CTC: Content-Aware Tailoring of Adaptive Video Streaming using Multi-Head Critic Network

Wangyu Choi and Jongwon Yoon
Department of Computer Science and Engineering, Hanyang University
Ansan, Republic of Korea
{wangyu, jongwon}@hanyang.ac.kr

***Abstract*—In this paper, we aim to enhance video streaming quality by taking into account a simple observation: users tend to focus on specific areas within a video. For instance, low-quality or stall events during scoring moments in sports videos can lead to user frustration. However, most existing video streaming solutions treat all scenes equally. In our work, we introduce CTC, an ABR algorithm that adjusts its policy based on scenes. To achieve this, we first model dynamic QoE based on scenes and then use reinforcement learning to adapt the policy in real-time. As a result, CTC significantly improves QoE by adjusting its policy according to content compared to existing work.**

***Index Terms*—Adaptive bitrate, video streaming, reinforcement learning, multi-head critic network**

I. Introduction

Video streaming applications such as YouTube and Zoom have become an essential part of our daily routine, and users expect high-quality streaming services that deliver a smooth and uninterrupted viewing experience. To achieve this, many providers adopt an adaptive bitrate (ABR) algorithm to optimize their services. The ABR algorithm adjusts the bitrate of videos based on the available bandwidth to deliver high-quality video streaming services to users.

Despite the growing popularity of ABR algorithms [1]–[3], they often overlook the differences in video scenes, resulting in variations in quality of experience (QoE). For example, when watching sports videos, we tend to pay more attention to scoring moments and feel more discomfort during low quality or stall events [13]. Conversely, credit scenes in movies have a relatively minor impact on QoE. While QoE in video streaming can vary depending on the content or scene, existing works tend to treat it uniformly.

To address this issue, we propose CTC (configure-to-content), an ABR algorithm, that adjusts the policy according to the video scene. We start by gathering user preferences or areas of interest for each video segment from large-scale video streaming service to model the time-varying user's QoE. Our QoE model gives varying degrees of importance to each segment based on the significance of the scene. Then, we employ reinforcement learning to adjust the policy in real-time, aiming to maximize the QoE model. Furthermore, we propose a multi-head critic network for estimating sub-rewards to enable fast adaptation of time-varying and complex QoE models.

To evaluate the performance of CTC, we conduct several experiments using an publicly available network dataset collected from real-word networks and YouTube videos. By adjusting the algorithm based on the video content, CTC achieves superior performance compared to existing solutions in terms of the time-varying QoE metric. Moreover, we demonstrate qualitatively that CTC adapts in real-time to unfold different policies in different scenes.

II. Related Work

Traditional ABR Algorithms. To improve the video streaming quality, many ABR algorithms have been proposed. FESTIVE [4] predicts future bandwidth using the harmonic mean of throughput over the past 20 segments. BBA [5] determines bitrate based on playback buffer occupancy. MPC [1] selects the bitrate using a model predictive control algorithm. BOLA [6] uses Lyapunov optimization to maximize bitrate utility and the fraction of time spent not rebuffering. Recent, machine learning-based approaches including reinforcement and supervised learning have been proposed. Pensieve [2] uses reinforcement learning to enhance QoE by directly update the neural newtork in a trial-and-error manner. To accurately predict the future bandwidth, CS2P [7] and Fugu [3] adopt supervised learning. However, all of the traditional ABR schemes mentioned earlier tend to overlook time-varying QoE.

Content-Aware QoE Modeling. Some other work [8], [9] has suggested modeling user perception systematically through objective quality metrics. This approach aims to investigate the effects of different types of low-quality events on QoE [10], and to develop crowdsourcing platforms using commercial platforms like Amazon MTurk [11] and microWorkers [12]. Though these algorithms aim to accurately model QoE through crowdsourcing, they did not consider it in conjunction with the ABR algorithm.

Zhuang et al. [13] proposed SENSEI, an ABR algorithm that improves video streaming quality by considering both QoE sensitivity and ABR policy. However, this algorithm requires pre-modeling the video for QoE, which can be expensive.

III. System Design

The goal of CTC is to maximize user QoE by adjusting the policy of the adaptation algorithm in real-time based on the importance of the scene, unlike existing algorithms that treat all video segments equally. To achieve this, we first obtain the number of replay times for each scene from a large-scale video streaming service. We introduce a content-aware QoE model by considering the number of plays as the scene's importance. Then, we train a neural network-based CTC using reinforcement learning in a trial-and-error manner to maximize QoE with time-varying scene importance. As shown in Fig. 1, we adopt an actor-critic architecture [14] for CTC's neural network. The critic network is updated to accurately predict future QoE, while the actor network is updated to determine the bitrate that is expected to achieve higher QoE with the help of the critic network. Furthermore, we propose a novel multi-head critic reinforcement learning technique that involves decomposing the summed-up reward function. This approach provides accurate predictions from the critic network and intuitive insights for the actor network.

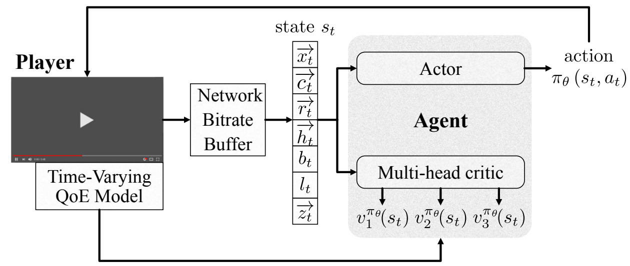

Fig. 1. System overview of the proposed model, CTC. It consists of a video player and an agent. The agent is updated to maximize time-varying QoE by interacting with the video player.

A. Time-Varying QoE Model

To ensure high QoE for users, an accurate QoE model must be established as the foundation before designing the ABR algorithm. While the most accurate way to achieve this is to directly obtain the importance of video scenes from users, it is practically impossible since users cannot access future scenes. Instead, we adopt a statistical approach obtained from large-scale video streaming services. Specifically, we extract the time-based replay count from YouTube and treat it as the importance of the corresponding scene. For instance, we set the importance to be low for scenes that most people skip and high for scenes with high replay counts, adjusting the weight of QoE. The QoE for a video with N segments can be defined as follows:

$$QoE = \sum_{n=1}^{N} \rho_n q(R_n) - \mu \sum_{n=1}^{N} T_n - \sum_{n=1}^{N-1} |q(R_{n+1}) - q(R_n)| \tag{1}$$

where, R_n denotes the bitrate for the n-th segment, ρ_n is the importance factor for the n-th segment, and the function $q(\cdot)$ maps bitrate to the user's perceived quality. T_n represents the rebuffering time for the n-th segment, and μ is the weight for the rebuffering penalty. The last term represents the penalty for quality changes.

Calculating scene importance ρ**.** We first obtain the number of replay times for each scene from a large-scale video streaming service, YouTube. YouTube provides a more detailed number of playback per section instead of a simple number of playback for the video. In general, it can be assumed that it represents the importance of the scene because a person sees more sections of interest. We obtain the number of plays per section in a given video, and then calculate the importance of having a value between 0 and 1 using the minimum and maximum values. Simply put, the most interested section in the video is 1, and the least interested section is 0.

B. Multi-Head Critic Policy Gradient

We utilize the powerful policy gradient algorithm, proximal policy optimization (PPO) [15], to train CTC's actor-critic neural network. Despite its simplicity and effectiveness, the value function (i.e., critic network) is hindered by the time-varying QoE model and the sum of various sub-rewards, ultimately limiting the performance of the actor network. To address this issue, we introduce a multi-critic head policy gradient algorithm, as depicted in Fig. 1, which breaks down complex rewards into sub-rewards to enhance the performance of the value function and thus make it more resilient to time-varying QoE.

The training objective of the actor network is formulated as follows:

$$\mathcal{L}(\theta) = \hat{\mathbb{E}}\left[\min(r_t(\theta)\hat{A}_t^K, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t^K)\right] \tag{2}$$

where, $r_t(\theta)$ is the ratio between the probability of action a_t used in the experience (old) and the new policy $\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$. ϵ is one of the hyper-parameters, usually set to 0.1 or 0.2, that has the effect of discarding samples that cause excessively large updates. $\hat{A}_t^K$ is the sum of the all sub-advantage function (i.e., generalized advantage estimation (GAE) [16]), which is defined as follows:

$$\hat{A}_t^K = \sum_{k=1}^{K} \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}^{V^{(k)}} \tag{3}$$

where, K is the number of sub-rewards. $\delta_{t+l} = r_{t+l} + \gamma V(s_{t+l+1}) - V(s_{t+l})$ represents the temporal-difference (TD) error at time step $t+l$. This error is the difference between the sum of rewards from time step $t+l$ onward and the estimated state-value function $V(s_{t+l})$ at time step $t+l$. The discount factor γ controls the importance of future rewards relative to immediate rewards. Meanwhile, the eligibility trace decay parameter λ determines how much past TD errors are attributed to the current estimate of the advantage function.

C. Network Architecture

CTC's neural network is composed of four main components: the state encoding layer, hidden layers, actor network, and critic network. The state encoding layer encodes the input state s_t and is made up of several 1D-CNN and fully connected layers. The hidden layer branches into the actor and critic networks, each containing 128 neurons and utilizing two fully-connected layers. The actor network outputs action probabilities after several fully connected layers, while the critic network estimates the accumulated future reward for

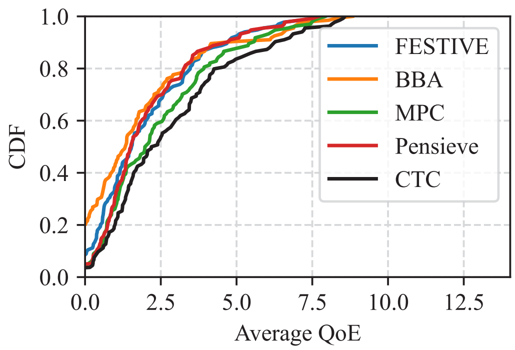

Fig. 2. Overall performance comparison between CTC and baseline algorithms.

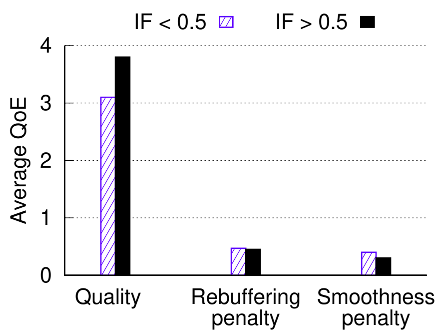

Fig. 3. Differences in sub-QoE depending on the areas of different importance factors. IF means the importance factor of the segment.

sub-rewards. The majority of parameters in each head of the critic network are shared, with only the output layer being independent. The input state consists of seven elements: past throughput $\overrightarrow{x_t}$, segment size $\overrightarrow{c_t}$, available bitrate of video $\overrightarrow{r_t}$, history of selected bitrate $\overrightarrow{h_t}$, buffer occupancy b_t, number of segment left l_t, and future scene importance $\overrightarrow{z_t}$.

IV. Evaluation

A. Experiment Setup

Network and video dataset. We used a UCC network trace dataset [17] from a real-world LTE network to conduct our study. The dataset was obtained during video streaming and contains a variety of mobility patterns, including static, pedestrian, car, tram, and train. The dataset has an average bandwidth of around 15Mbps. We partitioned the dataset into a 9:1 ratio for the purpose of training and testing. We downloaded the 100 videos with the "most replayed time" from YouTube. Next, we encode the videos using a range of bitrates from 0.2 to 14.9 Mbps with slight variations. Next, we randomly select a segment length between 2 and 15 seconds, and split the videos accordingly. Finally, each segment of videos is assigned a normalized number of replay counts.

B. Results

For the performance evaluation of CTC, we consider two aspects: (i) comparing the overall performance to existing baseline algorithms, and (ii) analyzing the behavior of the algorithm in relation to the time-varying QoE model.

Overall performance. We compare the performance of CTC with baseline algorithms on network traces extracted from the Puffer website [3]. The baseline algorithms considered are rate-based FESTIVE, buffer-based BBA, metric-based MPC, and traditional reinforcement learning-based Pensieve. QoE is calculated using Equation 1, and Fig. 2 show the CDF of the average QoE per segment. We found that CTC is the most effective algorithm with the time-varying QoE model. Note that, we do not disclose based on experimental results, but argue that CTC is similar to the performance of traditional RL-based algorithms such as Pensieve on traditional QoE metrics that are not time-varying as our framework inherits RL-based scheme's principle as it is.

Behavior analysis. To conduct a behavioral analysis of CTC, we differentiate video segments based on importance factors that exceed or do not reach 0.5, and calculate the average QoE for each segment. For detailed analysis, we classify QoE into three categories: quality, rebuffering penalty, and smoothness, as depicted in Fig. 3. CTC adopts a conservative approach for low-importance scenes, while prioritizing high-quality videos with importance factors greater than 0.5. It is notable that the rebuffering penalty and smoothness are either similar or better. This suggests that CTC adapts its policy effectively based on the importance of video scenes.

V. Conclusion

We developed a content-aware ABR algorithm, CTC, to maximize user QoE by adjusting adaptation policy in real-time based on scene importance. We obtained scene importance from a large-scale video streaming service by tracking the number of plays for each scene over time. Using reinforcement learning, we trained a neural network-based CTC to maximize QoE with time-varying scene importance. We also proposed a novel multi-head critic reinforcement learning technique that decomposes the reward function. Our approach outperforms existing schemes such as MPC and Pensieve in terms of QoE. We believe that our approach can be used in various video streaming services to provide better QoE for users.

Acknowledgment

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the Grand Information Technology Research Center support program(IITP-2023-2020-0-01741) supervised by the IITP(Institute for Information & communications Technology Planning & Evaluation)

References

[1] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP," in *Proc. 2015 ACM Conf. on Special Interest Group on Data Commun.*, pp. 325–338.

[2] H. Mao, R. Netravali, and M. Alizadeh, "Neural Adaptive Video Streaming with Pensieve," in *Proc. Conf. ACM Special Interest Group on Data Commun.*, 2017, pp. 197–210.

[3] F. Y. Yan, H. Ayers, C. Zhu, S. Fouladi, J. Hong, K. Zhang, P. Levis, and K. Winstein, "Learning in situ: a randomized experiment in video streaming," in *Proc. 17th USENIX Symp. Netw. Syst. Des. Implementation*, 2020.

[4] J. Jiang, V. Sekar, and H. Zhang, "Improving Fairness, Efficiency, and Stability in HTTP-Based Adaptive Video Streaming with FESTIVE," in *Proc. 8th int. Conf. emerg. Netw. EXp. Techn.*, 2012.

[5] T. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A Buffer-Based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service," in *Proc. 2014 ACM Conf. Special Interest Group on Data Commun.*, pp. 187–198.

[6] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "BOLA: Near-Optimal Bitrate Adaptation for Online Videos," *The 35th Annu. IEEE Int. Conf. Comput. Commun.*, 2016, pp. 1–9.

[7] Y. Sun, X. Yin, J. Jiang, V. Sekar, F. Lin, N. Wang, T. Liu, and B. Sinopoli, "CS2P: Improving Video Bitrate Selection and Adaptation with Data-Driven Throughput Prediction," in *Proc. 2016 ACM Conf. Special Interest Group on Data Commun.*, 2016, pp 272–285.

[8] C. Wu, K. Chen, Y. Chang, and C. Lei, "Crowdsourcing Multimedia QoE Evaluation: A Trusted Framework," *IEEE Trans. Multimedia*, vol. 15, no. 5, pp. 1121–1137, 2013.

[9] B. Rainer, M. Waltl, and C. Timmerer, "A Web Based Subjective Evaluation Platform," in *Proc. 5th Int. Workshop on Qual. Multimedia Experience (QoMEX)*, 2013.

[10] S. Egger, B. Gardlo, M. Seufert, and R. Schatz, "The Impact of Adaptation Strategies on Perceived Quality of HTTP Adaptive Streaming," in *Proc. 2014 Workshop on Des. Qual. Deployment of Adaptive Video Streaming*, 2014, pp. 31–36.

[11] "Amazon Mechanical Turk." https://www.mturk.com (accessed Mar. 20, 2023).

[12] "microWorkers." https://www.microworkers.com/ (accessed Mar. 20, 2023).

[13] X. Zhang, Y. Ou, S. Sen, and J. Jiang, "SENSEI: Aligning Video Streaming Quality with Dynamic User Sensitivity," in *Proc. 18th USENIX Symp. Netw. Syst. Des. Implementation*, 2021, pp. 303–320.

[14] V. Konda and J. Tsitsiklis, "Actor-Critic Algorithms," *Adv. Neural Inf. Process. Syst.*, vol. 12, pp. 1057–1063, 1999.

[15] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," 2017, *arXiv:1707.06347*.

[16] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-Dimensional Continuous Control Using Generalized Advantage Estimation," 2018, *arXiv:1506.02438*.

[17] D. Raca, J. J. Quinlan, A. H. Zahran, and C. J. Sreenan, "Beyond Throughput: A 4G LTE Dataset with Channel and Context Metrics," in *Proc. 9th ACM Multimedia Syst. Conf.*, 2018, pp. 460–465.