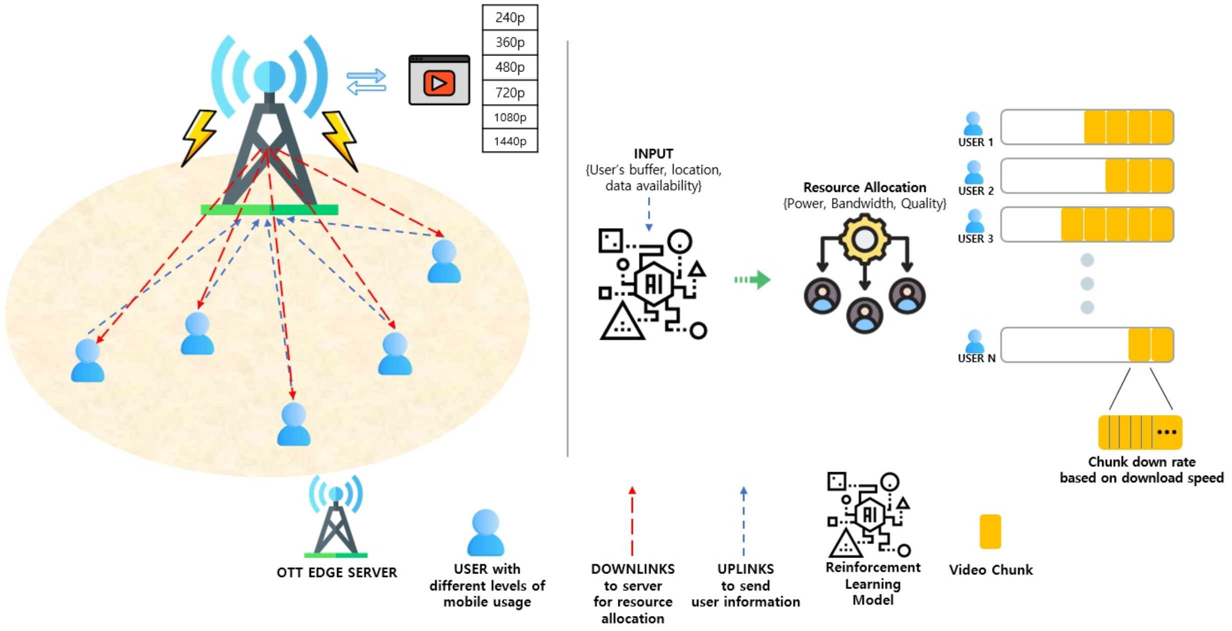
240p
360p
480p
720p
1080p
1440p
INPUT
{User's buffer, location, data availability}
Resource Allocation
{Power, Bandwidth, Quality}
USER 1
USER 2
USER 3
USER N
Chunk down rate based on download speed
OTT EDGE SERVER
USER with different levels of mobile usage
DOWNLINKS to server for resource allocation
UPLINKS to send user information
Reinforcement Learning Model
Video Chunk

Fig. 1. System Model. By allocating mobile resources based on the user's location and determining the quality of chunks based on the remaining mobile availability, it provides users with optimal download speeds and increases QoE.

streaming with data usage by penalizing when data availability falls below or exceeds a specific threshold interval.

$$P(a) = \begin{cases} -50, & \text{if } a_t^i < a_{\text{low}} \\ 0, & \text{if } a_{\text{low}} \le a_t^i \le a_{\text{high}} \\ -50, & \text{if } a_{\text{high}} < a_t^i \end{cases} \tag{9}$$

The expression to get the QoE from the above expressions is as follows:

$$\begin{aligned} QoE = {} & a_1 Quality + a2(1 - Quality\ Diff) \\ & + a3Buffer + a4Buffer\ Diff \\ & + a5Data\ Availability - a6latency \end{aligned} \tag{10}$$

where a1-a5 are the weights, latency is the sum of the user's download time, rebuffering time, and buffer off time in a single time step.

C. Communication Model

This section describes the communication model, focusing on the relationship between the user's remaining data and availability a_i, the distance d_i between user i and the server, the path loss model, the channel gain, and the user's data rate. The distance between user i's horizontal coordinates $\{x_i, y_i\}$ and the server's horizontal coordinates $\{x_b, y_b\}$ is:

$$d_i^b = \sqrt{(x_i - x_b)^2 + (y_i - y_b)^2},\ \forall i \in N \tag{11}$$

After that, the channel gain between user device i and server b is calculated according to [11] using a free-space path loss model.

$$g_i^b = \frac{g_0}{(d_i^b)^\theta},\ \forall i \in N \tag{12}$$

where g_o denotes the channel gain at the reference distance $d_0 = 1m$, and θ is the path loss exponent.

To analyze the overall performance of a communication system under simple assumptions, we typically set the channel gain to a degree that maintains the signal loss of the system under average circumstances. The data rate R_i for user i can be calculated:

$$R_i^t = B_i^t \log_2(\frac{1 + P_i^t g_i^b}{N_0})\ , \forall i \in N, \forall t \in T \tag{13}$$

Where N_0 is the interference power.

D. Problem Formulation

In this section, we formulate the optimization problem, where the main objective is maximizing the average QoE for each user.

$$\begin{aligned} \textbf{P1:} \ \underset{\boldsymbol{P},\boldsymbol{B},\boldsymbol{K}}{\text{maximize}} \quad & \sum_{t=1}^{T}\sum_{i=1}^{N} QoE_n(t) & \text{(14a)} \\ \text{subject to} \quad & \sum_{i=1}^{N} P_i \le P_{\text{total}},\ \forall t \in \mathcal{T} & \text{(14b)} \\ & \sum_{i=1}^{N} B_i \le B_{\text{total}},\ \forall t \in \mathcal{T} & \text{(14c)} \\ & R_i \ge R_{q_k},\ \forall i \in \mathcal{N},\ \forall t \in \mathcal{T} & \text{(14d)} \\ & A_{\text{low}} \le A_i \le A_{\text{high}},\ \forall i \in \mathcal{N} & \text{(14e)} \end{aligned}$$

Constraint (15a) ensures that the sum of the power allocated to all users does not exceed the total power of the BS, P_{total}. Constraint (15b) ensures that the sum of the bandwidth allocated to all users does not exceed the total bandwidth B_{total} of the BS. Constraint (15c) guarantees that each user's data throughput R_i is greater than or equal to the required bit rate R_{q_k} of the video chunk. Constraint (15d) guarantees that each user's current remaining data availability A_i has a capacity within a threshold.

IV. Suggestions

In this study, we propose a system that leverages deep reinforcement learning to maximize the user's QoE in a video streaming environment while efficiently managing power, bandwidth, and data availability. The proposed system can maximize QoE by ensuring optimal download speed without exceeding the user's data availability and providing a smooth and precise viewing experience.

A. Defining a Markov Decision Process (MDP)

In this paper, we use Proximal Policy Optimization(PPO) for optimal mobile resource allocation for high QoE. The MDP for reinforcement learning is as follows.

- **State:** In reinforcement learning, the state is information about the current state of the environment that plays an important role in determining what the agent will do. In this system, the state is the user's location in the current step, the availability of remaining data, and the allocated quality and buffer of the previous step.

$$s(t) = \{x_i^t, y_i^t, a_i^t, b_i^{t-1}, q_i^{t-1}\} \tag{16}$$

- **Action:** The action space represents the possible choices for each variable. In this system, bandwidth and power are normalized to values between 0 and 1; for each interval, they are replaced by their actual values. Therefore, the action space is defined as follows:

$$\begin{aligned} a(t) = \{(b, p, q) \mid b \in [0,1], p \in [0,1], \\ q \in [240p, 360p, 480p, 720p, 1080p, 1440p]\} \end{aligned} \tag{17}$$

- **Reward:** Reward is the feedback an agent receives when it performs a specific action in a particular state. It is used as a criterion to determine whether the action is good or bad and to learn the optimal policy. In this system, the reward is the sum of each user's QoE, calculated from the buffer size, chunk quality, and remaining data availability in the current and previous phases.

$$R(a(t), s(t)) = \{\sum_{i=1}^{N} QoE_i^t, \ \forall N\} \tag{15}$$

PPO is an algorithm that performs policy optimization. Unlike other algorithms, it has high sample efficiency and learns quickly and reliably by using a clipping mechanism to limit the difference from the previous policy when updating it. In this experiment, we used PPO and A2C because the action space is non-discrete, and the algorithm proposed in this paper is shown in Algorithm 1.

Algorithm 1 Proposed Mobile Resource Allocation algorithm

```
Initialize each user $a_i, x_i, y_i$ and class instance variables
  and calculate distance $d_i^b$
Execute action $a(t)$ according to policy $\pi_\theta(a|s(t))$
Reshape action $a(t)$ to the number of users $N$
for every time slot $t$ do
    for Iteration = 1 to $N$ do
        Calculate Data Rate $R_i$
        if The user has finished downloading all the chunks then
            skip download
        end if
        if buffer length $b_i$ is as full as the buffer is available then
            Playback of downloaded chunks starts a FIFO scheme
        end if
        Put a fully downloaded chunk from the time slot $t$ into the $b_i$
        Calculate remaining mobile exhaustion $a_i^t$
        Obtain reward $R(s(t), a(t))$ and observe the new state $s(t+1)$
    end for
end for
if All $N$ users have finished downloading then
    Obtain the final reward $R(s(t), a(t))$ for all $N$
end if
```

TABLE I
PARAMETERS FOR SIMULATION

Parameters	Description	Value
N_0	Additive white gaussian noise power	-174 dBm
g_0	Channel Gain	50 dB
B_{BS}	Bandwidth of BS	1MHz
P_{BS}	Power of BS	1W

V. Simulation Results

A. Simulation Setting

For the simulation, we assume that there are N users within a coverage area of 1km of an OTT server located at coordinates [0,0], each requesting a different video and that the user's location changes every time step due to the user's movement. In addition, each user has a different amount of remaining mobile data, and the parameters used in the experiment are shown in Table 1. We Assume that the Max Buffer Length b_{max} is 15 seconds, the number of users N is 3, and the number of video chunks L is 20.

We compared our proposed algorithms with the following two benchmarks:

- **Resource Allocation Algorithm with A2C:** Train the proposed algorithm as an A2C agent.
- **Algorithm to assign only high definition:** Static resource allocation algorithms that consistently assign only high quality to users.

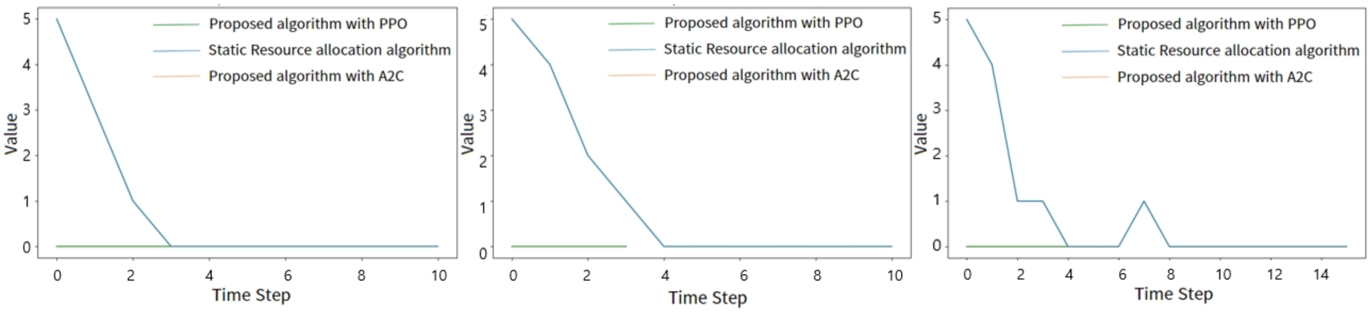

Fig. 2. Latency Comparison, The length of the graph is the number of time steps it took to download the chunk. The proposed algorithm using PPO and A2C has a low download time and no delay. Still, the static allocation algorithm, which allocates the same bandwidth and power to all users and serves only high-quality chunks, has a high delay due to the long time to fill the buffer after starting streaming. The chunks are not downloaded smoothly to the buffer even after starting playback, increasing the delay by 3% to 5%

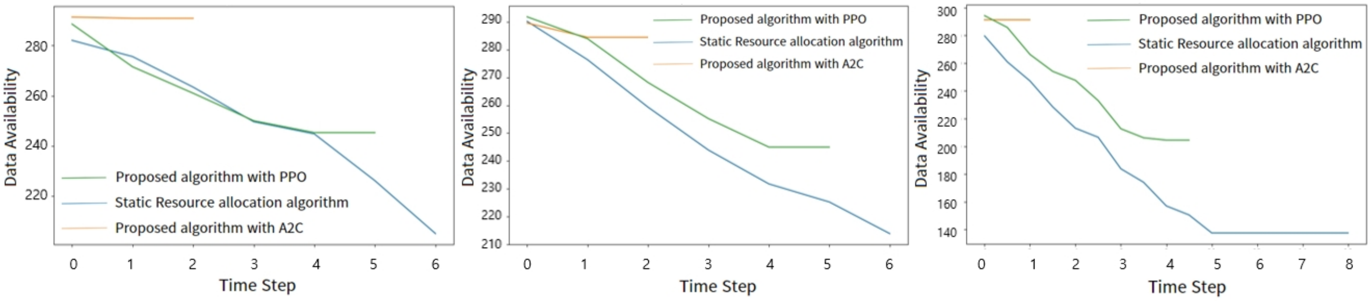

Fig. 3. Data Availability Comparison, The length of the graph is the number of time steps it took to download the chunk. The proposed PPO algorithm provides users with high-quality chunks based on the user's network conditions. When the residual data availability decreases and reaches a certain point, the quality can be adjusted to provide flexibility in data usage, while the proposed algorithm using A2C only provides low-quality chunks, achieving low QoE with little reduction in residual data usage, and the static allocation algorithm does not consider the user's data rate and only provides high-quality chunks, which quickly exhausts the user's mobile allowance.

B. Performance Evaluation

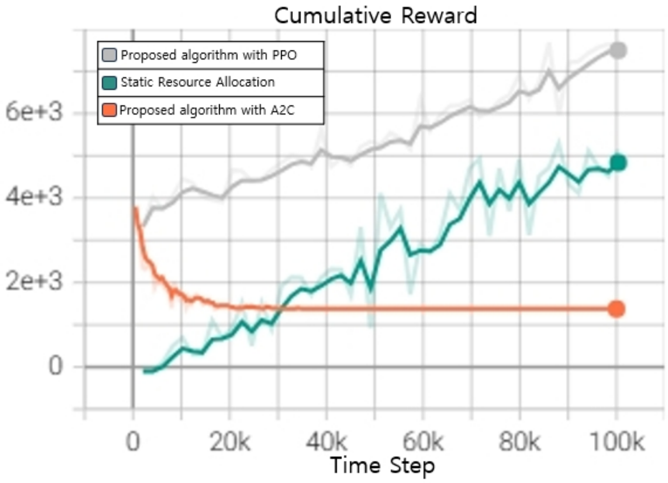

Fig. 4. Cumulative Reward, The performance of proposed algorithms using different methods in terms of cumulative rewards. The figure shows the performance of a proposed algorithm trained with PPO, a proposed algorithm trained with A2C, and a static resource allocation algorithm that consistently rewards only high quality. The experimental results show that the proposed method improves QoE by 55.01% compared to the method that does not consider data usage and by 444.21% compared to the static resource allocation method, providing users with a superior streaming experience.

Figure 4 compares the performance of suggestion algorithms using different methods in terms of cumulative rewards. This figure shows the performance of a suggestion algorithm trained with PPO, a suggestion algorithm trained with A2C, and a static resource allocation algorithm that consistently rewards only high quality. The cumulative reward is an essential metric to evaluate each algorithm's quality of experience (QoE). The graph shows that the proposed algorithm with PPO achieves significantly higher cumulative reward than the other algorithms and provides optimal QoE. PPO generally uses a clipping mechanism to improve data efficiency and sample reuse and provides stable learning. At the same time, A2C is relatively simple but has low data efficiency due to no sample reuse. Due to this difference, PPO achieves higher QoE values with more stable learning.

Figure 2 visualizes the incidence of rebuffering by algorithm, an essential factor that can have a significant impact on the viewer experience. As shown in the figure, the suggestion algorithm trained with A2C downloads chunks with lower quality than the user's data rate. In contrast, the suggestion algorithm trained with PPO gives the maximum quality without exceeding the user's data rate, so it does not cause rebuffering, resulting in no latency. On the other hand, an algorithm that consistently grants only high quality will download chunks at a slower rate because it grants quality that exceeds the user's data rate, resulting in a longer buffer-off time at the beginning of the download and continuous rebuffering thereafter.

Figure 3 compares the user's mobile residual usage by algorithm. The suggestion algorithm trained with A2C downloads chunks at a consistently lower quality, so downloads end quickly, and mobile data usage doesn't fluctuate as much. In contrast, the suggestion algorithm trained with PPO downloads chunks at the highest possible quality based on the user's data speed. Once the mobile residual usage reaches a certain level, it lowers the download quality to prevent the residual from being exhausted too quickly. Due to these characteristics, the proposed algorithm can efficiently manage the user's data usage while maintaining high experience quality. However, for an algorithm that only assigns consistently high quality, we observe that the monthly data residual quickly depletes and converges to zero because it does not consider the user's data speed and residual mobile usage.

VI. Conclusion

In this paper, we propose to use deep reinforcement learning in a video streaming environment using cellular data to allocate bandwidth and power considering the user's location and network conditions and to allocate quality considering mobile residual usage dynamically. By optimizing the quality of experience (QoE), which is an essential requirement in video streaming, the proposed algorithm can provide high QoE to users with fixed monthly mobile usage. Simulation results show that, compared with the benchmark algorithm, the proposed algorithm provides high QoE regarding rebuffering and mobile usage management and provides users with a balanced download and viewing experience. This dynamic resource allocation strategy is expected to improve streaming performance in wireless network environments. In future work, we will conduct further experiments and analysis to improve the performance of the proposed algorithm further and verify its practicality in different network environments and user scenarios.

References

[1] N. Barman and M. G. Martini, "Qoe modeling for http adaptive video streaming–a survey and open challenges," *Ieee Access*, vol. 7, pp. 30 831–30 859, 2019.

[2] S. Afzal, V. Testoni, C. E. Rothenberg, P. Kolan, and I. Bouazizi, "A holistic survey of multipath wireless video streaming," *Journal of Network and Computer Applications*, vol. 212, p. 103581, 2023.

[3] J. Huang, Z. Li, M. Chiang, and A. K. Katsaggelos, "Joint source adaptation and resource allocation for multi-user wireless video streaming," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 5, pp. 582–595, 2008.

[4] W. Liu, H. Ding, H. Zhang, and D. Yuan, "Low-latency oriented resource allocation for mec-assisted adaptive bitrate video streaming," *IEEE Transactions on Vehicular Technology*, 2023.

[5] Y. Chen, H. Zhang, and Y. Hu, "Optimal power and bandwidth allocation for multiuser video streaming in uav relay networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 6644–6655, 2020.

[6] N. Li, Y. Hu, Y. Chen, and B. Zeng, "Lyapunov optimized resource management for multiuser mobile video streaming," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 6, pp. 1795–1805, 2018.

[7] C. Liu, H. Zhang, H. Ji, and X. Li, "Mec-assisted flexible transcoding strategy for adaptive bitrate video streaming in small cell networks," *China Communications*, vol. 18, no. 2, pp. 200–214, 2021.

[8] S. Cheng, H. Hu, and X. Zhang, "Abrf: Adaptive bitrate-fec joint control for real-time video streaming," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[9] Y.-m. Kang, "An energy-aware buffer-based video streaming optimization scheme." *Journal of the Korea Institute of Information & Communication Engineering*, vol. 26, no. 10, 2022.

[10] S. Jung and K. Lim, "Design of dqn reward function for improving dash performance," *Dissertation for Master of Engineering, Kyungpook National University*, 2020.

[11] Y. K. Tun, K. T. Kim, L. Zou, Z. Han, G. Dán, and C. S. Hong, "Collaborative computing services at ground, air, and space: An optimization approach," *IEEE Transactions on Vehicular Technology*, 2023.