Personalized QoE Enhancement for Adaptive Video Streaming: A Digital Twin-Assisted Scheme

Xinyu Huang*, Conghao Zhou*, Wen Wu†, Mushu Li*, Huaping Wu‡, and Xuemin (Sherman) Shen*

*Department of Electrical & Computer Engineering, University of Waterloo, Canada
†Frontier Research Center, Peng Cheng Laboratory, China
‡Department of Electrical and Software Engineering, University of Calgary, Canada

Email: {x357huan, c89zhou, m475li, sshen}@uwaterloo.ca, wuw02@pcl.ac.cn, huaping.wu1@ucalgary.ca

Abstract—In this paper, we present a digital twin (DT)-assisted adaptive video streaming scheme to enhance personalized quality-of-experience (PQoE). Since PQoE models are user-specific and time-varying, existing schemes based on universal and time-invariant PQoE models may suffer from performance degradation. To address this issue, we first propose a DT-assisted PQoE model construction method to obtain accurate user-specific PQoE models. Specifically, user DTs (UDTs) are respectively constructed for individual users, which can acquire and utilize users’ data to accurately tune PQoE model parameters in real time. Next, given the obtained PQoE models, we formulate a resource management problem to maximize the overall long-term PQoE by taking the dynamics of users’ locations, video requests, and buffer statuses into account. To solve this problem, a deep reinforcement learning algorithm is developed to jointly optimize the segment version selection, and communication and computing resource allocation. Simulation results on the real-world dataset demonstrate that the proposed scheme can effectively enhance PQoE compared with benchmark schemes.

I. INTRODUCTION

With the rapid popularization of emerging video applications, video streaming data accounts for the majority of global mobile data [1], which has placed a growing strain on wireless networks. To guarantee users’ continuous playback under dynamic channel conditions, the adaptive bitrate (ABR) technology [2] splitting a complete video sequence into multiple segments with different bitrates enables adaptive video streaming. However, traditional ABR schemes usually adopt users’ predicted throughput as the principle of bitrate selection, which may not improve users’ personalized watching experience that depends on multiple factors [3]. To handle this issue, the personalized quality-of-experience (PQoE) model is proposed, which can characterize the user-specific perception of different QoE factors, including rebuffer time, video quality, and quality variation [4]. Based on the PQoE model, appropriate segments and network resources are allocated to the user to enhance its personalized watching experience.

Recently, significant research efforts have been put to construct the PQoE model. Wang et al. proposed a method of constructing PQoE model that combined the user’s service preference and network layer configuration, which can effectively reflect the user’s resource demands [4]. Gao et al. enhanced the PQoE model accuracy by leveraging sensing information and contextual information [5]. To solve the PQoE-enhanced resource allocation problem, a deep reinforcement learning (DRL)-based scheme was proposed in [6]. The above works focus on time-invariant PQoE models. However, PQoE models may vary across different video contents and playback statuses, such as buffer occupancy, video quality, and quality variation. Such dynamics make the previous constructed accurate PQoE models outdated over time, thereby rendering sub-optimal resource management decisions and degrading PQoE performance. Hence, constructing a real-time and accurate PQoE model is paramount.

To tackle this challenge, the digital twin (DT) technology is a potential solution. DT is a digital representation of a physical entity (PE) that can accurately reflect its status and feature via a real-time synchronization between DT and PE [7]. DT technology has been widely applied to fault diagnosis and predictive maintenance [8]. Inspired by these applications, we consider to leverage DT technology to store and analyze users’ video streaming data, so that accurate user-specific PQoE models can be adjusted in real time and then utilized to make resource management decisions for enhancing PQoE.

In this paper, we present a DT-assisted adaptive video streaming scheme to enhance PQoE. Specifically, we propose a DT-assisted PQoE model construction method to tune PQoE model parameters in real time, and design a tailored DRL-based algorithm for resource management. Firstly, user DTs (UDTs) are established to construct real-time PQoE models by storing and analyzing users’ video streaming data. The constructed PQoE models adopt a linear weighting combination of QoE factors, and incorporate the Ebbinghaus memory effect to characterize the fading effect of users’ watching experience. The personalization is reflected in PQoE model parameters, i.e., the relative memory length and the sensitivity degree of QoE factors. The PQoE model parameters are tuned in real time through a data fitting method. Secondly, we develop a DRL-based resource management algorithm to efficiently facilitate the PQoE-oriented adaptive video streaming. The objective is to maximize the overall long-term PQoE considering the dynamics of users’ locations, video requests, and buffer statuses, by jointly optimizing the segment version selection, and communication and computing resource allocation. To reduce the algorithm’s training complexity, we narrow down the action dimension by splitting the range of transmission and transcoding variables into multiple parts to represent the
We propose a DT-assisted PQoE model construction method, which can obtain user’s real-time and accurate PQoE model parameters.

We develop a DRL-based resource management algorithm to jointly determine the segment version selection, and communication and computing resource allocation.

The remainder of this paper is organized as follows. The DT-assisted adaptive video streaming scheme is presented in Section II. The DRL-based resource management algorithm is proposed in Section III. Simulation results are provided in Section IV, followed by the conclusion in Section V.

II. DT-ASSISTED ADAPTIVE VIDEO STREAMING SCHEME

A. System Model

As shown in Fig. 1, we consider the DT-assisted adaptive video streaming framework consisting of three parts: PEs, DTs, and interaction links.

- PEs: PEs consist of a base station (BS), an edge server, a cloud server, and multiple users. The sets of users and video sequences are denoted by \( U \) and \( F \), respectively, where \( u \) and \( f \) represent the corresponding indexes. Video sequences are encoded into multiple segments with different versions. The set of segment versions is denoted by \( L = \{1, ..., l, ..., L\} \). The edge server caches popular videos for low-latency content delivery due to limited storage capacity, while the cloud server caches all videos. PEs request adaptive video streaming, the network controller de-
Here, the decision variable $o_{kt,u}^l \in \{0, 1\}$ indicates whether segment $kt,u$ of version $l$ can be obtained by transcoding. If segment $kt,u$ of version $l$ can be obtained by transcoding, $o_{kt,u}^l = 1$; Otherwise, $o_{kt,u}^l = 0$. The computing intensity for transcoding per unit file size is denoted by $\mu$, and $c_t$ is the computing capacity of the edge server at scheduling slot $t$. Here, the decision variable $\omega_{t,u} \in [0, 1]$ represents the ratio of computing resources allocated to user $u$ at scheduling slot $t$.

**Case 3:** If the segment is fetched from the cloud server, the service delay $D_{t,u}^{(3)}$ that refers to the transmission delay from the cloud server to the BS, and then to user $u$, is given by

$$D_{t,u}^{(3)} = \sum_{l=1}^L g_{kt,u}^l (1 - \chi_{kt,u}) (1 - o_{kt,u}^l) \left( \frac{s_{kt,u}^l}{\rho^C} + \frac{s_{kt,u}^l}{\rho^B} \right),$$

where $\rho^C$ is the transmission capacity between the cloud server and the BS. Due to the wired link connection between the cloud server and the BS, $\rho^C$ is assumed to be a constant.

Taking three independent cases into account, the total service delay, $D_{t,u}$, can be represented by

$$D_{t,u} = D_{t,u}^{(1)} + D_{t,u}^{(2)} + D_{t,u}^{(3)},$$

which impacts the rebuffer time in the PQoE model.

**C. PQoE Model**

The PQoE model consists of the following three factors.

**Rebuffer time:** The rebuffer time is related to service delay (as analyzed in Eq. (4)) and playback buffer occupancy. The playback buffer occupancy at the current scheduling slot depends on the playback buffer occupancy at the previous scheduling slot, the current received segment time length, and the scheduling slot length. Let $B_{t,u}$ denote user $u$’s playback buffer occupancy at scheduling slot $t$, which is updated via

$$B_{t+1,u} = \left( B_{t,u} + \sum_{l=1}^L g_{kt,u}^l e - d \right)^+.$$

Here, $e$ is the time length of segment, and $d$ is the scheduling slot length. The function $(x)^+ = \max\{x, 0\}$.

When the user’s service delay exceeds the current buffer occupancy, the rebuffering event occurs. Therefore, the rebuffer time of user $u$ at scheduling slot $t$ is defined by

$$R_{t,u} = (D_{t,u} - B_{t,u})^+.$$

**Video quality:** Since one video sequence consists of multiple segments with different video qualities, the higher video quality can usually bring the higher PQoE to user. Here, we select the peak signal-to-noise ratio (PSNR), which is a common metric to characterize the distortion level of a segment [12]. At scheduling slot $t$, the video quality of the new segment added to user $u$’s playback buffer is defined by

$$V_{t,u} = \sum_{l=1}^L g_{kt,u}^l \varphi_{kt,u}^l.$$

1Note that if a user stops watching the current video and switches to another video, video packets in the current playback buffer will not be used and the playback buffer occupancy will be empty.

2The video sequence is encoded into multiple equal-length segments.

Here, $\varphi_{kt,u}^l$ is the PSNR of segment $kt,u$ of version $l$.

**Quality variation:** Since the content-coding complexity of adjacent segments is different, segments with the same version also exhibits different PSNRs [9]. However, in this case, a user cannot feel the quality variation. Therefore, we use the switching magnitude of segment versions to represent the quality variation, which is defined by

$$H_{t,u} = \sum_{l=1}^L g_{kt,u}^l |l_{kt,u} - l_{kt,u} - 1|.$$

Based on these factors, we can formulate the PQoE model. Specifically, since video playback is usually a long-term process for users, the impact of previous bad or good watching experiences, such as rebuffer time, video quality, and quality variation, on users’ current watching experience decreases gradually. Therefore, we incorporate the Ebbinghaus memory effect [10] into the linear weighting combination of PQoE factors [3] to obtain an accurate PQoE model, i.e.,

$$Z_{t,u}(g_{kt,u}, o_{kt,u}, \omega_{t,u}, \xi_{t,u}) = \exp \left( - \frac{(p_u - t)}{\lambda_{t,u}} \right) [\alpha_{t,u} V_{t,u} - \beta_{t,u} H_{t,u} - \gamma_{t,u} R_{t,u}],$$

where $g_{kt,u} = \{g_{kt,u}^l\}_{l \in L}$ and $o_{kt,u} = \{o_{kt,u}^l\}_{l \in L}$. Here, $\lambda_{t,u}$ is the relative memory length of user $u$ at scheduling slot $t$, and $p_u$ is the requested video sequence length. Here, $\alpha_{t,u}$, $\beta_{t,u}$, and $\gamma_{t,u}$ are user $u$’s sensitivity degrees of QoE factors at scheduling slot $t$. A relatively small $\alpha_{t,u}$ indicates that user $u$ is not particularly concerned about video quality, while a large $\alpha_{t,u}$ means that more communication resources need to be allocated to transmit the segment of high video quality. A large $\beta_{t,u}$, relatively to other parameters, indicates that user $u$ is deeply concerned about quality variation. The segment with suitable version needs to be transmitted to achieve a smoother change of video quality. In cases where user $u$ prefers low rebuffer time, a large $\gamma_{t,u}$ should be used.

**D. DT-Assisted PQoE Model**

To obtain user’s PQoE model parameters $\lambda_{t,u}$, $\alpha_{t,u}$, $\beta_{t,u}$, and $\gamma_{t,u}$ in real time, we utilize the UDT to analyze the user’s data related to the adaptive video streaming. The user’s engagement time is employed by UDT to calculate the objective PQoE reference value [11]. Specifically, assume that a user only stops watching the video due to the long rebuffer time, low video quality, and frequent quality variation. If the proportion of the user’s engagement time to the total video playback time is high, the user is insensitive to PQoE factors. Correspondingly, the PQoE is high; Otherwise, the PQoE is low. Therefore, the objective PQoE reference value is defined by

$$Z_{\tilde{u},f}^\text{ref} = \frac{5 \cdot \eta_{\tilde{u},f}}{K_f \cdot e + R_{\tilde{u},f}},$$

where $\eta_{\tilde{u},f}$ is the engagement time of user $\tilde{u}$ watching video $f$. Here, $\tilde{u}$ is the UDT index corresponding to user $u$. The parameter $K_f$ is the total number of segments of video $f$, and $R_{\tilde{u},f}$ is the total rebuffer time for user $\tilde{u}$ watching video $f$, which can be calculated based on the historical statistics.
on $R_{l,u}$. The PQoE range is from 0 to 5, aligning with the common principle [12]. To build the mapping relationship between $Z_{l,u}$ and $Z_{l,u}^e$, we need to accumulate the $Z_{l,u}$ of video $f$ in the time domain. Since the established PQoE model is a non-linear function, we employ the nonlinear regression method to obtain the PQoE model parameters. The input is a user’s historical buffer time, video quality, and segment bitrate: $s_t = \{\{R_{l,u}\} u \in \mathcal{U}, \{B_{l,u}\} u \in \mathcal{U}, \{l_{t,u}\} u \in \mathcal{U}, \{V_{l,u}\} u \in \mathcal{U}, \{s_{t,u}\} u \in \mathcal{U}\}$. (12)

**State:** The state includes users’ rebuffer time, buffer occupancy, segment version, video quality, and segment bitrate: $s_t = \{\{R_{l,u}\} u \in \mathcal{U}, \{B_{l,u}\} u \in \mathcal{U}, \{l_{t,u}\} u \in \mathcal{U}, \{V_{l,u}\} u \in \mathcal{U}, \{s_{t,u}\} u \in \mathcal{U}\}$. (13)

**Action:** The action includes all optimization variables in Eq. (11), which is defined by

$$a_t = \{\{g_{t,u}\} u \in \mathcal{U}, \{o_{t,u}\} u \in \mathcal{U}, \{\omega_{t,u}\} u \in \mathcal{U}, \{\xi_{t,u}\} u \in \mathcal{U}\}. \quad \text{(14)}$$

**State Transition Probability:** At each step, the current segment version selection, and communication and computing resource allocation affect users’ next state. Therefore, the state transition probability between $s_t$ and $s_{t+1}$ is given by

$$\Pr(s_{t+1} = s_{t+1} | s_t = s_t, a_t = a_t) = \prod_{u \in \mathcal{U}} \Pr(R_{l_{t+1}} = s_{t+1} | R_{l_t} = s_t, g_{t,u} = g_t, o_{t,u} = o_t, \omega_{t,u} = \omega_t, \xi_{t,u} = \xi_t, \omega_{t,u} = \omega_t). \quad \text{(15)}$$

### III. DRL-Based Resource Management Algorithm

#### A. Problem Formulation

Our objective is to maximize the overall long-term PQoE over $T$ scheduling slots. Correspondingly, the optimization problem is formulated as

$$\begin{align*}
\text{P}_0: \max_{(g_t, o_t, \omega_t, \xi_t)} & \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{u \in \mathcal{U}} \tilde{Z}_{t,u}(g_{t,u}, o_{t,u}, \omega_{t,u}, \xi_{t,u}) \\
\text{s.t.} & \sum_{l=1}^{L} g_{l_t,u} \leq 1, g_{l_t,u} \in \{0, 1\}, \\
& \sum_{l=1}^{L} o_{l_t,u} \leq 1, o_{l_t,u} \in \{0, 1\}, \\
& \sum_{l=1}^{L} \omega_{l_t,u} \leq 1, \omega_{l_t,u} \in \{0, 1\}, \\
& \sum_{u \in \mathcal{U}} \xi_{t,u} \leq 1, \xi_{t,u} \in \{0, 1\}. \quad \text{(11)}
\end{align*}$$

Constraint (11a) guarantees that a user can only receive a segment with one version at each scheduling slot. Constraint (11b) guarantees that a segment can only be transcoded for one target version for a user at each scheduling slot. Constraint (11c) guarantees that a segment can only be transcoded from the high version to the low version in the edge server. Constraints (11d, 11e) guarantee that the total allocated computing and bandwidth resources cannot exceed the computing capacity of the edge server and system bandwidth, respectively.

#### B. Proposed Algorithm

The formulated problem is mixed-integer nonlinear programming, which is hard to be directly solved. Considering that the user’s playback status satisfies Markov chain and the optimization objective is to maximize the long-term PQoE, the optimization problem can be modeled as a Markov decision process (MDP). Deep deterministic policy gradient (DDPG) algorithm can be applied to solve the MDP problem by using deep neural networks to approximate and update the policy function in an online manner [13] [14]. Therefor, the proposed algorithm is designed based on the DDPG-based algorithm.

1) **MDP:** MDP is a discrete-time stochastic control process, which consists of four elements, i.e., state, action, state transition probability, and reward. At each step (scheduling slot) $t$, the state $s_t$ can be transformed to $s_{t+1}$ by taking the action $a_t$. Correspondingly, the reward is $r_t$.

**State:** The state includes users’ rebuffer time, buffer occupancy, segment version, video quality, and segment bitrate: $s_t = \{\{R_{l,u}\} u \in \mathcal{U}, \{B_{l,u}\} u \in \mathcal{U}, \{l_{t,u}\} u \in \mathcal{U}, \{V_{l,u}\} u \in \mathcal{U}, \{s_{t,u}\} u \in \mathcal{U}\}$. (12)

**Action:** The action includes all optimization variables in Eq. (11), which is defined by

$$a_t = \{\{g_{t,u}\} u \in \mathcal{U}, \{o_{t,u}\} u \in \mathcal{U}, \{\omega_{t,u}\} u \in \mathcal{U}, \{\xi_{t,u}\} u \in \mathcal{U}\}. \quad \text{(13)}$$

**State Transition Probability:** At each step, the current segment version selection, and communication and computing resource allocation affect users’ next state. Therefore, the state transition probability between $s_t$ and $s_{t+1}$ is given by

$$\Pr(s_{t+1} = s_{t+1} | s_t = s_t, a_t = a_t) = \prod_{u \in \mathcal{U}} \Pr(R_{l_{t+1}} = s_{t+1} | R_{l_t} = s_t, g_{t,u} = g_t, o_{t,u} = o_t, \omega_{t,u} = \omega_t, \xi_{t,u} = \xi_t). \quad \text{(14)}$$

**Reward:** The reward at step $t$ is designed to maximize the overall PQoE, which is defined by

$$r_t(s_t, a_t) = \sum_{u \in \mathcal{U}} \tilde{Z}_{t,u}. \quad \text{(15)}$$
Algorithm 1: DCTRA

1. Initialize: primary and target network with parameters \( \theta_\pi, \theta_Q, \theta_{\pi'}, \theta_{Q'} \), and the replay memory \( M \).

2. For each episode do
   3. Initialize the video list for the edge server and the cloud server, users’ requests and buffer statuses.
   4. For each step \( t \in \{1, \ldots, t_{\text{max}}\} \) do
      5. Update each PQoE model based on parameters \( \lambda_{t,u}, \alpha_{t,u}, \beta_{t,u}, \gamma_{t,u} \) provided by UDTs;
      6. Discretize actions \( g_{t,u} \) and \( o_{t,u} \) based on the designed principle in Section III-B2;
      7. BS and the edge server execute action \( a_{t} \) based on \( a_{t} = \pi(s_{t}|\theta_{\pi}) + X_{t} \);
      8. Obtain \( r_{t} \) based on Eq. (15), update \( s_{t+1} \);
      9. Store \( \{s_{t}, a_{t}, r_{t}, s_{t+1}\} \) into \( M \), and sampling;
     10. Compute the target value \( y_{t} \) based on Eq. (18);
     11. Update primary critic network parameter \( \theta_Q \) based on Eq. (17);
     12. Update primary actor network parameter \( \theta_\pi \) based on Eq. (16);
     13. Update target network parameters via \( \theta_{\pi'} \leftarrow \tau_\pi \theta_\pi + (1 - \tau)\theta_{\pi'} \), \( \theta_{Q'} \leftarrow \tau_\theta \theta_{Q} + (1 - \tau)\theta_{Q'} \);
   end
end

Here, \( y_{t} \) is the target value that combines the current reward and the estimated Q value, i.e., \( Q' \), as follows
\[
y_{t} = r_{t} + \epsilon Q'(s_{t+1}, \pi'(s_{t+1}|\theta_{\pi'})|\theta_{Q'}),
\]
where \( \epsilon \) is the discounting factor.

Since the output values of the DDPG algorithm are continuous, we need to discretize a part of DDPG output values to obtain \( g_{t,u} \) and \( o_{t,u} \). To reduce the dimension of \( g_{t,u} \) and \( o_{t,u} \) caused by multiple segment versions, we utilize \( g_{t,u} \) and \( o_{t,u} \) to replace \( g_{t,u} \) and \( o_{t,u} \). Specifically, we first map the output value of the Tanh function from \([-1, 1]\) to \([0, 1]\). Then, we split the range of \( g_{t,u} \) into five parts, i.e., \([0, 0.2], [0.2, 0.4], [0.4, 0.6], [0.6, 0.8], [0.8, 1]\), which indicate no segment transmission, and segment transmission for version 1, 2, 3, 4 for user \( u \) at step \( t \), respectively. Similarly, the range of \( o_{t,u} \) is split into two parts, i.e., \([0, 0.5], [0.5, 1]\), which indicate no segment transcoding, and segment transcoding for user \( u \) at step \( t \), respectively.

IV. PERFORMANCE EVALUATION

In this section, simulations are conducted to evaluate the performance of the proposed DCTRA algorithm. We adopt the real-world dataset “Waterloo Streaming Quality-of-Experience Database-III”\(^3\), that includes users’ rebuffer time, video quality and quality variation, to evaluate our algorithm. The dataset contains 450 videos, and the length of each video is 10 seconds. The user’s leave probability in [15] is used to obtain the engagement time for calculating the PQoE reference value.

\(^3\)https://ieeexplore.ieee.org/abstract/document/9221336

TABLE I: Simulation Parameters

| Parameter                  | Value                  | Parameter                  | Value |
|----------------------------|------------------------|----------------------------|-------|
| BS radius                  | 600 m                  | \( U \)                    | 12    |
| Hidden layer shape         | (512, 256, 128)        | \( c \)                    | \( 10^3 \) cycles/s |
| Critic learning rate       | \( 10^{-4} \)          | \( W \)                    | 200 MHz |
| Actor learning rate        | \( 10^{-6} \)          | \( r_c \)                  | 200 Mbps |
| Mini-batch size            | 128                    | \( P \)                    | 25 dBm |
| Replay memory              | \( 6 \times 10^3 \)     | \( \mu \)                  | 10 cycles/bit |
| Scheduling slot length     | 100 ms                 |                            |       |
| Number of episodes         | 1500                   | \( t_{\text{max}} \)       | 100   |

Fig. 2: PQoE model parameters extracted by UDT.

To obtain PQoE model parameters, we employ the lsqcurvefit function\(^4\) in Matlab to fit the function curve. The main simulation parameters are presented in Table I.

We compare the proposed DCTRA algorithm with the following benchmark schemes.

- **Round-Robin (RR):** In each scheduling slot, the communication and computing resources are equally allocated to 3 users randomly selected from 12 users. The segment version is randomly selected from all versions.

- **Proportional Fair (PF):** The communication and computing resources are allocated based on the scheduling priority considering users’ current channel states and buffer occupancy. The segment version is selected based on the previous average version.

- **Joint Resource Allocatin (JRA)**[9]: The communication and computing resources are sequentially allocated to the user who can achieve the highest PQoE enhancement. The segment version selection is determined by the branch and bound method.

- **Communication and Transcoding Resource Allocation (CTA):** The proposed DCTRA algorithm does not use DT-assisted PQoE models, but uses static PQoE models.

Figure 2 shows dynamic PQoE model parameters of Users 1, 4, 7, and 11, whose relative memory lengths are 12.80 seconds, 11.93 seconds, 12.02 seconds, and 11.68 seconds.

\(^4\)https://www.mathworks.com/help/optim/ug/lsqcurvefit.html.
respectively. It can be observed that User 1 has the highest sensitivity degree of all QoE factors, while User 11 owns the lowest one. In addition, users’ PQoE model parameters change with varying video contents and playback statuses, indicating that DTs can well characterize users’ dynamic PQoE features.

Figure 3(a) shows the convergence property of the proposed DCTRA algorithm, which conducts three trials of training to draw the corresponding envelope curve and the mean curve. It can be observed that our algorithm can converge after near 1,100 episodes. In Fig. 3(b), we select 6 users from 12 users to show the individual user’s normalized PQoE among different algorithms. It can be observed that the proposed DCTRA algorithm can always achieve the highest PQoE for each user compared with other algorithms. This is because accurate user-specific PQoE models can be provided to the network controller in real time to perform appropriate resource management. Fig. 3(c) shows the proposed algorithm can well adapt to different user scales and always achieve the highest PQoE. The gap between the JRAT algorithm and the CTRA algorithm decreases gradually, because these two algorithms cannot find better resource management decisions based on static PQoE models.

V. CONCLUSION

In this paper, we have investigated a PQoE enhancement problem for adaptive video streaming. We have proposed a DT-assisted PQoE model construction method to obtain accurate user-specific PQoE models, and then developed a DRL-based resource management algorithm to enhance the overall long-term PQoE. The proposed scheme can extract users’ features through DT, and leverage them to enhance users’ video watching experience. In the future, we will investigate an efficient data collection and model update scheme for UDT to reduce communication and computing overhead.

ACKNOWLEDGMENT

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada, and the Major Key Project of the Peng Cheng Laboratory (PCL2021A09-B2).

REFERENCES

[1] A. A. Barakabitze, N. Barman, A. Ahmad, S. Zadootaghaji, L. Sun, M. G. Martini, and L. Atzori, “QoE management of multimedia streaming services in future networks: A tutorial and survey,” IEEE Commun. Surveys Tuts., vol. 22, no. 1, pp. 526–565, 2019.
[2] C. Liu, I. Bouazizi, and M. Gabbouj, “Rate adaptation for adaptive HTTP streaming,” in Proc. ACM MMSys, 2011, San Jose CA, USA, pp. 169–174.
[3] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, “A control-theoretic approach for dynamic adaptive video streaming over HTTP,” in Proc. ACM SIGCOMM, 2015, New York, NY, USA, pp. 325–338.
[4] Y. Wang, P. Li, L. Jiao, Z. Su, N. Cheng, X. Shen, and P. Zhang, “A data-driven architecture for personalized QoE management in 5G wireless networks,” IEEE Wireless Commun., vol. 24, no. 1, pp. 102–110, 2017.
[5] Y. Gao, X. Wei, and L. Zhou, “Personalized QoE improvement for networking video service,” IEEE J. Sel. Areas Commun., vol. 38, no. 10, pp. 2311–2323, 2020.
[6] F. Wang, C. Zhang, F. Wang, J. Liu, Y. Zhu, H. Pang, and L. Sun, “DeepCast: Towards personalized QoE for edge-assisted crowdcast with deep reinforcement learning,” IEEE Trans. Netw., vol. 28, no. 3, pp. 1255–1268, 2020.
[7] X. Shen, J. Gao, W. Wu, M. Li, C. Zhou, and W. Zhang, “Holistic network virtualization and pervasive network intelligence for 6G,” IEEE Commun. Surveys Tuts., vol. 24, no. 1, pp. 1–30, 2022.
[8] T. H. Luan, R. Liu, L. Gao, R. Li, and H. Zhou, “The paradigm of digital twin communications,” arxiv preprint arXiv:2105.07182, 2021.
[9] X. Huang, L. He, L. Wang, and F. Li, “Towards 5G: Joint optimization of video segment caching, transcoding and resource allocation for adaptive video streaming in a multi-access edge computing network,” IEEE Trans. Veh. Technol., vol. 70, no. 10, pp. 10 909–10 924, 2021.
[10] Z. Duanmu, K. Zheng, K. Ma, A. Rehman, and Z. Wang, “A quality-of-experience index for streaming video,” IEEE J. Sel. Topics Signal Process., vol. 11, no. 1, pp. 154–166, 2017.
[11] F. Dobrian, V. Sekar, A. K. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang, “Understanding the impact of video quality on user engagement,” in Proc. ACM SIGCOMM, 2011, Toronto, ON, CA, pp. 362–373.
[12] Y. Chen, K. Wu, and Q. Zhang, “From QoE to QoS: A tutorial on video quality assessment,” IEEE Commun. Surveys Tuts., vol. 17, no. 2, pp. 1126–1165, 2015.
[13] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, “Continuous control with deep reinforcement learning,” in Proc. ICLR, 2016, San Juan, Puerto Rico.
[14] C. Zhou, W. Wu, H. He, P. Yang, F. Lyu, N. Cheng, and X. Shen, “Deep reinforcement learning for delay-oriented IoT task scheduling in SAGIN,” IEEE Trans. Wireless Commun., vol. 20, no. 2, pp. 911–925, 2020.
[15] H. Nam, K. H. Kim, and H. Schulzrinne, “QoE matters more than QoS: Why people stop watching cat videos,” in Proc. IEEE INFOCOM, 2016, San Francisco, CA, USA.