Video Quality Management over the Software Defined Networking

Is-Haka Mkwawa, Alcardo Alex Barakabitze and Lingfen Sun
School of Computing, Electronics and Mathematics, Plymouth University, UK
E-mail: {is-haka.mkwawa,AlcardoAlex.barakabitze,l.sun}@plymouth.ac.uk

***Abstract*—Dynamic Adaptive Streaming over HTTP (DASH) or MPEG-DASH is a popular technique that allows video quality adaptation for high quality streaming over the Internet. However, with bandwidth fluctuations, DASH performs poorly due to annoying frequent number of stalls. Software Defined Networking (SDN) has emerged as an attractive technology which has found its way into datacentres. Since one of the main goals of the SDN architecture is to make the network programmable and accelerate network innovation by utilizing its control plane, this paper has used the SDN control plane to propose a video quality management scheme based on the traffic intensity under DASH. Experimental results obtained by using Mininet and OpenDaylight controller have shown that the proposed scheme can significantly reduce the number of frequently annoying stalls and their duration by at least 84% and 94%, respectively. This has been achieved by switching network flows from high to low congested network paths within an SDN architecture.**

***Keywords*—Software Defined Networking, Quality of Experience, OpenFlow, Multimedia Communication, Multimedia Services, Traffic Intensity, Performance Modelling and Analysis.**

I. Introduction

Recently, Cisco has released the complete Visual Networking Index (VNI) Global IP Traffic Forecast for 2015-2020. It projects that, globally, IP video will represent 82% of all traffic by 2020. Annual global IP traffic will reach 2.3 Zettabytes per year by 2020 [1]. With such an unprecedented amount of data, researchers in both industry and academia are posed with pressing challenges on how to handle such traffic and at the same time provide acceptable Quality of Experience (QoE) in multimedia communication and services.

As a measure to address these challenges, Software Defined Networking (SDN) has emerged to be one of the promising solutions. One of the main goals of SDN is to make the network programmable at the control plane by decoupling the network control plane and the forwarding plane from conventional switches and routers. The SDN architecture accelerates network innovation by using the OpenFlow protocol [2], also known as a Southbound API, which is an interface between the control plane and the forwarding plane. The forwarding plane provides forwarding functionality under the instructions of the control plane. The Northbound interface such as RESTful API Web services connects the control plane and network applications. The control plane maintains the global SDN information for management and optimization of the SDN architecture.

SDN has the ability to provide secure and reliable end-to-end communications that can satisfy specific transmission requirements [3], therefore, achieving stringent requirements on network transmission latency for specific networking services such as video streaming and online gaming.

In SDN, all transactions in the forwarding plane involve the controller by which a forwarding device such as a switch or a router has to request for forwarding rules when the first packet of each flow arrives at the forwarding device. Frequent communications also involve the controller whenever there is an update of forwarding rules or statistical data requests by the controller to the forwarding device. In this context, the SDN controller becomes the bottleneck in the SDN architecture and thus, can cause QoE degradation of delivered multimedia services. It is therefore, crucial to understand the performance of the SDN architecture under difference traffic patterns, especially the bursty nature of video streaming.

The European Network on Quality of Experience in Multimedia Systems and Services, Qualinet (COST Action IC 1003) [4] has defined the QoE as the degree of delight or annoyance of the user of an application or service. It results from the fulfillment of his or her expectations with respect to the utility and/or enjoyment of the application or service in the light of the users personality and current state. The QoE is influenced by several factors, including technical (e.g., network, device, content and applications) and non-technical ones (e,g., user expectations/preferences, environment and psychological impacts) [4]. Providing acceptable QoE to end users should be the ultimate goal for any service and network provider in order to improve churn rate and business growth.

Since the SDN controller has the global information of the underlying forwarding devices (e.g., packet loss and available bandwidth), it is tasked to issue instructions and forwarding rules in order to modify network flows if needed, and take different path if network congestion is detected in one of the forwarding devices due to queuing delay. Modifying network flows by switching to low congested paths can improve video quality.

Dynamic adaptive streaming over HTTP (DASH) is currently a popular technology for video streaming over the Internet because it allows video quality adaptation based on the network conditions. However, DASH has negative impact on video quality in a scenario that requires frequently bitrates and resolution switching [5]. It was also found to perform poorly under the presence of bursty traffic [6]. To mitigate the disadvantages of DASH, video quality management scheme based on the traffic intensity could be used to complement the

DASH in a scenario whereby all the SDN network flows are congested.

To this end, this paper proposes a traffic intensity based video quality management scheme over an SDN architecture. Traffic intensity has also been used in telecommunications industry to improve the call blocking rate by introducing a threshold on the emergency and general calls [7].

The main point of focus in this study is on the performance of the control plane and particularly the OpenFlow protocol. The Mininet [8], a network emulator is used to create network of virtual hosts, switches, controllers, and links. The OpenDaylight controller [9] is used as an SDN controller which allows developers to write network applications that can easily work across a wide variety of forwarding devices and southbound protocols.

The preliminary results show that the proposed video quality management scheme over an SDN architecture can significantly reduce the number of stalls and their duration by more than 84% and 94%, respectively.

The rest of this paper is organized as follows. Section II provides the related work relevant to this research. Section III describes the experimental setup which included the testbed and video sequences used in the experiment. The proposed video quality management scheme based on the traffic intensity is presented in Section IV. Results and discussions are outlined in Section V. Conclusions and future work in Section VI concludes this paper.

II. Related Work

The OpenFlow protocol is an open interface protocol used by the SDN controller to control the forwarding devices. Forwarding devices in the SDN are also known as OpenFlow forwarding devices such as OpenFlow switches and routers. The controller is commonly implemented remotely on a PC securely connected to forwarding devices. The forwarding devices hold flow tables which contain, packet headers for matching incoming packets, list of actions which must be followed to handle matched packets and collection of statistics for each flow such as number of packets and bytes received and transmitted. Fig. 1 depicts the packet flow through an OpenFlow forwarding device (OpenFlow Ver 1.3). The table-miss flow entry must specify how to handle packets that are unmatched, and the forwarding device may decide to send packets to the controller, drop them or direct them to a subsequent table.

Most of the SDN performance studies have been conducted under simulation experiments and analytical modelling. Jarschel et al. [10] proposed a model to estimate the packet loss probability and sojourn time for the SDN controller. The model based on the queuing theory captured the packet delay caused by the processing in the controller instead of the processing by just the forwarding device. Although the analytical model results were successfully validated by simulation based on OMNET++, the model did not capture the delay and packet loss caused by the ability of the control

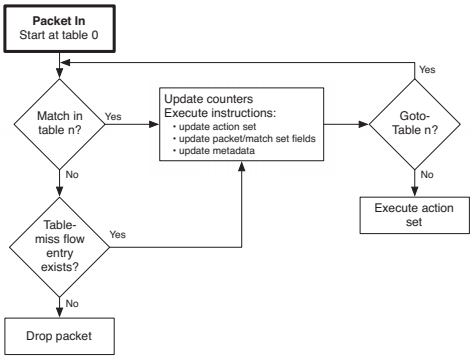


Fig. 1. The packet flow through an OpenFlow forwarding device

plane to modify network flows within a forwarding device or among the connected forwarding devices.

Authors in [11] developed a network visualization and performance prediction (NVPP) tool for SDN based on queuing theory analytical models. It predicts a network performance arising from traffic variations and a real time view of a network. This is the only paper that has been found to implement analytical models in the virtualized SDN architecture. However, the the proposed model has not been validated by either simulation or real world experiments.

The scalability of the SDN control plane was modeled and evaluated in [12]. Three SDN structures of the control plane, i.e. centralized, decentralized and hierarchical structures, were investigated and analyzed, and their scalabilty were compared. However, the proposed model has not been validated by simulation or real world experiments.

Liotou et al. [13] introduced the framework of an SDN based QoE service to guarantee QoE level for on demand services of Over The Top (OTT) applications by monitoring network parameters via an SDN controller. This framework also proposed the mapping of QoS parameters to QoE. However, the framework was neither simulated nor implemented in SDN emulators.

The SDN was used in [14] to propose a bandwidth management scheme to optimize the overall QoE under multiple streaming sessions. The proposed scheme tailored the bandwidth allocation based on the video content and playout buffer instead of equally allocating bandwidth among the competing network flows. However, the paper did not propose any procedure on how to deal with the congested network flows.

Kleinrouweler et al. [15] have used the global view of the SDN controller to provide signaling feedback on target bitrates to DASH players in order to maintain stable video quality under the SDN architecture and reduce frequent bitrate and resolution switches. Although this approach was demonstrated to work efficiently, with increasing bursty background traffic in the network, it could still result into disturbing and frequent bitrate and resolution switches. Another disadvantage of the

proposed approach is that, DASH players have to be modified in order to interface with the proposed service manager for signaling.

In this paper, the proposed video quality management scheme based on the traffic intensity over SDN does not need to modify the DASH players. The scheme is implemented to interface with the controller through the RESTful API Web service in order to issue network flows redirection caused by network congestion.

III. Experimental Setup

A. SDN testbed

The testbed implemented in this paper is depicted in Fig. 2. It uses Mininet version 2.2.0 [8] network emulator to create a network of virtual hosts, switches and links. Mininet was running in Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86_64) with 1GB of RAM and Intel(R) Xeon(R) CPU E5-2609 v3 @ 1.90GHz.

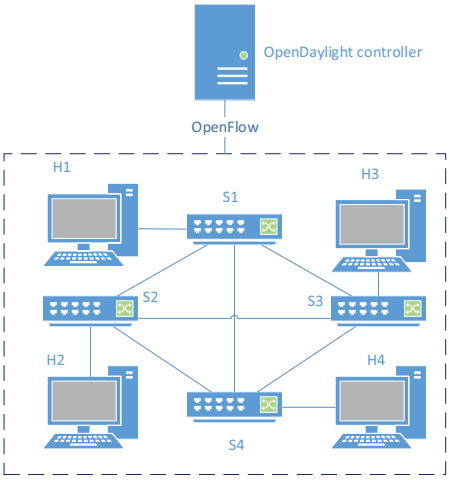


Fig. 2. The SDN testbed

The OpenDaylight Beryllium (Be) controller [9] has been used as a controller which uses RESTful API Web services to interface (North-bound) with network applications and the OpenFlow version 1.3 protocol as an interface (South-bound) between virtual switches and the controller. The controller was running under Ubuntu 14.04.5 LTS (GNU/Linux 3.19.0-68-generic x86_ 64) with 2GB of RAM and Intel(R) Xeon(R) CPU E5-2609 v3 @ 1.90GHz.

Mininet and OpenDaylight controller were installed in separate virtual machines by using VMware ESXi version 6 Hypervisor.

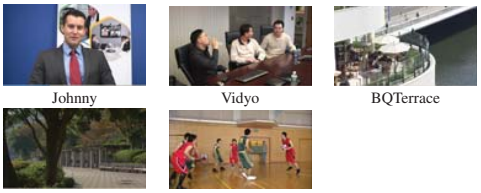
A Python based application that interfaces with the controller via the RESTful API Web services was developed. The application collects virtual switch port statistics and issues instructions on how to modify network flows facing network congestion due to queuing delay and directing it to a path with low congestion.

The bandwidth of network links connecting all virtual switches (S1, S2, S3 and S4) is set to 10Gbps and bandwidth connecting virtual hosts and virtual switches is set to 1Gbps.

YouTube video streaming is performed via the virtual host H1 to host H4. Host H4 can be reached by five paths, either S1→S4, S1→S2→S4, S1→S3→S4, S1→S3→S2→S4 and S1→S2→S3→S4. The remaining hosts H2 and H3 are used to generate TCP background traffic through virtual switches S2 and S3, respectively.

B. Video Sequences

TABLE I
Encoded Video Sequences



H264 encoded video sequences used were categorized into slow, medium and fast movements in order to evaluate the impact of modifying network flows on the quality of video streaming of different content types. The Johnny sequence was categorized as slow movement because the man is talking while seated and his upper body parts move slowly when talking. The Vidyo sequence was categorized as medium movement because the men are talking while seated and their upper body parts move gently while talking. The BQTerrace and ParkScene video sequences are categorized as fast movement videos because motorists and cyclists are moving in fast motion and the camera was also in motion. The BasketballDrive video sequence is also categorized as fast movement due to rapid movements of the basketball players. Table I depicts the thumbnails of each encoded video sequence used in the experiment. Video sequences bitrates and frame rates are listed in Table II. Each video sequence was 2 minutes long.

TABLE II
Video Sequences Parameters

Video Sequence	Bit rate (Kbps)	Frame rate (fps)	Resolution
Basketball	3547	30	720p
BQTerrace	3114	30	720p
Park Scene	3023	30	720p
Johnny	870	30	720p
Vidyo	890	30	720p

YouTube was used as a DASH server and its API player was deployed into an Apache 2 Web server in one (H1) of the SDN virtual hosts. Another virtual host (H4) was used as a Web-based YouTube client. All encoded videos were uploaded into YouTube and they were re-encoded into three quality levels (720p, 480p, 360p). The YouTube API player has

been used to collect relevant information such as the buffering duration, the number of stalls, the length and time of stalls. These parameters will be used to evaluate and compare the performance of the system when DASH is used with and without the proposed scheme.

IV. Video Quality Management Scheme

The network flow switching scheme is triggered by the virtual port queue congestion which can occur in any of the virtual ports connecting the virtual switches and hosts. The scheme utilizes the following parameters.

- the average arrival rate λ_{ij} of video packets into port i of switch j
- the average service rate μ_{ij} of video packets at port i of switch j
- the queue size q_{ij} of port i of switch j
- the video packet size distribution p_{ij} arriving at port i of switch j
- the average packet loss rate l_{ij} at port i of switch j

Via the controller, the average arrival and service rates into a virtual port is computed by periodically accessing JavaScript Object Notation (JSON) values of packets and bytes received and transmitted on each port. The maximum queue size value is manually set at the beginning of the experiment.

The average packet loss rate is calculated by periodically retrieving JSON values of packet transmit-errors.

The main goal of the scheme is to avoid the queuing overflow which will cause packet loss and hence, adversely affect the quality of video streaming from one host to another.

To track the queue size, the traffic intensity is defined as below.

$$\rho_{ij} = \lambda_{ij}/\mu_{ij} \tag{1}$$

where ρ_{ij} is the traffic intensity at port $i = 1...n$ of switch $j = 1...m$, the traffic intensity can also be described as the port utilization. If $\lambda_{ij} \geq \mu_{ij}$, then the queue will overflow, the goal is to keep $\lambda_{ij} < \mu_{ij}$. In this scheme, the traffic intensity is categorized into three levels: Low, Medium and High. The network flow which is in High level of traffic intensity will be modified and redirected to another path with Low or Medium traffic intensity if its threshold $(T_{ij} \leq \rho_{ij})$ has been reached (c.f., Fig. 3). The traffic under investigation is the video streaming, the rest of the traffic is considered as the background traffic. The traffic intensity levels are computed based on the background traffic. When the background traffic is higher, the available bandwidth for video streaming traffic will be lower.

The collection of these statistics is possible because the SDN architecture through the controller has the global view and information of all the virtual switch port statistics. If the destination path has high traffic intensity too, then the redirection will not take place. In this case, other techniques such as bitrate adaptation scheme can be carried out, however, the bitrate adaptation is out of the scope of this paper.

If a decision is reached to update the flow based on the traffic intensity, the SDN controller is instructed by the POST method to update the flow that is congested. A flow is updated when its identifier is not changed. The flow identifier is made up of: cookie, table id, priority and match.

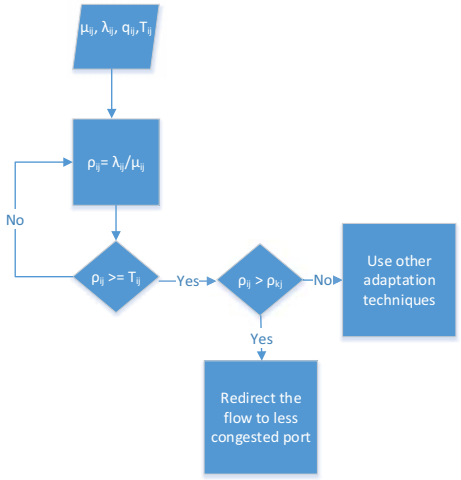


Fig. 3. Network flow switching

V. Results and Discussion

Extensive experiments have been performed under various maximum queue sizes and on different video sequences for three quality levels (720p, 480p, 360p). The average arrival rates are depicted in Fig. 4 for each quality level. As expected, the average arrival rate is higher for fast movement video sequences (BasketballDrive, BQterrace and Parkscene) for each quality level than medium (Vidyo) and slow (Johnny) movements sequences. The average service rate is approximately the same because the average Ethernet frame size for each sequence was almost the same at 1365 bytes.

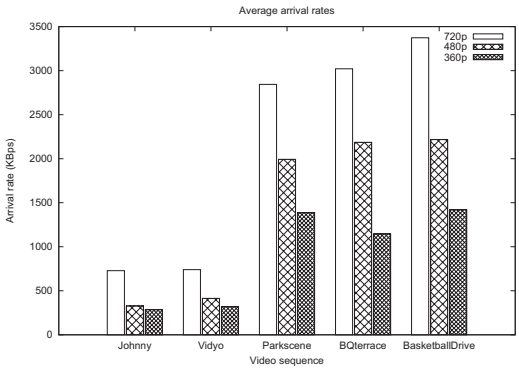


Fig. 4. Average arrival rates

A. DASH without the Proposed Quality Management Scheme

The maximum queue size was set at 400 number packets and the bandwidth was fixed at 100 Mbps to demonstrate

the behavior of DASH without the proposed video quality management scheme.

For high level of traffic intensity ($\rho_{ij} \geq 0.75$), Fig. 5 depicts the number of stalls experienced by each video sequence at each quality level. If the YouTube client buffer is empty due to bandwidth fluctuation, the re-buffering will start, this operation causes stalling. Stalling plays an important role in the evaluation of video streaming quality.

As expected more number of stalls were experienced for fast movement video sequences at 720p quality level (BasketballDrive = 15, BQTerrace = 12 and Parkscene = 13) than for medium (Vidyo = 4) and slow (Johnny = 2) movement sequences. Similar trend can be observed at 480p and 360p quality levels. This is because the fast movement sequences occupy more bandwidth than medium and slow movements videos due to high bitrates.

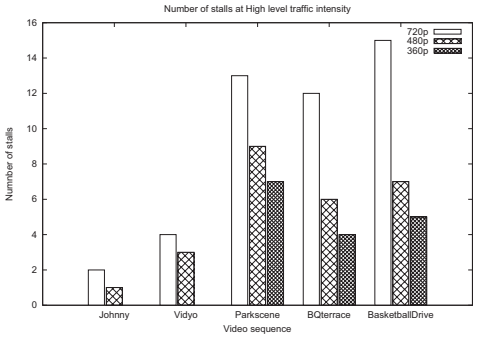


Fig. 5. Number of stalls for High level traffic intensity

The average stalling length in seconds is depicted in Fig. 6 for High level of traffic intensity for all video sequences at each quality level. Due to high bandwidth consumption, fast movement video sequences experienced longer stalling time (BasketballDrive = 23 seconds, BQTerrace = 17 seconds and Parkscene = 20 seconds) than for medium (Vidyo = 9 seconds) and slow (Johnny = 5 seconds) video sequences. Similar trend can be seen for 480p and 360p quality levels.

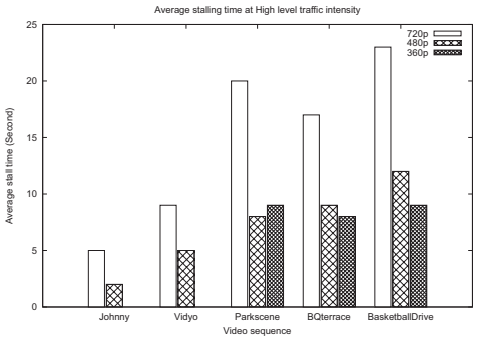


Fig. 6. Average stalling length for High level traffic intensity

For the Medium level of traffic intensity ($0.6 \leq \rho_{ij} < 0.75$), Fig. 7 illustrates the number of stalls experienced by each video sequence at each quality level. For the Medium level of traffic intensity, the number of stalls were less than that of the High level of traffic intensity, this is because the available bandwidth is bigger than that of the High level traffic intensity. There were no video stalls reported for medium and slow movement video sequences.

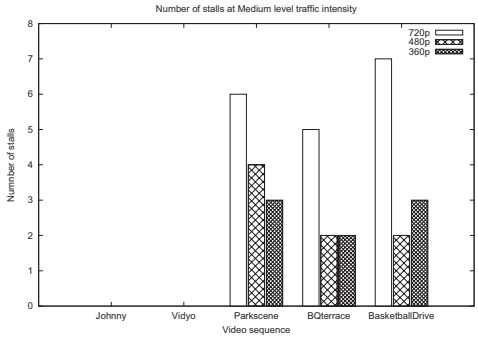


Fig. 7. Number of stalls for Medium level traffic intensity

For the Low level of traffic intensity, no stalls were experienced in all video sequences for all levels of quality.

B. DASH with the Proposed Quality Management Scheme

By deploying traffic intensity to trigger the network flow switching, the traffic intensity threshold of 0.75 was empirically selected. This threshold value performed better than other thresholds under the same maximum queue size of 400 packets, the bandwidth of 100 Mbps and the selected video sequences.

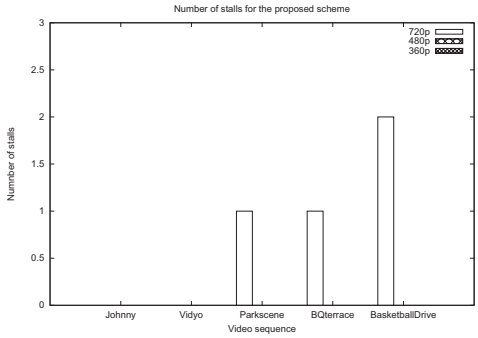


Fig. 8. Number of stalls in the proposed scheme

Fig. 8 depicts the number of stalls after the current network flow with High traffic intensity was redirected to a flow with the Low level of traffic intensity. The number of stalls were only reported for fast movement video sequences (BasketballDrive = 2, BQTerrace = 1 and Parkscene = 1), and only at 720p quality level. The results show that the number of stalls in the proposed video quality management

scheme is significantly smaller compared to DASH without the proposed scheme. No stalls were reported once the DASH video streaming was in the new path with the Low level of traffic intensity. For demonstration purposes, only High level traffic intensity was used to compare DASH with and without the proposed scheme.

Table III compares the number of stalls for DASH with and without the proposed scheme. The results show that the number of stalls have been significantly reduced by more than 84% at High level of traffic intensity for all quality levels.

TABLE III
COMPARISON: DASH WITH AND WITHOUT THE PROPOSED SCHEME

	Without the scheme			With the scheme			Reduced (%)		
	720p	480p	360p	720p	480p	360p	720p	480p	360p
Basketball	15	7	5	1	0	0	93.33	100.00	100.00
BQTerrace	12	6	4	1	0	0	91.67	100.00	100.00
Parkscene	13	9	7	2	0	0	84.62	100.00	100.00
Vidyo	4	3	0	0	0	0	100.00	100.00	NA
Johnny	2	1	0	0	0	0	100.00	100.00	NA

The stall duration also plays a major role in the evaluation of video quality over DASH, the longer the stalls, the more annoyed the viewers will be. To this end, the average stall lengths in seconds for DASH with and without the proposed scheme are also compared in Table IV. The results have shown that the average stall length has also been significantly reduced by more than 94% at High level of traffic intensity for all quality levels.

TABLE IV
COMPARISON: DASH WITH AND WITHOUT THE PROPOSED SCHEME

	Without the scheme			With the scheme			Reduced (%)		
	720p	480p	360p	720p	480p	360p	720p	480p	360p
Basketball	23	12	9	1	0	0	95.65	100.00	100.00
BQTerrace	17	9	8	1	0	0	94.11	100.00	100.00
Parkscene	20	8	9	1	0	0	95.00	100.00	100.00
Vidyo	9	5	0	0	0	0	100.00	100.00	NA
Johnny	5	2	0	0	0	0	100.00	100.00	NA

VI. CONCLUSION AND FUTURE WORK

This paper has proposed the video quality management scheme based on the traffic intensity over the SDN architecture. The experimental results based on the Mininet network emulator and the OpenDaylight controller have shown to significantly reduce the number of stalls due to bandwidth fluctuations. This was achieved by switching the network flows from High to Low level of the traffic intensity. The number of stalls and their duration in the DASH paradigm play an import role in defining the QoE in video streaming over the Internet. The more the number of stalls and the longer the stalls duration during video streaming, the more the end user is getting annoyed.

This scheme was implemented as a Northbound network application in Python utilizing RESTful API Web services provided by the OpenDaylight controller. The proposed scheme has shown to be effective in mitigating the disadvantages of DASH technique attributed to disturbing and frequent bitrates and resolution switches under bursty background traffic. For the High level of traffic intensity at 720p quality level, the number of stalls and their duration were significantly reduced by more than 84% and 94%, respectively. There were no stalls occurrences at 480p and 360p quality levels.

Future work will include the performance analysis, modelling and evaluation of Northbound network application response time and its impact on the video quality. It is envisaged that the response time could vary under different loads on the controller and forwarding devices and hence, could result into delays and eventually packet losses.

ACKNOWLEDGMENT

The work presented in this paper is partially funded by the European Union in the context of Horizon2020 Research and Innovation Programme under Marie Sklodowska-Curie Innovative Training Networks (MSCA-ITN-2014-ETN), Grant Agreement No.643072, Network QoE-NET.

REFERENCES

[1] V. N. I. Cisco, "Global mobile traffic forecast update," 2016.

[2] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[3] P. Qin, B. Dai, B. Huang, and G. Xu, "Bandwidth-aware scheduling with sdn in hadoop: A new trend for big data," *IEEE Systems Journal*, vol. PP, no. 99, pp. 1–8, 2015.

[4] K. Brunnström, S. A. Beker, K. De Moor, A. Dooms, S. Egger, M.-N. Garcia, T. Hossfeld, S. Jumisko-Pyykkö, C. Keimel, M.-C. Larabi *et al.*, "Qualinet white paper on definitions of quality of experience," 2013.

[5] T. Hoßfeld, M. Seufert, C. Sieber, T. Zinner, and P. Tran-Gia, "Identifying qoe optimal adaptation of http adaptive streaming based on subjective studies," *Computer Networks*, vol. 81, pp. 320–332, 2015.

[6] T.-Y. Huang, N. Handigol, B. Heller, N. McKeown, and R. Johari, "Confused, timid, and unstable: picking a video streaming rate is hard," in *Proceedings of the 2012 ACM conference on Internet measurement conference*. ACM, 2012, pp. 225–238.

[7] K. Tanabe, S. Miyata, K. i. Baba, and K. Yamaoka, "Threshold configuration of emergency trunk reservation considering traffic intensity for accepting more general telephone calls," in *Reliable Networks Design and Modeling, 2014 6th Int. Workshop*, Nov 2014, pp. 165–170.

[8] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: rapid prototyping for software-defined networks," in *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, 2010, p. 19.

[9] J. Medved, R. Varga, A. Tkacik, and K. Gray, "Opendaylight: Towards a model-driven sdn controller architecture," in *Proceeding of IEEE Int. Symp. on a World of Wireless, Mobile and Multimedia Networks*, 2014.

[10] M. Jarschel, S. Oechsner, D. Schlosser, R. Pries, S. Goll, and P. Tran-Gia, "Modeling and performance evaluation of an openflow architecture," in *Proceedings of the 23rd international teletraffic congress*. International Teletraffic Congress, 2011, pp. 1–7.

[11] J. Ansell, W. K. G. Seah, B. Ng, and S. Marshall, "Making queueing theory more palatable to sdn/openflow-based network practitioners," in *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, April 2016, pp. 1119–1124.

[12] J. Hu, C. Lin, X. Li, and J. Huang, "Scalability of control planes for software defined networks: Modeling and evaluation," in *IEEE 22nd International Symposium of Quality of Service*, May 2014, pp. 147–152.

[13] E. Liotou, G. Tseliou, K. Samdanis, D. Tsolkas, F. Adelantado, and C. Verikoukis, "An sdn qoe-service for dynamically enhancing the performance of ott applications," in *Quality of Multimedia Experience (QoMEX), 2015 Seventh International Workshop on*, May 2015, pp. 1–2.

[14] S. Ramakrishnan, X. Zhu, F. Chan, and K. Kambhatla, "Sdn based qoe optimization for http-based adaptive video streaming," in *2015 IEEE International Symposium on Multimedia (ISM)*, Dec 2015, pp. 120–123.

[15] J. W. Kleinrouweler, S. Cabrero, and P. Cesar, "Delivering stable high-quality video: An sdn architecture with dash assisting network elements," in *Proceedings of the 7th International Conference on Multimedia Systems*, ser. MMSys '16. New York, NY, USA: ACM, 2016, pp. 4:1–4:10.