Cooperating SDN and MEC for Lowering Latency of Adaptive Ultra-high Video Streaming in NFV

[1]Linh Van Ma, [2]Sang Woo Kim, [3]Akm Ashiquzzaman, [5]Jinsul Kim
School of Electronics and Computer Engineering
Chonnam National University
77, Yongbong-ro, Buk-gu, Gwangju 500-757, Rep.of Korea
{linh.mavan, swmax88, zamanashiq3}@gmail.com, jsworld@jnu.ac.kr

[4]Kwangki Kim
School of IT Convergence
Korea Nazarene University
Seobuk-gu, Cheonan-si, 31172, Rep.of Korea
k2kim@kornu.ac.kr

***Abstract*—Ultra-high-definition (UHD) is expected to become the standard for video streaming over the Internet in the next decade. This article addresses an aspect of delivering UHD content in the context of software-defined networking (SDN) cooperating with mobile edge computing (MEC) in the virtualized environment such as network function virtualization (NFV). More specifically, we virtualize a network environment using NFV. Then, we deploy MEC and streaming system on the NFV. Consequently, we employ the MEC concept to deliver UHD streaming. In the delivery scheme, we use software-defined networking (SDN) to control packages between MEC servers to optimally provide streaming to UHD streaming clients. In the experiment, we implement our virtualized environment as well as the streaming system. As a result, our streaming system achieves low latency which is about 20% compared to our previous research on NFV. Furthermore, our system outperforms compared with two research methods regarding SDN in the context of streaming quality.**

***Index Terms*—MEC, NFV, SDN, UHD, Adaptive Streaming**

I. Introduction

With the development of high-speed internet and easy access, demand for Ultra-high-definition (UHD) streaming is gaining rapid popularity. Regardless of end-devices computational resource, the delivery streaming process faces bandwidth, latency which might cause stalling while playing back a video at the client-side. Such streaming service needs to handle complex applications as well as tremendous device types. For example, a social media service does not only support a virtual space for a user but also provide a streaming service either low or ultra-high quality for users. Besides, users can have different characteristics because they use different devices to connect to social services. This demand requires a network infrastructure to handle the problem of ultra-low latency, high scalability, and reliable data transmission.

Software Defined Network (SDN) and Network Function Virtualization (NFV) has now become an industry standard for implementing high scaled reliable networks. Recently, the European Telecommunications Standards Institute (ETSI) proposed a network solution which is called Mobile Edge Computing (MEC) [1] to overcome the aforementioned problem. Though, the recent research with MEC still is cumbersome such as handing load balancing between MEC streaming servers. More specifically, some MEC servers handle computational requests from users while other MECs are idle, which causes increasing service latency because heavy load MEC servers cannot transfer requested requests to idle MEC servers. We can solve this problem by placing a coordinator server which independently executes instances of MEC application. Besides, it can also decide to choose an optimal MEC server for a given request from a user. In fact, this coordinator can be handled by software-defined networking (SDN) concept [2], [3].

The contribution of this paper is mainly focused on SDN for MEC in delivering UHD streaming with the following reasons. SDN enables us to deploy better network functionality without the concern for hardware compatibility. MEC gives us more adaptability to reduce latency while providing computation offloading in the Dynamic Adaptive Bitrate Streaming for HTTP or MPEG DASH. MPEG-DASH is a universally accepted method for streaming, and it is deployed to real streaming service providers such as YouTube. However, the main drawback of DASH is the latency for media streaming. Although many cutting-edge researches have been carried out to address the issue improving DASH with reduced latency. Furthermore, implementing MEC for this problem has not been researched properly yet.

The article is organized as the following. In the next section, we address related research about SDN and NFV with adaptive streaming. In Section III, we present our SDN-based adaptive streaming in MEC. Consequently, we simulate the system using a visualized environment implemented by the Electronics and Telecommunications Research Institute (ETRI). Lastly, we conclude our research with the limitations of our research and the future investigation of the UHD streaming transmission.

II. Related Research

In the paper by Zhao et al. [4], they reviewed all of the current cutting-edge technologies of SDN assisted DASH methods. In their research, they have explored the experiential setup with MPEG-DASH encoded video datasets to evaluate their results. They used the SDN network topology as shown in Fig. 1. Experimental MPEG-DASH server was prepared with Ryu SDN Controller and OpenFlow v1.3 as southbound APIs. The forwarding devices were simulated using Openvswitch (OVS v2.3.1). One DASH Apache web server followed by one DASH client were deployed, where 10 Mbps bandwidth

978-1-5386-5553-5/19/$31.00 ©2019 IEEE

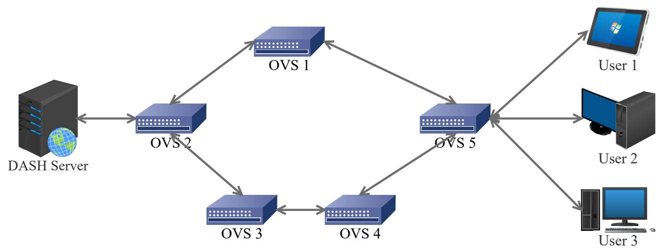

Fig. 1. SDN-assisted adaptive streaming framework.

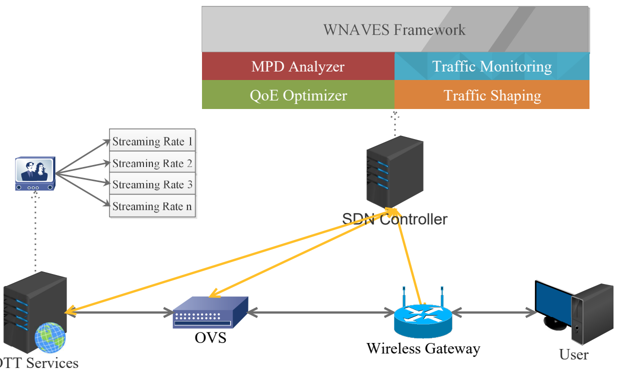

Fig. 2. Wireless network assisted video streaming (WNAVS).

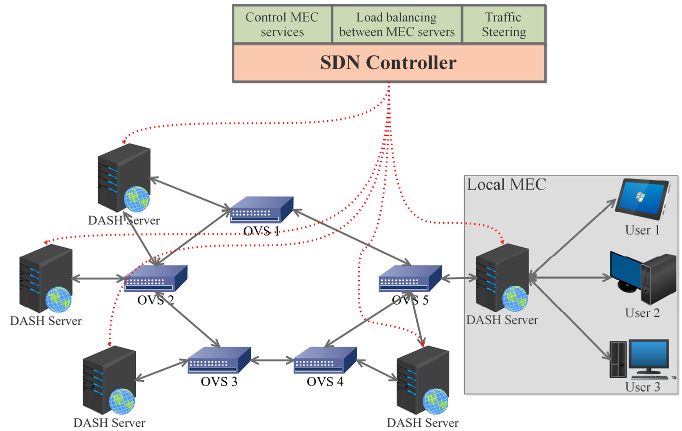

Fig. 3. Cooperating between SDN and MEC for adaptive streaming.

allocation was maintained for connection link. They used the Astream Media Player, a python based command line tool to emulate media player and evaluated the DASH player performance. Their proposed general application interface regarding a traffic flow alteration mechanism used optimization in the bitrate for the viewer's Region Of Interest (ROI). This proposed framework enables QoE utilizing the average ROI's bitrate in both fixed ROI and ROI switchover cases.

One of the most common scenarios in a wireless home network is the multiple client access simultaneously at a time. The situation results in a bottleneck problem for the network, which later causes poor performance for video streaming. The authors in the research in [5] proposed a new framework WNAVS (Wireless Network Assisted Video Streaming) that relies on the deployment of SDN. As shown in Fig. 2, their proposed framework WNAVES deals directly with DASH during high demand access to improve QoE. WNAVES maintains proportional bandwidth allocation to a client as a mechanism to prevent concomitant transmission. They implemented WNAVS framework in NS3 simulator [6]. They have introduced new module in NS3 for DASH as NS3 currently did not offer such functionality. Their adaptive framework performed overwhelmingly over traditional networks.

Mobile Edge Computing is a new concept for modern network architecture. MEC addresses the issue of processing of data by distributing the process in the outer layer with the help of NFV. Modern network devices are now more data exclusive than ever. As a result, with a newer application, more processing time, power and memory are needed to handle user requests. Higher data processing time causes delay, also known as latency. The authors [7] proposes a new process with NFV based MEC that eventually lowered the latency of 4K Ultra High Definition (UHD) Video Streaming. The proposed model is an NFV-based MEC to transmit ultra-high quality multimedia with DASH utilized for efficiency. The whole system provided 4K and 8K video streaming effectively and lower latency up to 10% than traditional methods.

III. SDN-based Streaming System

The first propose of MEC from ETSI relies on a virtualized environment such as NFV. In fact, NFV supports Virtual Network Functions (VNF) as a software which can be considered as a middle-box, to build a network service. Besides, we can build a service chaining of VNFs, which can be easily deployed and implemented on MEC. While we can flexibly build a service chain in a local service, we can also enable more dynamism and flexibility by adding SDN to the MEC platform to the network service. Moreover, we can obtain the global view of SDN allows for a global view of a network structure by interacting with the SDN controller. As a result, we can implement traffic steering rules which can be applied to any complex service chaining scenarios.

As shown in Fig. 3, we consider MEC as an SDN application and use SDN northbound interface to communicate with the SDN controller. The SDN controller monitor services which run on MEC servers to optimally choose a MEC server for a given request coming from a user. More specifically, it can optimize a service based on different conditions, such as latency, computational capacity, storage. Besides, we employ northbound APIs to trigger commands from the SDN controller SDN applications, such as routers or switches (OpenFlow switches).

MEC coordinator northbound application commands are high level and cannot send command directly to the devices. For resolving this problem, the command is later translated into OpenFlow-based traffic steering rules and transfer to the OpenFlow Devices. This component can both be connected to the MEC server or be part of it, in the network. Such rules for these specialized servers can be added with the help of

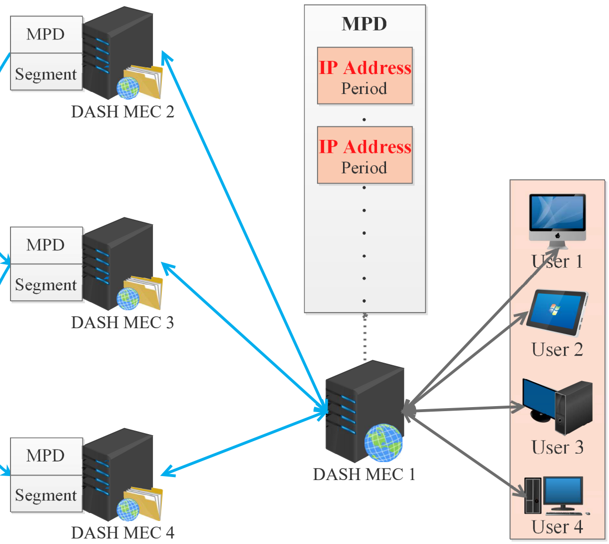

Fig. 4. MEC-based adaptive streaming system.

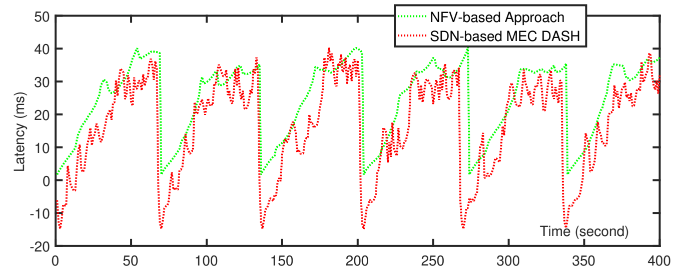

Fig. 5. Latency comparison between our proposed system with NFV-based DASH.

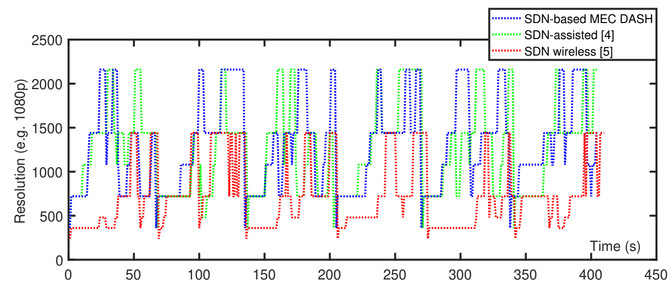

Fig. 6. Resolution comparison between our proposed system with other methods.

the OpenFlow rules as "traffic offload service" running on the MEC servers. This service is the main mechanism for traffic routing for the MEC applications.

The design of the MEC server is crucial to ensure that the SDN controller predetermines the best path because the service continuity depends on the shortest time. SDN controller redistributes MEC server periodically and updates the network flow for the best fit. But this process renders useless upon the introduction of MEC. It is not optimal for the computation being moved to a new MEC server.

The overview of our adaptive streaming system is shown in Fig. 4. MEC servers are located in a location where is near to users. Besides, each MEC server runs an adaptive streaming service which has MPD (Media Presentation Description) [8] and segment files corresponding to available streaming videos. Furthermore, it contains another record which stores information of available videos in different DASH MEC servers. With the IP addition in the record, a client can stream with different CDN servers. We locate the best streaming server for each video streaming resolution. For example, the client can optimally stream with MEC-1 with 360p resolution, streaming with MEC-3 with 720p resolution, and MEC-2 with 2160p which is the 4K resolution. Broadly, with the extension of multi-streaming MEC, we create a table for each streaming segment.

IV. Experiment

In this section, we discuss the overall experimental setup for our proposed method for adaptive streaming (DASH) with research outcome. In short, we compare latency between our proposed system with the NFV-based adaptive streaming system [7]. Besides, we evaluate and examine the improvement of streaming quality of our system compared with other previous research on the same field of SDN adaptive streaming [4], [5].

Mininet is very common network simulator often used for SDN simulation. For our experiment, we follow the processes described by Zhao et al. [4] and used Ryu SDN Controller and OpenFlow v1.3 as southbound APIs for the network. The whole network topology for the implemented network was described in Fig. 4. We allocate ten MEC servers and one-thousand users on the experiment. Each MEC is responsible for 100 connected local users. We implement a component to measure DASH streaming information, such as packets latency, streaming resolution on each user. SDN Controller assists MEC by routing its packets where OpenSwitch is placed in each MEC server. We employ AStream [9] which supports a rate adaptation model for DASH on Mininet. The whole setup was constructed with 10 Mbps link bandwidth and $1ms$ latency for real-life application emulations. The virtualized network and MEC server are constructed using FNCP (Future Network Computing Platform) which is also the testbed environment for our previous research [7].

The main focus for this research is to improve streaming quality of UHD videos. Hence, we implement MPEG-DASH (Moving Picture Experts Group) as a streaming technique in our experiment. In detail, We use a raw video file with duration is about six and a half minutes. Then, we encode and transcode UHD videos into different resolutions and fragment these resolutions into segment files which are the format files for MPEG-DASH. This task is separated from our network topology and it is implemented using Node.js and an FFMPEG (Fast Forward Motion Picture Experts Group) Node.js module. For simplicity, we decode an original UHD video with the following bitrates $\{144, 240, 360, 480, 720, 1080, 1440, 2160\}$ progressive video.

We randomly select one user among one-thousand users and obtain measured streaming information. At first, Fig. 5 illustrates latency comparison between our method and NFV-based adaptive streaming method [7]. It shows that we achieve low latency which is about 20% compared to our previous research. Two methods employ MEC as a technique to deliver UHD content over the Internet, but the SDN-based approach lowers latency because it can control and balance the working load between MEC servers. In second step, we compare streaming quality between three methods which are SDN-assisted [4], SDN wireless [5] and our proposed method, SDN-based MEC DASH. For simplicity, we only calculate the number of download bit during a streaming session for each method. As a result, Fig. 6 shows that our method obtained 5% downloaded bits compared to SDN-assisted and 40% downloaded bits compared SDN wireless method. The difference between two SDN-based approaches and our approach is a MEC supporting component. The two approaches without MEC cannot handle massive requests efficiently from one-thousand users since all of the users try to get UHD streaming quality which is a heavy task with high bandwidth consumption. In fact, DASH streaming is sensitive to bandwidth fluctuation and it might drop streaming resolution if the throughput is reduced. In oppose, our approach employs MEC which supports users to locally handle requests from fewer users (one-hundred), which prevents heavy bandwidth consumption burden on the back haul links between MEC and a streaming server.

V. Conclusion

In this article, we carried out research on SDN cooperating with MEC and NFV. The main contribution of this research was focused on the improvement of UHD streaming over the Internet. In fact, we controlled services which are deployed at MEC servers and balance load between MEC servers. Though we put lots of effort to built the system, our measurement on a streaming session was stopped at two measurement parameters which are packets latency and streaming representation. Besides, we did not involve the research of adaptive algorithm for DASH which significantly affects the quality of experience for our network topology. Consequently, in the future research, we intend to investigate in reinforcement learning algorithms at MEC and SDN controller as well. As a result, we can make a local service which is served by MEC intelligently.

Acknowledgment

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 and 2017R1D1A3B03034951). Also, 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). Furthermore, 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] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing-a key technology towards 5g," *ETSI white paper*, vol. 11, no. 11, pp. 1–16, 2015.

[2] E. Haleplidis, K. Pentikousis, S. Denazis, J. H. Salim, D. Meyer, and O. Koufopavlou, "Software-defined networking (sdn): Layers and architecture terminology," Tech. Rep., 2015.

[3] L. Cui, F. R. Yu, and Q. Yan, "When big data meets software-defined networking: Sdn for big data and big data for sdn," *IEEE network*, vol. 30, no. 1, pp. 58–65, 2016.

[4] S. Zhao and D. Medhi, "Sdn-assisted adaptive streaming framework for tile-based immersive content using mpeg-dash," in *Network Function Virtualization and Software Defined Networks (NFV-SDN), 2017 IEEE Conference on*. IEEE, 2017, pp. 1–6.

[5] R. M. Abuteir, A. Fladenmuller, and O. Fourmaux, "An sdn approach to adaptive video streaming in wireless home networks," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2016 International*. IEEE, 2016, pp. 321–326.

[6] S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnsim 2: An updated ndn simulator for ns-3," *Dept. Comput. Sci., Univ. California, Los Angeles, Los Angeles, CA, USA, Tech. Rep. NDN-0028*, 2016.

[7] 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.

[8] 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.

[9] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A buffer-based approach to rate adaptation: Evidence from a large video streaming service," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4, pp. 187–198, 2015.