An Application of Reinforcement Learning Algorithm at MEC for 4K Adaptive Video Streaming in NFV

Linh Van Ma[1], Sang Woo Kim[2], Jaehyung Park[3], Jinsul Kim[5*]
School of Electronics and Computer Engineering, Chonnam National University, Gwangju, South Korea
linh.mavan@gmail.com, swmax88@gmail.com, hyeoung@jnu.ac.kr, jsworld@chonnam.ac.kr

Um Tae-won[4]
Department of Information and Communication Engineering, Chosun University, Gwangju, South Korea
twum@chosun.ac.kr

***Abstract*—Streaming a video with ultra-high quality are dealing with current network inherent problems, such as bandwidth and latency. This paper proposes a reinforcement learning method in adaptive streaming to improve streaming quality as well as reduce latency in NFV. More specifically, we implement the reinforcement algorithm in Mobile Edge Computing (MEC), which is a network architecture, to delivery and receive streaming requests from clients. As a result, our proposed method reduces network latency up to 20% and improves 15% of quality compared to approaches in adaptive streaming for ultra-high quality.**

***Keywords*—mobile edge computing; adaptive streaming; reinforcement learning; NFV; latency, ultra-high quality**

I. INTRODUCTION

Dynamic adaptive streaming [1, 2] is a technique which provides efficient ultra-high quality streaming over the current Internet infrastructure. On the one hand, it provides and switches between the variety of video resolutions in one streaming session seamlessly. On another hand, it supports users deal with any unexpected network condition such as bandwidth fluctuation. However, the mechanism to select streaming resolution based on a certain network condition is still the hot topic of dynamic adaptive streaming recently. In fact, we previously proposed a fuzzy-based method [2] to overcome network bandwidth fluctuation problems. It mainly focuses on the switching algorithm between resolutions. Furthermore, we carried out a research about MEC to reduce network latency [1]. The MEC research paid attention to deliver a streaming service in a large network as well as balance a streaming server. Consequently, we address a new research on the switching mechanism utilizing reinforcement learning (RL) such as a study on deep Q-learning for dynamic adaptive streaming over HTTP (DASH) [3]. The Q-learning research aims to optimize the quality of experience (QoE) with faster convergence to chosen a streaming rate.

The contribution of this research is primarily focused on the adaptive algorithm in DASH employing a previous research on RL [4]. Moreover, we implement the algorithm on MEC to reduce latency as well as assist a client in reducing computational resources. Besides, the research environment is based on virtualization technology which is network function virtualization (NFV) [1, 5].

The article is organized as the following. In the Related Works section, we briefly introduce our based RL algorithm. Besides, we also address some applications and its advantage in NFV for multimedia. In the next section, we provide an overview of our proposed system in DASH. Experiment section evaluates our research. Finally, we conclude this study with future research direction.

II. RELATED WORKS

Network function virtualization (NFV) and software-defined networks (SDN) emerged as promising solutions; these enable the network to be flexible and easy to maintain. NFV is a complementary technology of SDN proposed for future 5G networks. It utilizes IT virtualization technologies to virtualize network node functions on top of standard general purpose hardware, which changes the communication network infrastructure [6].

There have also been extensive works on the task scheduling mechanism for mobile edge computing [7, 8]. In the research [8], the authors discussed Markov decision process to approach problem involving whether to execute a task locally on the mobile device or to offload a task to the MEC server for clouding computing. This work based on the queueing state of the task buffer, the execution state of the local processing unit, and the state of the transmission unit to schedule computational tasks. Take a different approach, [7] designed a distributed task scheduling mechanism based on game theory, in which contact duration, communication and computing cost, and monetary cost of the mobile devices are taken into account.

RL is the hot topic recently due to its application in many fields. The authors [4] proposed an asynchronous for deep RL. In detail, they utilized four RL methods and build them work together with parallel actor-learners. As a result, their proposed method outperforms compared to state-of-the-art methods with a multi-core CPU (Central Processing Unit) instead of GPU (Graphics Processing Unit).

III. Application of Reinforcement Learning for Adaptive Streaming

In this section, we first present the overview of our streaming system. Subsequently, we explain our two models using MEC to deliver adaptive streaming. More specifically, the first model is shown in Fig. 1. All requests from clients are passed through a MEC before reaching a central streaming server. In fact, MEC acts as a mediator to monitor QoE, Quality of Service (QoS) information of clients. Then, it analyzes data pattern which aims to predict the next suitable streaming rate for a certain client. As a result, it assists clients to retrieve DASH streaming resolutions in advance, which could lead reducing network latency.

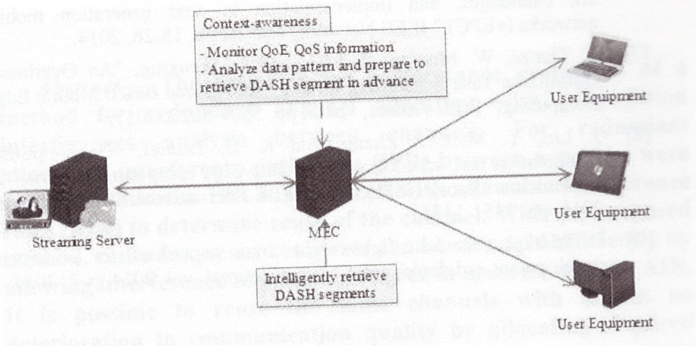

Fig. 1. Context-awareness at MEC to handle request from clients

In contrast, the second model is presented in Fig. 2. We only concern about one client compared to the first model that we have several clients which could be a large number. In fact, a client can stream with various CDN (Content Delivery Network) servers via MEC. For example, a client makes a video streaming request to MEC. Subsequently, MEC looks for CDN which has the requested video. Then, the MEC intelligently retrieves streaming data from different servers. The client might stream with CDN1 at 720p, CDN2 at 4K resolution (2160p) and CDN3 at 360p.

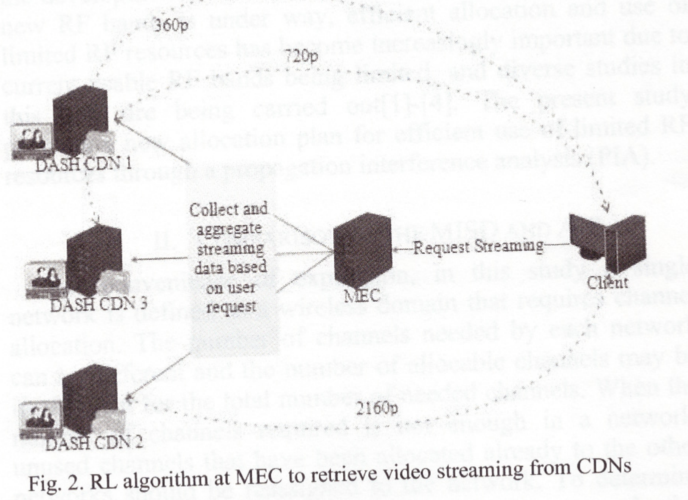

Fig. 2. RL algorithm at MEC to retrieve video streaming from CDNs

As addressed above, the two models need an intelligent component to control and adapt to uncertainties of network such as bandwidth fluctuation. In this study, we only consider a client as an actor making streaming requests. Besides, it does not spend any computational resource to achieve optimal streaming rate. All the actions of adaptive algorithms are done by MEC. This approach is quietly different compared to a client-based algorithm [3]. Moreover, this approach has another benefit when using MEC. That is, MEC can temporarily cache streaming data which has been requested by previous users. The cache memory assists clients to reduce network latency since MEC is really closed to those clients.

We employ asynchronous one-step Q-learning [4] which was introduced in the previous section. In detail, we proposed an algorithm to handle the two aforementioned models above. The detail of the algorithm is shown in TABLE I. Recall that, the algorithm is deployed and implemented in MEC. Moreover, we can have several MEC depending on the location of clients.

TABLE I. Application of Reinforcement Learning Algorithm in MEC

```
Initialize number of clients (i), CDN (j), MEC
While (stop streaming request)
    Clients request streaming;
    For cl=0 to i // MEC handle request from clients
        If (client = cl)
            break;
        End
    End
    Applying RL Algorithm() [4];
    RetrieveDASH Segment();
    TemporarilyCaching Retrieve Data();
    SendDataToClient();
End
```

IV. Experiment

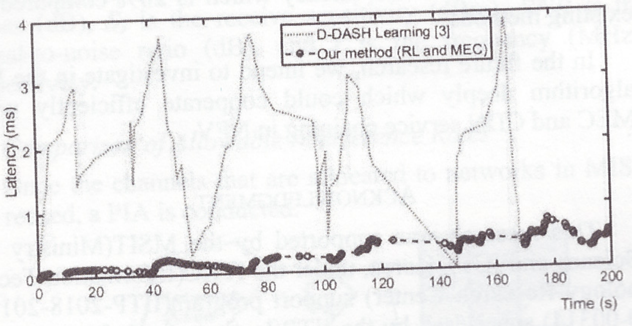

Fig. 3. Latency comparison between our method and the client-based algorithm

We evaluate our proposed method by implementing the two models as shown in Fig. 1 and Fig. 2. More specifically, we implement five CDN servers, two MEC servers, and twenty-five clients. The first fifteen clients are assigned to one MEC and another rest are located near another MEC. All of those clients are Ubuntu 16.04 virtual machines using Docker [9]. In the CDN, we simulate an adaptive streaming service with quality rates are integer numbers. In fact, the real adaptive streaming which has been implemented in many service providers such as YouTube has streaming rates which are discrete numbers. These rates have a huge different gap. However, we simulate the adaptive streaming aims to provide an accurate of rate estimation in adaptive streaming rather than an approximation.

The RL algorithm is implemented in the two MECs. As a result, Fig. 3 illustrates the latency comparison between our proposed method and D-DASH learning algorithm [3]. It shows that we obtain lower latency which is about 20% under 200 seconds of a streaming season. Besides, we compare streaming quality between the application of RL in this research to our previous MEC-based method [1] as shown in Fig. 4. We archive higher streaming quality which is about 15% with shorter streaming duration.

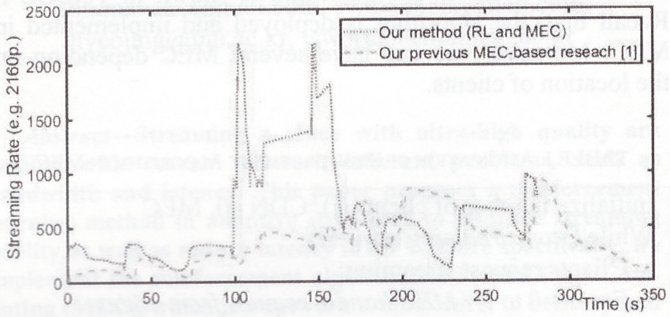

Fig. 4. Streaming quality comparison between our method and our previous research on NFV

V. Conclusion

In this article, we carried out research on an application of reinforcement learning algorithm. The algorithm is implemented in MEC with two streaming models. In the experiment, we simulate an adaptive streaming service to obtain the accurate estimation of streaming bitrate. As a result, we achieve high streaming quality which is 15%. Besides, our method lowers network latency which is 20% compared to existing methods.

In the future research, we intend to investigate in the RL algorithm deeply which could cooperate efficiently with MEC and CDN service chaining in NFV.

Acknowledgment

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2018-2016-0-00314) supervised by the IITP(Institute for Information & communications Technology Promotion). Besides, this research was supported by the IT R&D program of MSIT(Ministry of Science and ICT), Korea / NIPA(National IT Industry Promotion Agency) 12221-14-1001, Next Generation Network Computing Platform Testbed.

References

[1] L. Van Ma, V. Q. Nguyen, J. Park, and J. Kim, "NFV-Based Mobile Edge Computing for Lowering Latency of 4K Video Streaming," in 2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN), 2018, pp. 670-673.

[2] L. V. Ma, J. Park, J. Nam, H. Ryu, and J. Kim, "A Fuzzy-Based Adaptive Streaming Algorithm for Reducing Entropy Rate of DASH Bitrate Fluctuation to Improve Mobile Quality of Service," Entropy, vol. 19, p. 477, 2017.

[3] M. Gadaleta, F. Chiariotti, M. Rossi, and A. Zanella, "D-DASH: A deep Q-learning framework for DASH video streaming," IEEE Transactions on Cognitive Communications and Networking, vol. 3, pp. 703-718, 2017.

[4] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, et al., "Asynchronous methods for deep reinforcement learning," in International conference on machine learning, 2016, pp. 1928-1937.

[5] N. Bouten, J. Famaey, R. Mijumbi, B. Naudts, J. Serrat, S. Latré, et al., "Towards NFV-based multimedia delivery," in Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on, 2015, pp. 738-741.

[6] H. Hawilo, A. Shami, M. Mirahmadi, and R. Asal, "NFV: state of the art, challenges, and implementation in next generation mobile networks (vEPC)," IEEE Network, vol. 28, pp. 18-26, 2014.

[7] L. Tianze, W. Muqing, Z. Min, and L. Wenxing, "An Overhead-Optimizing Task Scheduling Strategy for Ad-hoc Based Mobile Edge Computing," IEEE Access, vol. 5, pp. 5609-5622, 2017.

[8] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in Information Theory (ISIT), 2016 IEEE International Symposium on, 2016, pp. 1451-1455.

[9] D. Merkel, "Docker: lightweight linux containers for consistent development and deployment," Linux Journal, vol. 2014, p. 2, 2014.