Regular Article

Performance of DASH over Multipath TCP

Trung Thuan Nguyen[1]**, Duy Khanh Vu**[2]**, Chau Le**[1]**, Duy Quang Nguyen**[1]**, Phuong Luu Vo**[1]

[1] International University - VNUHCM, Vietnam
[2] Posts & Telecoms Institute of Technology

Correspondence: Phuong Luu Vo, vtlphuong@hcmiu.edu.vn

Communication: received 14 November 2018, revised 5 April 2019, accepted 6 April 2019
Online publication: 30 June 2019, Digital Object Identifier: 10.21553/rev-jec.216
The associate editor coordinating the review of this article and recommending it for publication was Dr. Nguyen Thi Mai Trang.

Abstract– **Recently, dynamic adaptive streaming over HTTP (DASH) is a dominated traffic in Internet. The client requests a suitable representation depending on the current network condition. On the other hand, multipath transmission control protocols emerges as potential data transmission utilizing multiple network paths concurrently. In this paper, we conduct extensively experiments to evaluate the performance of DASH over MPTCP. Four different performance metrics are investigated, *i.e.*, time on high quality, impactful switches, switch frequency, and average bitrate. The results show that the performance of DASH decreases when the paths of MPTCP have different bandwidths.**

Keywords– **MPTCP, DASH.**

1 Introduction

In recent years, the world has witnessed the rapid growth of global internet traffic. According to Cisco Visual Networking Index annual report [1], video traffic occupies a large share of internet traffic today. By 2015, video traffic consumes 70 percent of all IP traffic and this number will reach 82 percent by 2020. Today, dynamic adaptive streaming over HTTP (DASH) is a major technique to stream a video from a server to a client.

On the other hand, also according Cisco's VNI report, the number of mobile broadband subscribers grows at a double-digit rate. By 2017, the number of mobile-broadband subscriptions reaches 4.3 billion, more than 50% of the population [1]. The user equipments (UE) are usually equipped with multiple network interfaces such as 3G/4G and wi-fi. Utilizing multiple paths to transmit data achieves a high throughput, provides a smooth hand-off, and improves the high-availability. The multipath transmission control protocol (MPTCP) couples the congestion windows of the subfows from one source. It is designed to satisfy three design goals, *i.e.*, 1) improving the throughput, 2) balancing the congestion between the subfows, and 3) being friendly to the single-path TCP (*a.k.a.* do-not-harm) [2–6].

In this paper, we evaluate the performance of DASH over a MPTCP connection. Various performance metrics are evaluated such as time on high quality, number of impactful switches, number of switches, buffer level, and average bitrate. A test-bed is set up and extensively experiments are conducted.

There are several works has evaluated the performance of DASH over MPTCP. The work [7] evaluates the performance of different congestion control algorithms when transmitting DASH. The work [8] also evaluates the performance of DASH over MPTCP. However, only two performance metrics are measured in [8]. Our work is similar to the work [9]. However, the work [9] evaluates the benefit of the secondary path when fixing the primary path whereas our work evaluate the performance of the video when varying the bandwidth ratio between the paths, but the sum of bandwidth is unchanged.

The structure of the paper is as follows. Section 2 contains an overview of adaptive HTTP streaming and performance metrics. Section 3 describes the test-bed and experiment results. The conclusions are given in Section 4.

2 Dynamic Adaptive HTTP Streaming and Performance Metrics

In DASH, a multimedia content is delivered from server to client using an adaptive bitrate, client-based control protocol due to the instability of the network connection from media server to user [10–12]. In DASH, a video content is chunked into equal-playback-time segments. Furthermore, a video is encoded in multiple representations with different bitrates and resolutions. A better-performance representation has a higher bitrate and resolution, but results to more latency and consumes more bandwidth. Table I describes the bitrates and resolutions of 20 representations of "Big Buck Bunny" video [13].

The segments are downloaded in chronological order with a suitable representation based on the estimated throughput or buffer level. The downloaded segments are buffered in the client memory before being played. Video's buffer level is total playing time of the wait-to-

Table I
Big Buck Bunny Datasets

Index	1-sec.		2-sec.		6-sec.		10-sec.	
	Bitrate (kbps)	**Resolution**	**Bitrate (kbps)**	**Resolution**	**Bitrate (kbps)**	**Resolution**	**Bitrate (kbps)**	**Resolution**
1	47	320×240	46	320×240	46	320×240	45	320×240
2	92	320×240	89	320×240	89	320×240	88	320×240
3	135	320×240	131	320×240	128	320×240	127	320×240
4	182	480×360	178	480×360	177	480×360	177	480×360
5	226	480×360	222	480×360	218	480×360	217	480×360
6	270	480×360	263	480×360	255	480×360	253	480×360
7	353	480×360	334	480×360	321	480×360	317	480×360
8	425	480×360	396	480×360	374	480×360	369	480×360
9	538	854×480	522	854×480	506	854×480	503	854×480
10	621	854×480	595	854×480	573	854×480	569	854×480
11	808	1280×720	791	1280×720	780	1280×720	771	1280×720
12	1100	1280×720	1000	1280×720	1000	1280×720	987	1280×720
13	1300	1280×720	1200	1280×720	1200	1280×720	1200	1280×720
14	1700	1280×720	1500	1280×720	1500	1280×720	1400	1280×720
15	2200	1920×1080	2100	1920×1080	2100	1920×1080	2100	1920×1080
16	2600	1920×1080	2500	1920×1080	2400	1920×1080	2400	1920×1080
17	3300	1920×1080	3100	1920×1080	3100	1920×1080	2900	1920×1080
18	3800	1920×1080	3500	1920×1080	3300	1920×1080	3200	1920×1080
19	4200	1920×1080	3800	1920×1080	3600	1920×1080	3500	1920×1080
20	4700	1920×1080	4200	1920×1080	3900	1920×1080	3800	1920×1080

be-played segments. When a new segment is successful downloaded, the buffer increases by the segment length and when a segment is played, the buffer decreases by the segment length. In DASH, video player requires an *initial buffer level*, *e.g.*, one segment, before start playing. When the buffer exceeds a *target buffer level*, client will stop downloading the new segment, wait for the buffer level decreasing below the target level and then resume the downloading. The client will *re-buffer* when the buffer level is under a defined threshold. The player will stop playing and wait until the buffer level is over the initial threshold. Re-buffering causes video freezes and reduces the quality-of-experience (QoE) of the users [14, 15].

Due to the fluctuation in network environment, the client-based adaptation algorithm plays an essential role in providing a high QoE for the users. Depending on the estimated throughput (for throughput-based methods) or the buffer level (for buffer-based methods) or a combination of these two parameters, the client decides a suitable representation to request for the new segment.

2.1 Performance Metrics

The most correct method to measure the quality of video is using subjective method mean-opinion-score (MOS). The video is played, and the users are asked about the quality of the playing video. The quality of the video is rated using a one-to-five scale. However, a subjective measurement is costly and time consuming. In some recent studies [16–19], the authors have shown that there are several factors that affect the MOS such as the video time on high quality, number of bitrate and resolution switches, and average bitrate of the video session.

2.1.1 Time on high quality:

According to [17], the time on high quality is a main influenced factor since its effect in performance is very visible, user's MOS possibly decreases up to 37.5 percent (from 4 MOS to 2.5 MOS) when this value decreases. In our experiments, the highest resolution of the video played at the client is 1280 × 720. The time on high quality is the percentage of time the player playing at this resolution..

2.1.2 Switching amplitude:

Not every representation switch causes the same effect on MOS. In [17], the switching amplitude represents how different in bitrate or resolution between representations when the switching event happens. If the video quality oscillates between the highest resolution and the medium resolution (next lower one), it is considered as a low amplitude switch. If the switching

Table II
Impactful and Non-Impactful Switches

Switch	Description
Non-impactful	switching between bitrates within a same resolution (*e.g.*, from 1500 kbps to 1200 kbps
Impactful	switching to a different resolution (*e.g.*, from 1280 × 720 to 854 × 480)

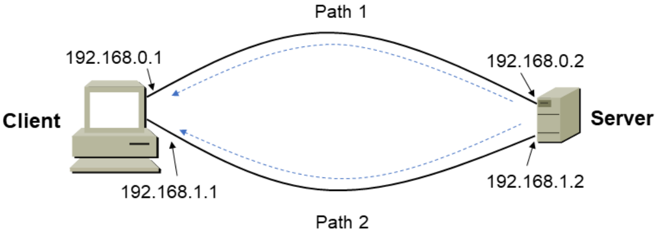


Figure 1. Network topology.

is between the highest and the lowest quality, it is considered a higher quality switch. However, they encoded their test sample into only 3 quality levels resolution: 640 × 360, 320 × 180, 160 × 90 which is different to our experiment sample (4 types of resolution and 14 different bitrate value).

Influenced by the works [16, 17], the switch amplitudes are classified in 2 types in this paper, *non-impactful switch* and *impactful switch* as described in Table II. A switch is considered an impactful one if it changes the resolution of the playing video.

2.1.3 Switch frequency:

Switching frequency is a rate of video quality switch over a specific time unit during playing session. However, the work [16] shows that switching frequency does not have a significant effect in user's MOS. In our experiment, if the representation switch is a non-impactful switch (within the same resolution), the change in video quality seems undetectable.

2.1.4 Average bitrate:

Average bitrate is the mean of video bitrate over the playing session. Theoretically, a higher average bitrate yields a better video quality and higher bandwidth utilization, but average bitrate is not a QoE's influenced factor since videos with similar average bitrates can result to different subjective perceptions, *i.e.*, a 14-second video played with at 1157 kbps is perceived differently from the same video played at 1500 kbps for first 10 second but drop down to 300 kbps in last 4 seconds. The work [19] shows that average bitrate cannot be used alone to evaluate performance of DASH.

3 Performance Evaluation

3.1 Experiments Setup

We setup a test-bed to evaluate the performance of DASH over Multipath TCP. Our test-bed includes two virtual machines running Linux Kernel MultiPath TCP [20] (see Figure 1). One machine installed Apache

Table III
Bandwidth Setting of Experiment Scenarios

Bandwidth ratios	Subflow 1 (kbps)	Subflow 2 (kbps)
1:1	1000	1000
1:2	667	1333
1:3	500	1500
1:4	400	1600
1:5	333	1667

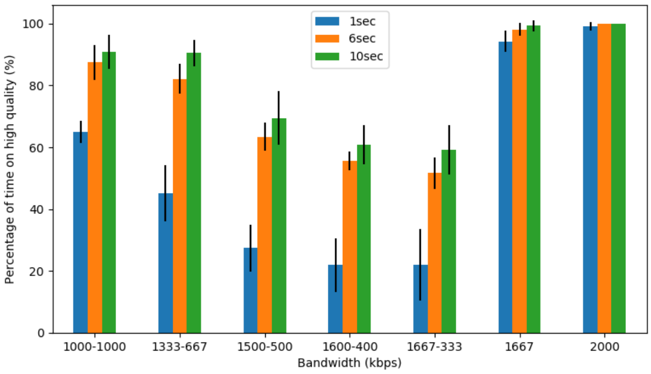


Figure 2. Percentage of time on high quality.

HTTP server [21] run as the server which holds all video segments. The other installed DASH.js [15] run as the client downloading the video segments from the server and playing them. We use Linux traffic control utility [22] to change bandwidth, packet delay and loss of the paths to emulate various network conditions.

Total bandwidth of two subflows is 2000 kbps, random packet loss is set 0.5 percent and propagation delay is 30 ms for each subflow. We investigate the performance of video in several scenarios described in Table III. The bandwidth ratio between two subflows is gradually decreased from 1:1 to 1:5.

The sample video "Big Buck Bunny" is used [13]. The video length is about 580-second. We evaluate three types of segment lengths: 1-, 6-, and 10-second segment. The video is encoded in 20 levels of bitrates and 5 resolutions as described in Table I. Each experiment is run in 10 times. Average values used in the plots are calculated from 60-th to 540-th second to avoid the initial phase. The errors are mean-square errors. Buffer-based adaptation is applied for all the experiments.

3.2 Experiment Results

Our experiment evaluates four metrics when the bandwidth ratio between two subflows vary from 1:1 to 1:5.

3.2.1 Time on high quality:

Figure 2 shows the percentage of time the video playing at high quality, *i.e.*, resolution 1280 × 720 or better. With 1-second segment length, the time on high quality decreases more than 42.9 percent when the bandwidth ratio varies from 1:1 to 1:5 (from 64.9 percent of time at ratio 1:1 to 22.0 percent of time at

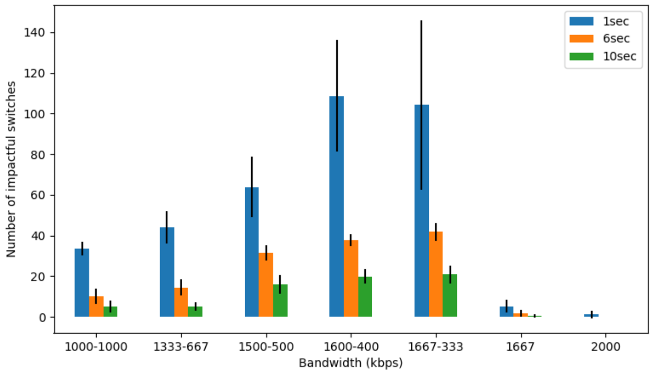


Figure 3. Number of impactful switches.

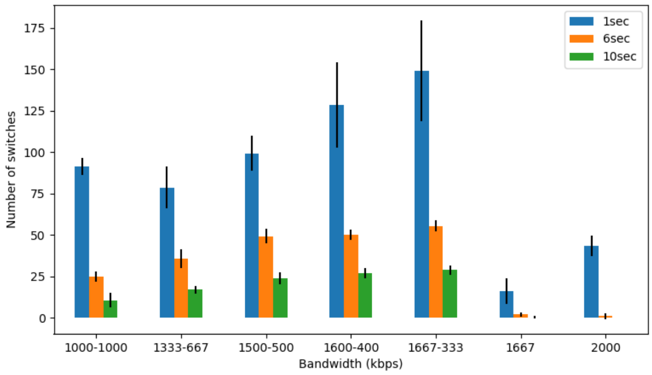


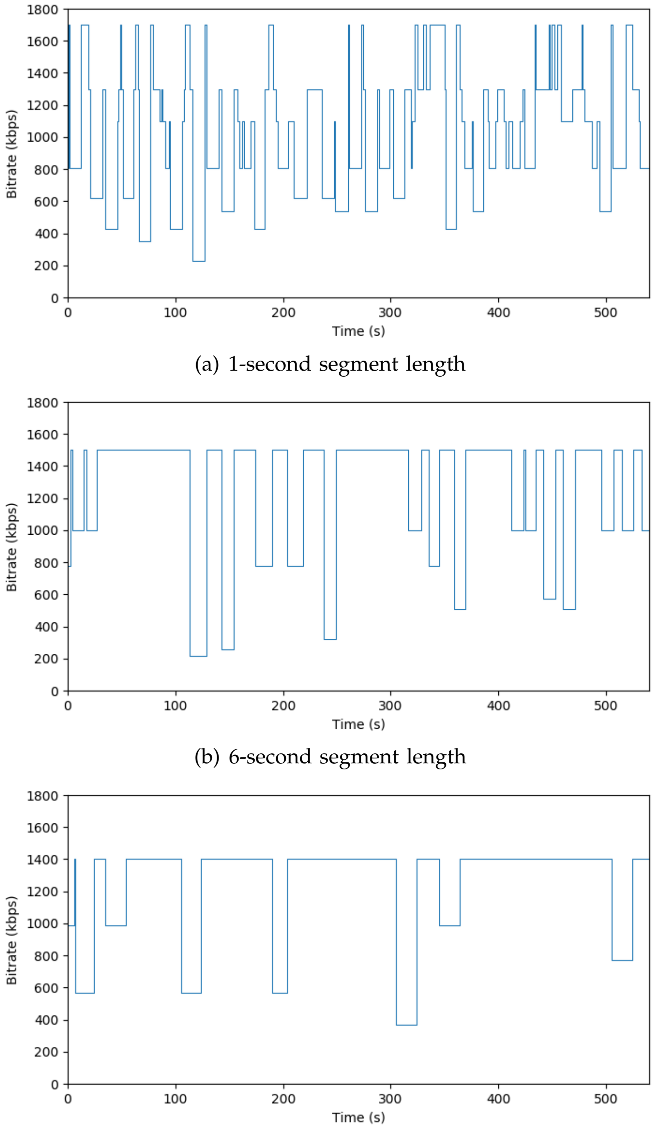
Figure 4. Number of switches.

bandwidth ratio 1:5). With 10-second segment length, the reduction is 31.5 percent, smaller than the one of 1-second segment (from 90.7 percent at ratio 1:1 to 59.2 percent at ratio 1:5). The more difference of the bandwidths between two subflows yields a less percentage of time the video plays at high quality. In addition, this metric for longer segment length yields a better outcome than the one for a shorter one.

3.2.2 *Bitrate and resolution switches:*

We next analyze the number of impactful switches (resolution switches) and the number of switches (bitrate switches). A higher number of impactful switches and number of switches yields a lower QoE.

Figures 3 and 4 show the average number of the impactful switches as well as the number of switches when varying the bandwidth ratio between two paths with different segment duration. In general, the number of impactful switches increases when the bandwidth ratio decreases (see Figure 3). For example, with 10-second segment, the average number of impactful switches is 5 with bandwidth ratio 1:1 and 20.8 with the ratio 1:5. The performance of 1-second segment is the worst in all test cases. The average numbers of impactful switches of 10-second segment in both figures are always the lowest of all the segment lengths. For example, with the case 1000-1000 kbps of two paths, 10-second segment has 5 impactful switches, whereae 1- and 6- segment lengths have 33.5 and 9.9 impactful switches in average, respectively. The average number of switches of 1-second segment is much higher than the one of 6- and 10-second segment lengths in all the cases (see Figure 4). This metric tends to increase as the two paths are more different in bandwidths.



(a) 1-second segment length

(b) 6-second segment length

(c) 10-second segment length

Figure 5. Video bitrates with 1000-1000 kbps of two paths.

The performance metrics of 1-second segment video are less stable than the ones of other segment lengths. the increasing in two figures is not monotonic with 1-second segment. For example, the number of impactful switches of 1-second segment with ratio 1:5 is slightly lower than the one of bandwidth ratio 1:4 (see Figure 3), or the average number of switches of bandwidth ratio 1:2 is lower than the one with the ratio 1:1 (see Figure 4). However, there is a large deviation with 1-second segment.

3.2.3 *Average bitrate:*

Figure 6 shows the average bitrate of three types of segment lengths. In overall, it also indicates a same property as previous experiments: a smaller bandwidth ratio yields a worse performance.

3.2.4 *Buffer levels:*

We observe the buffer level of the player with different segment lengths. In previous experiments, 1-second segment length has a noticeable reduction in

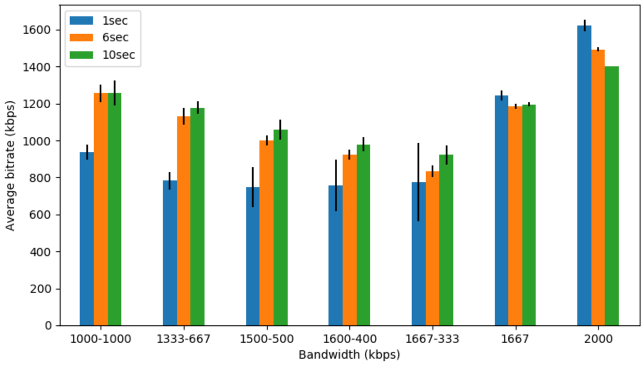


Figure 6. Average bitrate.

Table IV
Video Latencies with 1000-1000 Kbps of Two Paths

Segment length	Latency (s)
1-second	0.8 ± 0.42
6-second	3.7 ± 1.16
10-second	6.6 ± 1.17

performance. However, a shorter segment length yields a more stable buffer level as observed in Figure 7, the latency from sending the request to playing the first segment of a shorter segment is also shorter (see Table IV). 1-second segment length has almost instant start time of video.

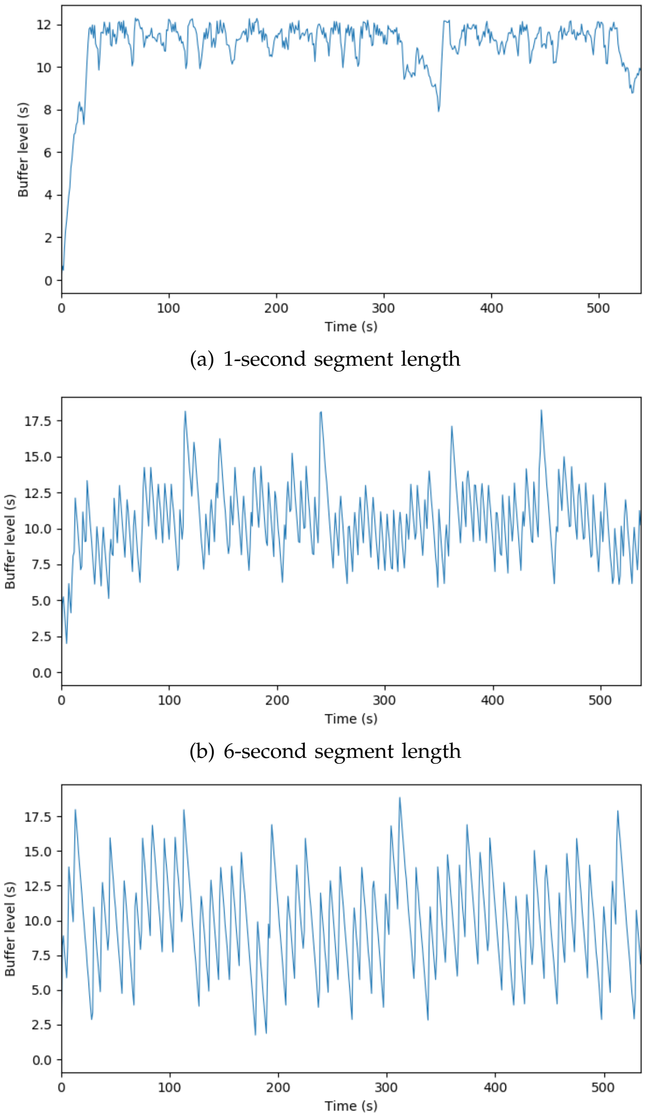
3.3 MPTCP vs. TCP

We conduct the experiments to compare the performance of DASH between TCP and MPTCP. In case the bandwidth of the TCP is 2000 kbps, which equals to the total bandwidth of two MPTCP's subflows, the performance of TCP always outperforms MPTCP (see Figures 2-6). When the bandwidth of TCP flow equals to the larger subflow of MPTCP (1667 kbps). The performance metrics of the MPTCP still is even worse than the one of TCP.

The reason of the reduction in performance of MPCTP shown in the above experiments is from the out-of-order of the video packets delivered over heterogeneous paths. Packet scheduling mechanism of MPTCP decides which packet is dispatched via which path. However, due to the different bandwidths of the paths, video packets may arrive out-of-order at the receiver and must be re-ordered before being played [23]. The packet scheduling of MPTCP currently is not designed specific for video streaming. It needs a cross-layer design between the application layer and transport layer for video streaming. This work is reserved for our future study.

4 Conclusions

We have evaluated the performance of DASH over a MPTCP flow. As the bandwidth difference between the subflows increases, the QoE metrics of DASH decrease due to the out-of-order packets arriving at the receiver. To improve the performance DASH over MPTCP, we should redesign packet scheduling mechanism of MPTCP to support DASH, which is reserved for our future studies.



(a) 1-second segment length

(b) 6-second segment length

(c) 10-second segment length

Figure 7. Buffer levels with 1000-1000 Kbps of two paths.

Acknowledgment

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.02-2015.36.

References

[1] Cisco, "Cisco Visual Networking Index: Forecast and Methodology, 2016-2021."

[2] D. Wischik, M. Handley, and C. Raiciu, "Control of Multipath TCP and Optimization of Multipath Routing in the Internet series," *Lecture Notes in Computer Science, R. Nez-Queija and J. Resing, Eds.*, vol. 5894, 2009, Springer Berlin/Heidelberg.

[3] R. Khalili, N. Gast, M. Popovic, and J.-Y. Le Boudec, "MPTCP is not pareto-optimal: performance issues and a possible solution," *IEEE/ACM Transactions on Networking*, vol. 21, no. 5, pp. 1651–1665, 2013.

[4] Q. Peng, A. Walid, J. Hwang, and S. H. Low, "Multipath TCP: Analysis, design, and implementation," *IEEE/ACM Transactions on Networking*, vol. 24, no. 1, pp. 596–609, 2016.

[5] O. Bonaventure, M. Handley, and C. Raiciu, "An overview of multipath TCP," *; login:*, vol. 37, no. 5, pp. 17–23, 2012.

[6] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, "TCP extensions for multipath operation with multiple addresses," RFC 6824, Tech. Rep., 2013.

[7] S. Bikas, C. Cetinkaya, and M. Sayit, "Evaluation of MPTCP congestion control for DASH," in *Proceedings of the IEEE 7th International Conference on Consumer Electronics-Berlin (ICCE-Berlin)*, Sep. 2017, pp. 279–283.

[8] L. Peschke, "Combining DASH with MPTCP for video streaming," Master's thesis, Ecole polytechnique de Louvain, 2017.

[9] C. James, E. Halepovic, M. Wang, R. Jana, and N. K. Shankaranarayanan, "Is multipath TCP (MPTCP) beneficial for video streaming over DASH?" in *Proceedings of the IEEE 24th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, 2016, pp. 331–336.

[10] T. Stockhammer, "Dynamic adaptive streaming over HTTP– standards and design principles," in *Proceedings of the second annual ACM conference on Multimedia systems*. ACM, Feb. 2011, pp. 133–144.

[11] I. Sodagar, "The MPEG-DASH standard for multimedia streaming over the Internet," *IEEE MultiMedia*, vol. 18, no. 4, pp. 62–67, 2011.

[12] ISO/IEC 23009-1:2014, "Dynamic Adaptive Streaming over HTTP (DASH)– part 1: Media Description and Segments format."

[13] S. Lederer, C. Müller, and C. Timmerer, "Dynamic adaptive streaming over HTTP dataset," in *Proceedings of the 3rd Multimedia Systems Conference*. ACM, Feb. 2012, pp. 22–24.

[14] T. C. Thang, H. T. Le, A. T. Pham, and Y. M. Ro, "An evaluation of bitrate adaptation methods for HTTP live streaming," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 4, pp. 693–705, 2014.

[15] DASH Industry forum, "Dash.js." [Online]. Available: http://cdn.dashjs.org, Accessed: Mar. 18, 2019

[16] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hoßfeld, and P. Tran-Gia, "A survey on quality of experience of HTTP adaptive streaming," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 469–492, 2015.

[17] A. K. Moorthy, L. K. Choi, A. C. Bovik, and G. De Veciana, "Video quality assessment on mobile devices: Subjective, behavioral and objective studies," *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 6, pp. 652–671, 2012.

[18] B. Lewcio, B. Belmudez, A. Mehmood, M. Wältermann, and S. Möller, "Video quality in next generation mobile networks – perception of time-varying transmission," in *Proceedings of the IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR)*, 2011, pp. 1–6.

[19] C. Timmerer, M. Maiero, B. Rainer, S. Petscharing, D. Weinberger, C. Mueller, and S. Lederer, "Quality of experience of adaptive HTTP streaming in real-world environments," *IEEE Comsoc MTC E-Letter*, vol. 10, no. 3, 2015.

[20] ICTeam, "MultiPath TCP - Linux Kernel implementation." [Online]. Available: https://multipath-tcp.org, Accessed: Mar. 18, 2019

[21] Apache, "Apache http server project." [Online]. Available: http://httpd.apache.org, Accessed: Mar. 18, 2019

[22] Traffic Control HOWTO. [Online]. Available: http://tldp.org/HOWTO/Traffic-Control-HOWTO/index.html

[23] T.-A. Le and L. X. Bui, "Forward delay-based packet scheduling algorithm for multipath TCP," *Mobile Networks and Applications*, vol. 23, no. 1, pp. 4–12, 2018.

Trung Thuan Nguyen was a student of International University - Vietnam National University Ho Chi Minh City and received his degree in information technology in 2016. Currently, he is a software engineer at Bosch Engineering and Solution Vietnam. He is interest on Data analysis, IoT applications and machine learning.

Duy Khanh Vu received a bachelor of information technology at SAIGON University, Vietnam in 2014, and the master's degree Information System at Posts and Telecommunications Institute of Technology, Vietnam in 2016.

Currently, his job is the system engineer for glassegg company. His research interests includes networking and system security.

Chau Le is an undergraduate student at International University, Vietnam National University Ho Chi Minh City and expects to receive her Bachelor degree in Computer Science by 2020. Her current interests are in the fields of software engineering, machine learning and data science.

Duy Quang Nguyen is a batch 2020 undergraduate student who majors in Computer Science at International University - Vietnam National University Ho Chi Minh City. His current interests includes machine learning, artificial intelligence and other related topics.

Phuong Luu Vo received her B.Eng and M.Eng degrees in electrical-electronics engineering from Ho Chi Minh City University of Technology, Vietnam in 1998, 2002, respectively and the Ph.D. degree at Kyung Hee University, Korea in 2014. Currently, she is a lecturer of School of Computer Science and Engineering at International University - Vietnam National University Ho Chi Minh City. Her research interests includes applying optimization and game theories to network.