4K 스트리밍 품질 향상을 위한 MEC 환경에서의 QoE 기반 강화학습

Ashiquzzman AKM, 이동수, 오승민, 김상우, 김진술
전남대학교 공과대학 전자컴퓨터공학과
zamanashiq3@gmail.com
ready1819@gmail.com
osm5252kr@gmail.com
swmax88@gmail.com
jsworld@jnu.ac.kr

엄태원
조선대학교 IT융합대학 정보통신공학부
twum@chosun.ac.kr

QoE-based Reinforcement Learning at MEC for Improving 4K Streaming Quality

Akm Ashiquzzaman, Dongsu Lee, Oh Seung Min, Sang Woo Kim, Jin sul Kim
School of Electronics and Computer Engineering
Chonnam National University
77, Yongbong-ro, Buk-gu, Gwangju 500-757, South Korea
{zamanashiq3, ready1819, osm5252kr, swmax88}@gmail.com,
jsworld@jnu.ac.kr

Um Tae-won
Department of Information and Communication Engineering
Chosun University
Seonam-dong, Dong-gu, Gwangju, South Korea
twum@chosun.ac.kr

요 약

4K or ultra-high-definition (UHD) will be the standard for video streaming in the next decade. In this research, we carry a study on dynamic adaptive streaming over HTTP (DASH) which is a crucial adaptive algorithm. More specifically, we employ asynchronous reinforcement learning (RL) algorithm in the form of quality of experience (QoE). In fact, QoE is an important factor to evaluate the efficiency of streaming transmission models. Hence, we made every streaming request must be passed through a MEC server, and the MEC make a decision to retrieve segments for clients. Experiment result shows that our proposed QoE RL algorithm achieves higher quality compared to our previous research on the same context of content-awareness.

Index Terms－QoE, MEC, NFV, Reinforcement Learning, adaptive streaming

1. Instruction

In the past, the concept of Quality of Service (QoS) has been brought to the attention of a large number of users as well as service providers and network operators. QoS also has high the investment rate of telecom operators as well as high concentration of the network research community, leading to solutions that ensure highly stable and highly efficient. Compared with the concept of QoS, QoE is a newer concept. QoS simply gives users a fairly technical sense of service quality. QoS is primarily focused on describing the objective, technical criteria that the network infrastructure or application needs to achieve in order to guarantee quality service. In other words, QoS can be considered as the general technical language of the quality that applications and network infrastructure used. Consequently, we must establish a general description model which easy to understand for end-users of a service. That is the concept of QoE. In fact, QoE is a common language for applications and end users. It is used in the approach of a user to evaluate QoS. In other words, QoE is a measure of the satisfaction of people with the service they are using based on subjective judgment. Thus, QoE can be synthesized from pure QoS and other non-technical factors such as the characteristics of the human visual and auditory system, etc. The emergence of the QoE concept will most likely lead to certain changes in the market approach of telecom service providers. Rather than focus on QoS, QoE-related issues will be centered.

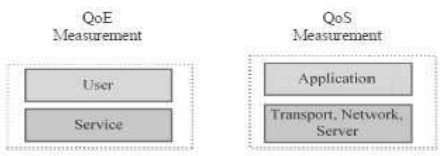

Fig. 1. Difference between QoE and QoS.

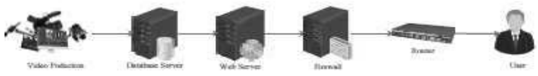

Fig. 2. Video delivery process from video production to user.

The development and implementation of SLAs (Service-level agreement) are based on QoE customers. Besides, applying QoE-based service-adjustment policies is just one example of a trend for growth in the coming years.

The difference between QoE and QoS is shown in Fig. 1. It shows that QoE mainly focuses on user experience (human perspective). In oppose, QoS pays attention to applications with machine perspective based on the transport network and server infrastructure. In fact, most of the services are provided for users (human). Hence, QoE would be an important factor to evaluate a service from providers. Multimedia delivery processes are facing many problems over the Internet because it must pass through many components of the Internet infrastructure which is deployed and implemented by service providers. For example, as shown in Fig. 2, a video is produced in the Video Production department, and it is starting to be delivered from a database server, web server, firewall, and router. Video packets go through these components might has some error which degrades video quality. Therefore, it is crucial to measure the quality of the video (measure QoE) after transmitted compared to the original video and evaluate the streaming system. In consequence, we can detect and understand the causes of our streaming system. As a improve the system.

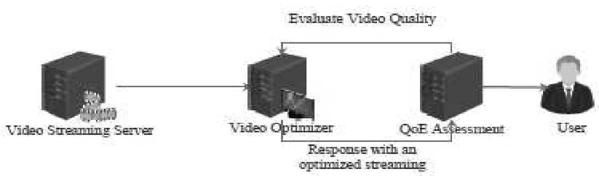

Fig. 3. Overview of system streaming with QoE assessment component.

2. RELATED WORKS

This research was inspired by our previous research [1] about MEC with content-awareness component which is placed at MEC to retrieve DASH information for clients. On the basic of research on fuzzy logic to obtain DASH segments with high quality [2], we deploy segment selection for DASH streaming to MEC. As a result, it reduces network latency as well as the computation resource of clients with high streaming quality.

3. SYSTEM OVERVIEW

The overview of our proposed system streaming system is shown in Fig. 3. A video is available on a Video Streaming Server which supports DASH (Dynamic Adaptive Streaming over Hyper Transfer Text Protocol). Then, video packets are transferred to a user via a Video Optimizer and a QoE assessment component. Every streaming session the Video Optimizer optimizes streaming video to reduce network resource consumption such as bandwidth. Subsequently, it forwards video packets to QoE assessment which evaluates the streaming quality session. If the streaming session is not optimized yet, it sends back a message to the Video Optimizer to improve the video quality.

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. 4. 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.

In contrast, the second model is presented in Fig. 5. 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.

In contrast, the second model is presented in Fig. 5. 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.

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.

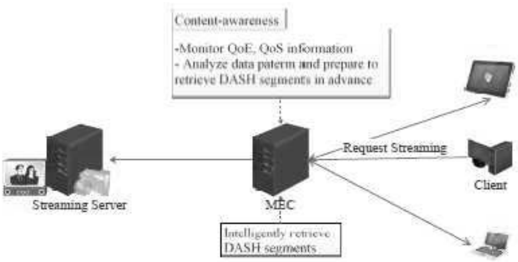

Fig. 4. Context-awareness at MEC to handle request from clients.

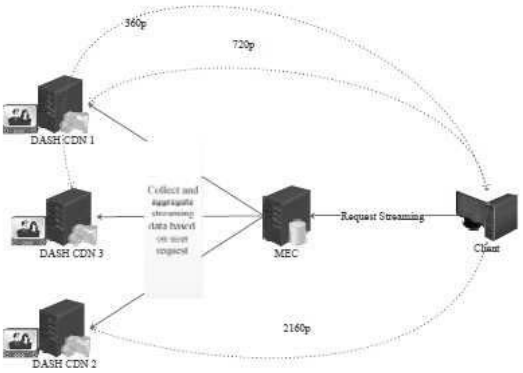

Fig. 5. RL algorithm at MEC to retrieve video streaming from CDNs.

We employ asynchronous one-step Q-learning [3] 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.

In the research, we also apply a concept of CDN (Content Delivery Network) slides [4] as shown in Figure 17. More specifically, a CDN slice consists of one coordinator, one or more transcoders, streamers, and caches. Once a CDN slice is created, the CDN owner can manage his videos through the coordinator, which is a mandatory component used to manage the entire CDN slice including caches, transcoders, and streamers. It enables users to upload, modify or delete videos, select the preferred transcoder, cache, and streamer among the available ones to transcode the video to desired resolutions, store and stream the transcoded videos. Caches are mainly in charge of storing videos after being uploaded by users and after being transcoded by the selected transcoder server. Transcoders get a request from the coordinator and transcode videos at rates specified by the CDN owner. The role of streamers is load balancing and receiving end-users requests for playing a specific video and redirecting the request to a proper cache server to show the video content using available resolutions.

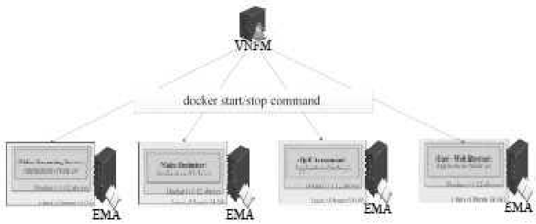

Fig. 6. System implementation with Docker and NFV.

4. EXPERIMENT

We employ FNCP (Future Network Computing Platform) [1], [5], [6] to build a video delivery streaming system. The FNCP is built based on the NFV (Network Function Virtualization) proposed by ETSI (European Telecommunications Standards Institute). Thanks to its virtualized functions, we can quickly place a database server or a web server within a short time (in minutes). Besides, we can also reorganize or re-construct network functions if we want to add or remove a component aiming to improve streaming quality. This task was intensively in the past because all of the tasks were done/processed with physical machines (computers). The illustration of our system is shown in Fig. 6. We measure QoE between Service and User under a system infrastructure deployed by network virtualized functions.

In the NFV environment, we manage virtualized network functions (VNF) by using a VNFM (Virtual Network Functions Manager) [7]. Each VNF represents as an EMA (Element Management Agent) which runs on Ubuntu 14.04. In the EMA, we install Docker [8]-[10] with a version greater than 1.12 which supports us deploy our CDN streaming system easily. Inside the Docker, we implement our source code for each component such as Video Streaming Server, Video Optimizer, QoE Assessment and User with a web browser. In fact, User component is not included in the streaming system but we use it for testing purpose. The VNFM can manage the VNFs by using docker start/stop commands. By doing this way, we can quickly add or remove a component from our streaming system. The details of the streaming infrastructure are shown in Figure 18.

We compare the result with our previous research on context-awareness [1].

5. CONCLUSION

4K or ultra-high-definition (UHD) will be the standard for video streaming in the next decade. In this research, we carry a study on dynamic adaptive streaming over HTTP (DASH) which is a crucial adaptive algorithm. More specifically, we employ asynchronous reinforcement learning algorithm in the form of quality of experience (QoE). In fact, QoE is an important factor to evaluate the efficiency of streaming transmission models. In this research article, we mainly focus on QoE performance analysis of streaming models to improve and ensure high quality (4K and UHD) streaming service in NFV.

ACKNOWLEDGEMENT

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2018-20160-00314) supervised by the IITP(Institute for Information & communications Technology Promotion). Besides, this research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science, and Technology (MEST)(Grant No. NRF-2017R1D1A1B03034429). Lastly, this work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT). [2018-0-00691, Development of Autonomous Collaborative Swarm Intelligence Technologies for Disposable IoT Devices]

REFERENCE

[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). IEEE, 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, no. 9, p. 477, 2017.

[3] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, “Asynchronous methods for deep reinforcement learning,” in International conference on machine learning, 2016, pp. 1928 – 1937.

[4] S. Retal, M. Bagaa, T. Taleb, and H. Flinck, “Content delivery network slicing: Qoe and cost awareness,” in Communications (ICC), 2017 IEEE International Conference on. IEEE, 2017, pp. 1 – 6.

[5] H. B. Lee, S. I. Kim, and H. S. Kim, “A fault management system for nfv,” in Information Networking (ICOIN), 2018 International Conference on. IEEE, 2018, pp. 640 – 645.

[6] S. H. Yoon, T. H. NA, and H. Y. Ryu, “Apparatus for testing and developing products of network computing based on open-source virtualized cloud,” Nov. 2 2017, uS Patent App. 15/217,435.

[7] R. Mijumbi, J. Serrat, J.-L. Gorricho, S. Latre´, M. Charalambides, and D. Lopez, “Management and orchestration challenges in network functions virtualization,” IEEE Communications Magazine, vol. 54, no. 1, pp. 98 – 105, 2016.

[8] B. I. Ismail, E. M. Goortani, M. B. Ab Karim, W. M. Tat, S. Setapa, J. Y. Luke, and O. H. Hoe, “Evaluation of docker as edge computing platform,” in Open Systems (ICOS), 2015 IEEE Confernece on. IEEE, 2015, pp. 130 – 135.

[9] D. Merkel, “Docker: lightweight linux containers for consistent development and deployment,” Linux Journal, vol. 2014, no. 239, p. 2, 2014.

[10] J. Stubbs, W. Moreira, and R. Dooley, “Distributed systems of microservices using docker and serfnode,” in Science Gateways (IWSG), 2015 7th International Workshop on. IEEE, 2015, pp. 34 – 39.