DASH Live Video Streaming Control Using Actor-Critic Reinforcement Learning Method

Bo Wei[1(✉)], Hang Song[2], Quang Ngoc Nguyen[1], and Jiro Katto[1]

[1] Department of Computer Science and Communication Engineering, Waseda University, Tokyo, Japan
weibo@aoni.waseda.jp
[2] School of Engineering, The University of Tokyo, Tokyo, Japan

Abstract. With the COVID19 pandemic, video streaming traffic is increasing rapidly. Especially, the live streaming traffic accounts for large amount due to the fact that many events have been switched to the online forms. Therefore, the demand to ensure a high-quality streaming experience is increasing urgently. Since the network condition is expected to fluctuate dynamically, the video streaming needs to be controlled adaptively according to the network condition to provide high quality of experience (QoE) for users. In this paper, a method was proposed to control the live video streaming using the actor-critic reinforcement learning (RL) technique. In this method, the historical video streaming logs such as throughput, buffer size, rebuffering time, latency are taken consideration as the states of RL, then the model is established to map the states to an action such as bitrate decision. In this study, the live streaming simulation is utilized to evaluate the method since the model needs training and the simulation can generate data much faster than real experiment. Experiments were conducted to evaluate the proposed method. Results demonstrate that the total QoE in Bus and Car scenarios show the best performance. The QoE of Tram case shows the lowest due to the low bandwidth.

Keywords: QoE · Dash · Live streaming · Actor-critic

1 Introduction

Recently, the video streaming traffic has been increasing dramatically which account for a large amount of the total internet traffic. Especially, during the pandemic period, the live streaming becomes a major part since educations, works, meetings have been shifted to online forms. How to ensure a high-quality video streaming is essential. Since the network conditions of different users are expected to be changing dynamically, keeping the same video streaming quality may cause impairment to the quality of experience (QoE) of users. If the streaming is kept at a low level, the network may not be fully exploited. While keeping the streaming at a high level, the video may stall due to bad network conditions. Therefore, a common way to control the video streaming is to adaptively choose the streaming quality according to the network condition. This dynamic

C. T. Calafate et al. (Eds.): MONAMI 2021, LNICST 418, pp. 17–24, 2022.
https://doi.org/10.1007/978-3-030-94763-7_2

strategy provides a solution to video streaming in fluctuating networks especially in mobile networks.

Over the past years, adaptive video streaming has been widely studied [1, 2]. And dynamic adaptive streaming over HTTP (DASH) [3] has become the standard for dynamic streaming. In DASH, the videos are encoded into different qualities and the users can request different quality level according to their own network conditions. In DASH mechanism, the key part is the adaptive bitrate (ABR) control technique. The ABR will decide the bitrate level to be requested according to the network conditions. Many video streaming providers, such as YouTube and Netflix have utilized the ABR technology. By using ABR methods, the proper bitrate of video chunk can be selected to realize high QoE video transmission.

ABR methods have been studied for on-demand videos [4], where the contents are created completely before the access from users. While for live streaming, the videos are generated in real time and there are some different factors need to be taken consideration such as the latency [5, 6]. In the live streaming scenarios, the factors which influence the QoE bitrate, quality switch, rebuffering time, latency and etc. In is paper, the ABR method for DASH live streaming is proposed which utilizing actor-critic (AC) reinforcement learning technique. The proposed method is a model which relates the network condition status to the bitrate selection and other decisions. To train the model, data are necessary. However, if the data generated by real experiment, it will be extremely slow since the data will only be generated together with the real live streaming. Here, the live streaming simulator provided in MMGC2019 is employed which can experience the streaming and record the status logs much faster than real experiment [7]. In the experiment, the QoE metric provided in MMGC2019 is also utilized to evaluate the performance of the proposed method.

Experiments are conducted to evaluate the proposal, the QoE of different methods are evaluated and compared in different scenarios. Experiment results indicate that the QoE in Bus and Car scenarios show the best performance. The QoE of Tram case shows the lowest due to the low bandwidth.

2 Related Work

The final goal of ABR method is to control video transmission according to network conditions to ensure a high-quality streaming experience. The selected bitrate is adjusted to maximize the QoE of the video service, by increasing the video quality, and decreasing the rebuffering event and delay.

The existing methods can be classified into three categories. Throughput-based strategy utilize the predicted throughput for the bitrate selection [8–13]. These methods predict future throughput with techniques such as harmonic mean or moving average, and then choose the maximum bitrate which is lower than the predicted value. Buffer-based strategy utilize buffer information to control the video streaming [14, 15]. Buffer-based method only takes the buffer occupancy into account and selects the lower bitrate when the buffer status is low and vice versa. Hybrid strategy takes different information such as throughput prediction, buffer occupancy, rebuffering time and so on into account when select the action to reach the best QoE. Leading hybrid methods include the model

predictive control (MPC) approach [16] and machine learning approach Pensieve [17]. In MPC, a principled control-theoretical model is developed and the MPC algorithm is proposed to optimally combine the throughput prediction and buffer occupancy for decision of the future video chunks downloading. In Pensieve, a neural-network model is established and the reinforcement learning model is used for training and generate the selection algorithm of the bitrate based on former collected observations of bandwidth, buffer occupancy and bitrate.

Traditionally, the ABR methods are designed from the view of single users. While in the networks, the global quality of different users is also important which means that whether different users are getting fair video streaming experience. In the former studies, it was found that ABR methods works well for single user consideration may not be effective in multi-user scenarios [18]. Several methods have been proposed to ensure the fair, stable, and efficient video streaming in multiuser networks [19–22].

While for the live streaming, it has some differences compared with on-demand counterpart. The biggest difference is that the latency is an essential factor in QoE. If the latency is too large, the live streaming will be meaningless since the real-time experience is the core. Therefore, when the latency is too large, there will be a skip mechanism to give up some contents to catch up with the newest contents. Since the real experiment will take a long time to develop and evaluate the ABR methods, simulation is considered to be the suitable methodology. MMGC19 has provided a live streaming simulator. In this study, the development and evaluation will be based on the simulator.

3 Simulation Mechanism

The general framework of the live streaming is shown in Fig. 1. The live event is recorded by the video camera and the raw video is transmitted to the transcoding server to generate the video contents at different levels. Then the contents are transmitted to CDN servers to be accessed by the user. On the client side, the streaming application has an ABR engine to control the video streaming, it will generate decisions according to the current network condition.

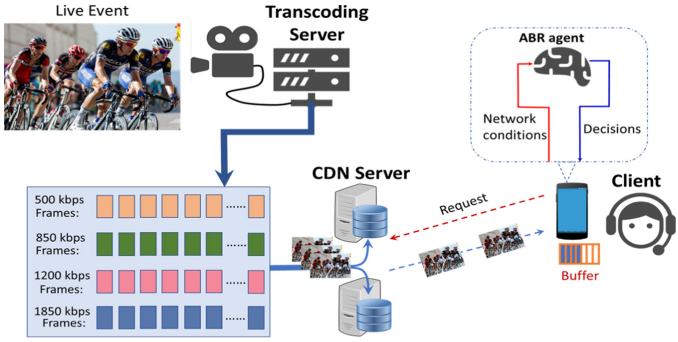

Fig. 1. The framework for the live streaming.

To simulate the live streaming mechanism, the simulator behaves in a frame-level way. In the simulator, the video information and the network trace information are necessary to run the simulation. During the simulation, every frame will be "downloaded" and the time consumption will be calculated numerically using the network bandwidth and the size of the frame. Meanwhile, other logs such as client buffer size, rebuffering time, newest content, latency and others are calculated and updated. The QoE for the frame will be calculated and stored. At the end of the simulation, the QoE will be summed up to assess the streaming. In the simulator, there is a skip-frame mechanism by which the client will give up the downloading of some frames to catch up with the live event, in case of a large latency.

The video content in the simulator is described by frame and the frame rate is 25 fps. It is divided into 2-s segments and each segment is encoded into 4 bitrate level for being requested. The available bitrates are 500 kbps, 850 kbps, 1200 kbps, and 1850 kbps. During the simulation, the bitrate selection will be triggered at the beginning of each segment. There are three parameters to be input, the bitrate, the target buffer and the latency limit. The bitrate is the chosen video quality level. The target buffer is the threshold to resume the streaming after rebuffering. The latency limit is the threshold to trigger skip frame mechanism.

4 Proposed Method

In this paper, reinforcement learning technique is utilized to develop a policy model to control the video streaming as shown in Fig. 2. When the live streaming is on-going, the client will monitor the network conditions by a set of parameters, which are defined as states in the RL model. The states include the throughput, buffer level, latency, skip time, rebuffering etc. During the streaming, the control model will generate an action based on current states. Then, the bitrate, target buffer, latency limit included in the action, will be applied to the next segment downloading. At the end, the total QoE will be utilized. Based on the <state, action, reward> data sets, the parameters of the policy model will be trained using actor-critic policy gradient method. The RL model will be optimized repeatedly to improve the QoE.

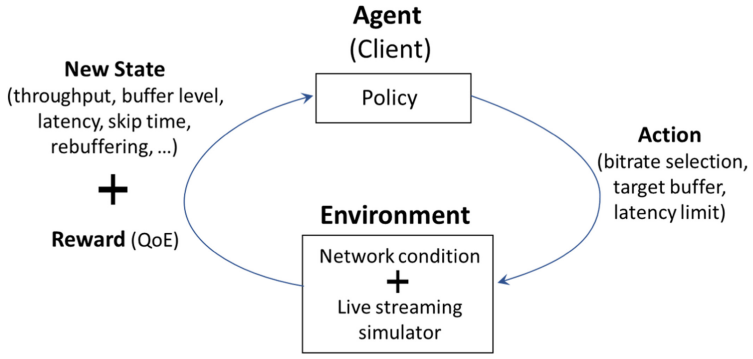

Fig. 2. Reinforcement learning framework for ABR control.

For the policy model, the three actions are independently built. The bitrate selection is modeled as discrete actions where each action represents a bitrate level. Here, 4 bitrate level are set. The target buffer is also modeled as discrete actions which has two choice, 0.5 s or 1 s. The latency limit is modeled as continuous actions. Here, the latency limit can be selected within the range from 0 to 8. The state parameters are connected to the policy layer using linear combination as the following equation:

$$\varphi = \overrightarrow{w} \bullet \overrightarrow{S} \tag{1}$$

where $\overrightarrow{S}$ are the state values of the network condition. $\overrightarrow{w}$ are the coefficient for different state parameter. φ is the output of the linear combination. In the ABR policy, the hyperbolic tangent function is utilized to output a value within -1 to 1 as follows:

$$y_{abr} = \frac{e^{2\varphi_{abr}} - 1}{e^{2\varphi_{abr}} + 1} \tag{2}$$

Then, a piecewise function is utilized to decide the bitrate selection as:

$$a_{abr} = \begin{cases} 1, \ y_{abr} < -0.5 \\ 2, \ -0.5 < y_{abr} < 0 \\ 3, \ 0 < y_{abr} < 0.5 \\ 4, \ y_{abr} > 0.5 \end{cases} \tag{3}$$

Similar to ABR, in the target buffer policy, the logistic function is used to output the value within 0 to 1. If the value is lower than 0.5, the target buffer will be selected as 0.5 s and vice versa. In latency limit polity, the logistic function is utilized to output a value and multiplied by 8 to decide the latency limit as follows:

$$a_{latency} = 8 * \frac{1}{1 + e^{-\varphi_{latency}}} \tag{4}$$

5 Evaluation

5.1 Metrics

In order to evaluate the performance of the control method, an evaluation standard is necessary. In MMGC2019, a QoE metric is provided which includes different factors. This metric is also adopted as follows:

$$qoe[n] = 0.04 * q[n] - 1.85 * T_{rebuf} - w_1 * T_{latency} - 0.02 * |q[n] - q[n-1]| - 0.5 * T_{skip} \tag{5}$$

This equation stands for the QoE reward of the nth frame. It is calculated every time when one frame is downloaded. At the end of the streaming, all the $qoe[n]$ are summed up to generate a total QoE for the streaming session. In Eq. (5), $q[n]$ is the bitrate, T_{rebuf} is the rebuffering time, T_{delay} is the latency, T_{skip} is the skipped time. w_1 is the penalty coefficient of latency. It has two values. When the latency is larger than 1 s, it is set as 0.01. When it is less than 1 s, it is set as 0.005.

5.2 Experiment Results

The live streaming simulation was carried out using 6 different network traces. The traces include the bandwidth logs at every second. These traces are obtained from an open dataset by Mobile High-Speed Downlink Packet Access (HSDPA) [23]. The 6 network bandwidth logs were measured in the scenarios of Bus, Metro, Tram, Ferry, Car and Train. Three of the traces are shown in Fig. 3 which demonstrate the dynamics of the network conditions. In bus case, it can be observed that the bandwidth is almost under 2 Mbps before 200 s. Between 130 s and 200 s, the bandwidth is low. And after 200 s, the bandwidth increases to about 5 Mbps. In car case, the bandwidth is about 2 Mbps. In tram case, the bandwidth is relatively low compared with the other cases which may only support video streaming with a lower quality.

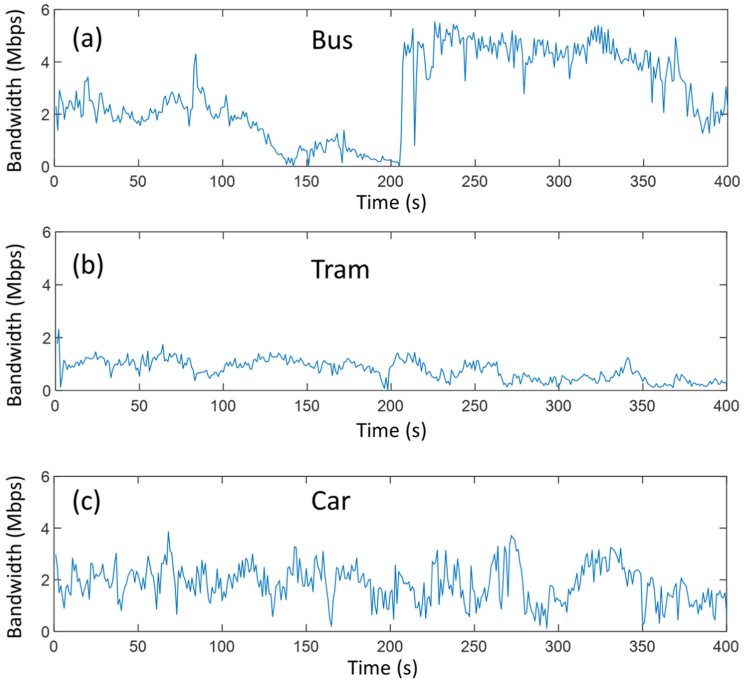

Fig. 3. The bandwidth traces from different conditions.

Figure 4 shows the simulation results in all the scenarios, from which we can obtain the total QoE performance of the live streaming in different conditions. It is found that the QoE in tram case is the lowest. This is because the bandwidth in tram case is the smallest compared with other conditions, thus the video quality selected in tram case is always small. In this case, the total QoE in tram case is the smallest. The total QoE in bus case shows the highest value because the bandwidth of which is the highest after about 200 s, and the bandwidth before 130 s is not low. The total QoE of Car case is also high, as the bandwidth is always not low and fluctuate between the value from 0−4 Mbps. From

the experiment results, we can conclude that the bandwidth is fluctuating differently in various communication conditions. The ABR method control scheme plays differently in different network conditions.

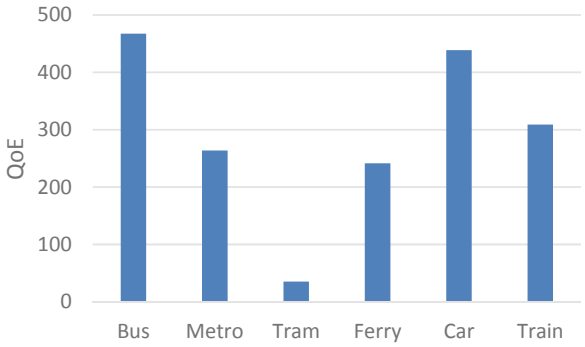

Fig. 4. The experiment results of different scenarios.

6 Conclusion and Future Work

In this paper, ABR method was proposed to control the video streaming using the actor-critic reinforcement learning (RL) technique. In this method, the historical streaming logs such as throughput, buffer size, rebuffering time, latency are taken into consideration. And the live streaming simulation is utilized to evaluate the proposal since the model needs training and the simulation can generate data much faster than real experiment. Experiment results indicate that the QoE in Bus and Car scenarios show the best performance. The QoE of Tram case shows the lowest due to the low bandwidth.

Acknowledgement. This research is supported by JSPS KAKENHI Grant Number 20K14740 and Waseda University Grant for Special Research Projects (Project Number: 2021C-132, 2021E-013).

References

1. Sani, Y., Mauthe, A., Edwards, C.: Adaptive bitrate selection: a survey. IEEE Commun. Surv. Tutor. **19**(4), 2985–3014 (2017)
2. Miller, K., Al-Tamimi, A.K., Wolisz, A.: QoE-based low-delay live streaming using throughput predictions. ACM Trans. Multimed. Comput. Commun. Appl. **13**(1), 4–41 (2016)
3. Sodagar, I.: The MPEG-DASH standard for multimedia streaming over the internet. IEEE Multimedia **18**(4), 62–67 (2011)
4. Kua, J., Armitage, G., Branch, P.: A survey of rate adaptation techniques for dynamic adaptive streaming over HTTP. IEEE Commun. Surv. Tutor. **19**(3), 1842–1866 (2017)
5. Bouzakaria, M., Concolato, C., Feuvre, J.L.: Overhead and performance of low latency live streaming using MPEG-DASH. In: Proceedings of IISA 2014, pp. 92–97. United States (2014)

6. Wang, B., Ren, F., Zhou, C.: Hybrid control-based ABR: towards low-delay live streaming. In: Proceedings of ICME 2019, pp. 754–759. Shanghai, China (2019)
7. https://www.aitrans.online/MMGC/
8. Wei, B., Song, H., Wang, S., Kanai, K., Katto, J.: Evaluation of throughput prediction for adaptive bitrate control using trace-based emulation. IEEE Access **7**, 51346–51356 (2019)
9. Wei, B., Okano, M., Kanai, K., Kawakami, W., Katto, J.: Throughput prediction using recurrent neural network model. In: Proceedings IEEE 7th Global Conference on Consumer Electronics (GCCE), pp. 107–108. Nara, Japan (2018)
10. He, Q., Dovrolis, C., Ammar, M.: On the predictability of large transfer TCP throughput. ACM SIGCOMM Comp. Commun. Rev. **35**(4), 145–156 (2005)
11. Liu, Y., Lee, J.Y.: An empirical study of throughput prediction in mobile data networks. In: Proceedings of IEEE GLOBECOM 2015, pp. 1–6. San Diego, CA, USA (2015)
12. Wei, B., Kanai, K., Kawakami, W., Katto, J.: HOAH: a hybrid TCP throughput prediction with autoregressive model and hidden markov model for mobile networks. In: IEICE Transactions on Communications, E101. B(7), pp. 1612–1624 (2018)
13. Wei, B., Kawakami, W., Kanai, K., Katto, J., Wang, S.: TRUST: a TCP throughput prediction method in mobile networks. In: Proceedings of IEEE Global Commun. Conference (GLOBECOM), pp. 1–6. Abu Dhabi, UAE (2018)
14. Huang, T.Y., Johari, R., McKeown, N., Trunnell, M., Watson, M.: A buffer-based approach to rate adaptation: evidence from a large video streaming service. In: Proceedings of ACM SIGCOMM 2014, pp. 187–198. Chicago, IL, USA (2014)
15. Spiteri, K., Urgaonkar, R., Sitaraman, R.K.: BOLA: near-optimal bitrate adaptation for online videos. In: Proceedings of IEEE INFOCOM 2016, pp. 1–9. San Francisco, CA, USA (2016)
16. Yin, X., Jindal, A., Sekar, V., Sinopoli, B.: A control-theoretic approach for dynamic adaptive video streaming over HTTP. ACM SIGCOMM Comp. Commun. Rev. **45**(4), 325–338 (2015)
17. Mao, H., Netravali, R., Alizadeh, M.: Neural adaptive video streaming with pensieve. In: Proceedings of ACM SIGCOMM 2017, pp. 197–210. Los Angeles, CA, USA (2017)
18. Wei, B., Song, H., Wang, S., Katto, J.: Performance analysis of adaptive bitrate algorithms for multi-user DASH video streaming. In: Proceedings of IEEE WCNC 2021, pp. 1–6. Nanjing, China (2021)
19. Jiang, J., Sekar, V., Zhang, H.: Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with festive. IEEE/ACM Trans. Netw. **22**(1), 326–340 (2014)
20. Li, Z., et al.: Probe and adapt: Rate adaptation for HTTP video streaming at scale. IEEE J. Sel. Areas Commun. **32**(4), 719–733 (2014)
21. Zhou, C., Lin, C.W., Zhang, X., Guo, Z.: TFDASH: a fairness, stability, and efficiency aware rate control approach for multiple clients over DASH. IEEE Trans. Circuits Syst. Video Technol. **29**(1), 198–211 (2019)
22. Wei, B., Song, H., Katto, J.: FRAB: a flexible relaxation method for fair, stable, efficient multi-user dash video streaming. In: Proceedings of IEEE ICC 2021, pp.1–6. Montreal, Canada (2021)
23. HSDPA Dataset. http://home.ifi.uio.no/paalh/dataset/hsdpa-tcp-logs