Performance Evaluation of Edge Computing Assisted Adaptive Streaming Algorithms

Waqas ur Rahman, Choong Seon Hong and Eui-nam Huh
Dept. of Computer Science and Engineering
Kyung Hee University
Yongin, South Korea
waqasrahman@khu.ac.kr, cshong@khu.ac.kr, johnhuh@khu.ac.kr

***Abstract*— Multi-Access Edge Computing (MEC) adaptive streaming presents an opportunity to jointly optimize the quality of experience in cellular networks by moving the adaptation intelligence from the client to the edge cloud. In this paper, we investigate the performance of MEC-assisted algorithms and compare their performance with the client based adaptation logic. We conduct extensive experiments and quantify benefits and drawbacks of edge computing–assisted adaptation algorithms. The results from our experiments reveal that MEC-assisted algorithms outperforms the purely client-based heuristics in most of the video quality metrics. However, the results also show that the MEC-assisted algorithms are not able to protect the playback buffer from drying up under different network settings.**

Keywords—HTTP adaptive streaming, rate adaptation algorithm, multi-access edge computing, MEC, DASH

I. Introduction

Multimedia contents account for a majority of the traffic over the Internet. According to Cisco's Visual Networking Index, the global mobile data traffic is expected to reach 82% by 2022 [1]. The most common solution for managing the traffic demands is to use Hypertext Transport Protocol (HTTP). The most popular video streaming services such as YouTube, Netflix employ HTTP over TCP for streaming multimedia over computer networks.

In HTTP Adaptive Streaming (HAS), a video stream is encoded into multiple video rates and each encoded stream is divided into segments of fixed duration. The video content is stored at the HTTP server. The rate adaptation algorithm picks an appropriate segment according to the network conditions. The objective of the rate adaptation algorithm is to maximize the quality experience by meeting multiple conflicting video quality objectives. The potential objectives include selecting the highest feasible set of video bit rates, avoiding needless video bit rate changes, assigning equitable video rates among the competing video clients, and preserving the buffer level to avoid any interruption in the playback [2-3].

In cellular networks, video streaming can be subject to low video quality and playback rebuffering because of bandwidth limitations or unstable networks. The rate adaptation algorithms strive to select the highest feasible video rate. However, in an unstable network, it leads to higher frequency of vide bitrate switches and higher risk of buffer underflow. In [3-5], the authors observed that in the presence of the competing clients, the clients inefficiently share the bandwidth and achieve unfair video rates.

Traditionally, the rate adaptation algorithm runs at the client side [6-8]. The clients are oblivious of the bottlenecks in the radio channel and the competing clients. The decisions are made independently which may affect the performance of other competing clients. Recently, the edge computing paradigm has been proposed to provide better performance compared to cloud computing [9]. Multi-Access Edge Computing (MEC) [9] brings computation and storage capabilities to the edge of the mobile network by deploying servers within the radio access network (RAN). The MECs have real-time access to the application and RAN information. The user experience could be enriched by transferring the video quality adaptation intelligence from the client side to the edge of the cloud.

Extensive work has been done to evaluate the performance of HTTP adaptive streaming algorithms under different experimental scenarios [10][11]. However, they only focus on client side rate adaptation algorithm. The client does not have the knowledge of the other clients sharing the network. The algorithms take decisions based on the limited knowledge about the network conditions. The mobile edge computing assisted approach presents an opportunity to enhance the user experience by centrally adapting the video quality. In our work, we make the following contributions:

- We compare the performance of edge computing assisted algorithms with client side rate adaptation algorithm.
- We conduct extensive experiments by varying different client and server side parameters and find several observations on the performance of HAS algorithms.
- We quantify the benefits and drawbacks of edge computing assisted adaptation compared with the client side approaches in edge cloud scenarios. Based on the observations from experiments, we suggest several guidelines to help improve the performance of edge computing assisted algorithms.

The rest of this paper is organized as follows. Section II reviews the existing work on video streaming algorithms. Section III introduces our experimental framework and detailed operation of HAS algorithms. Section IV evaluates the performance of HAS algorithms and summarizes the key observations. Finally, Section V concludes the paper.

II. Related work

The HAS rate adaptation algorithms can be classified into three categories: (1) throughput-based, buffer-based and hybrid-based. References [12] and [13] propose rate adaptation algorithms that select video rates based only on the throughput. Many methods have been proposed to incorporate the information concerning the playback buffer for selecting the video rate [14] [15]. The algorithms divide the playback buffer into multiple predefined thresholds: B_1, B_2, B_3, B_{max} such that $B_1 < B_2 < B_3 < B_{max}$. The algorithms increase or decrease the video rate aggressively or conservatively based

This work was supported by the MSIT(Ministry of Science and ICT), Korea, under the ICT Consilience Creative program (IITP-2019-2015-0-00742) supervised by the IITP (Institute for Information and Communications Technology Planning and Evaluation). Prof. Eui-Nam Huh and Prof. Choong Seon Hong are the corresponding authors.

on whether the buffer level increases above the next higher buffer threshold or decreases below the next lower buffer threshold, respectively. The hybrid-based algorithms consider both throughput and playback buffer occupancy to select the video quality [6-8]. In [16], the authors propose algorithms that consider the size of the upcoming segments in addition to the throughput and the buffer occupancy to predict the segment download time. W. U. Rahman et al. [17] present an optimized solution for multi-access edge computing-assisted HAS by using edge cloud capabilities for jointly optimizing the rate adaptation in a cellular network. Mehrabi et al. [18] design an INLP problem to jointly optimize the QoE while fairly allocating bit rates among the clients and balancing the utilized resources among multiple edge servers.

Many studies have been conducted to analyze the performance of HAS algorithms. Seufert et al. [19] surveys the quality adaption in video streaming and explains its influence on QoE. Kua et al. [20] survey key rate adaptation algorithms and classify them based on feedback signals used to select video quality. Akashbi et al. [10] evaluate three commercial HAS players and show the effectiveness and inefficiencies of the video players. The results show that the video players do not perform consistently under different network conditions. Mueller et al. [11] evaluated multiple HAS systems including Microsoft Smooth Streaming, Apple HLS and Adobe HDS in vehicular environment. Akashbi et al. [5] evaluated the performance of HAS clients competing for the network bandwidth. The authors show that during the steady-state phase, when multiple streams compete for the network resources, the clients share the bandwidth unfairly. Stohr et al. [21] provide an evaluation framework to analyze the performance of HAS players. These works only analyze the performance of client side algorithms. Our work conducts extensive experiments to analyze both MEC-assisted rate adaptation algorithms and compare their performance with client based approaches.

III. Experimental Framework

In this section, we present the test bed and implementation details of the rate adaptation algorithms used for evaluation. For the rate adaptation problem, a video is fragmented into S segments of the duration of τ seconds.

A. Metrics

We use the following video quality metrics to analyze the performance of the algorithms

- Average video rate: The high video rate improves the user experience but increases the risk of playback interruption. The average video rate of the k^{th} client Q_k, can be obtained:

$$Q_k = \frac{\sum_{n=1}^{S} r_{ik}(n)}{S} \tag{1}$$

where r_{ik} is the i^{th} video rate assigned to the k^{th} client, n is the segment index.

- Video rate switches: Frequent bit rate switches inversely affect the QoE. The bit rate switching metric is given by:

$$QS = \frac{\sum_{n=1}^{S} r_i(n) - r_i(n-1)}{S} \tag{2}$$

- Buffer underflow: The playback interruption due to buffer underflow affects user experience the most. One long interruption is preferred to multiple short ones.
- Fairness: Multiple clients competing at the bottleneck must be able to achieve equitable video rates. We use the Jain fairness index to quantify the fairness [22]. The Jain fairness index of r_i over all the players i is given by:

$$Fairness = \frac{\left(\sum_{i=1}^{C} r_i\right)^2}{C\sum_{i-1}^{C} r_i^2}, i \geq 0 \tag{3}$$

- Inefficiency: Low inefficiency values are desirable because it would mean that the client selects the highest feasible bit rates that are lower than the actual throughput. The inefficiency at the time t is given as follows [17]:

$$Inefficiency = \frac{\left|\sum_{i=1}^{C} r_i - W\right|}{W} \tag{4}$$

B. Objectives

We conduct extensive experiments to find the key characteristics of MEC-assisted adaptation algorithms. We compare their performance with client based algorithms to point out the benefits and drawbacks of MEC-assisted algorithms. We evaluate the algorithms in different server and client settings to see how the algorithm adapt to different conditions. Finally, we recommend some guidelines to improve the performance of algorithms.

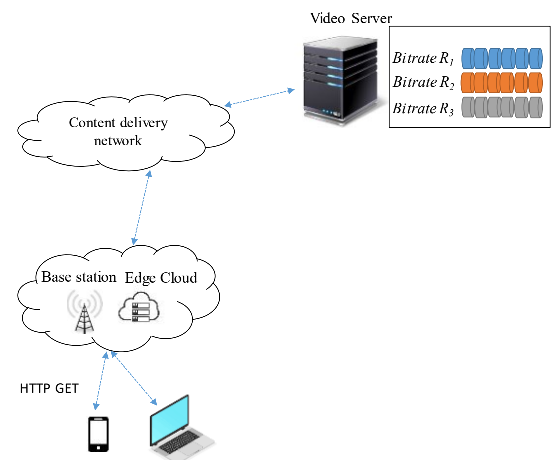


Fig. 1. Network topology

C. Experiment Setup

We implement HTTP–based adaptive video streaming in the multi-access edge computing scenario shown in Fig. 1. We implement the LTE network as the underlying cellular network. A detailed configuration of the cellular network is shown in Table I.

D. Tested HAS Algorithms

In this section, we briefly explain the operation of HAS algorithms used for experimentation and analysis. We adopt the solutions proposed in [17], [18], [6], [7] and [8] to analayze the performance of HAS algorithms. We refer to the algorithms proposed in [17], [18], [6], [7] and [8] as *MECA*, *ECAA*, *SARA, NALD* and *QLSA* respectively. In addition, we analyze the performance of algorithm that selects the highest video rate less than the available bandwidth. We refer to this algorithm as *Instant*.

TABLE I
CELLULAR NETWORK CONFIGURATION

Cell Layout	Single hexagonal cell
UE distribution	Random
Path loss model	Hata Model PCS Extension
BS transmission power	38 dBm
UE distance	1 ~ 500 m
Scheduler	Proportional fairness
UE speed	75 km/h

1) MECA

The algorithm presents a hybrid heuristic adaptation algorithm to jointly enhance multiple conflicting video quality objectives. Since the MEC is unware of the client's capabilities, therefore, the client suggests the MEC the highest video rate it can play back based on the display and the buffer level. Based on the suggestion from the client, the algorithm jointly selects the video rates of the competing clients while ensuring that video rates are fairly assigned to the clients. The algorithm selects the lowest available video rate as the first segment. To increase and decrease the video rate, the algorithm calculates *Fairness* and *Switching* indexes. The proposed algorithm considers the thresholds δ_F and δ_S of fairness and switching respectively. To increase the video rate, only the fairness condition should be satisfied. The highest video rate less than the estimated throughput that satisfies the condition, *Fairness* index > δ_F, is selected. To decrease the video rate, the selected video rate should satisfy the following two conditions: (1) $R_{prev} > r_{avg}$, (2) Switching index $\leq \delta_S$ where r_{avg} is the average of the selected video rates of the competing clients.

2) ECAA

ECAA proposes a heuristic algorithm to efficiently solve the video client to edge server mapping and video rate selection. The algorithm allocates the mobile client to the nearest base station or to the base station with the highest achievable video rate. Unlike the MECA algorithm, ECAA algorithm selects video rates independently. Once the client has been allocated to the base station, the algorithm selects the highest available video rate for the first segment. For the following segments, the algorithm picks the video rate which results in low switching level and high fairness value. The algorithm considers the thresholds δ_F and δ_S of fairness and switching respectively. The algorithm selects the video rate that results in a switching less than δ_S and fairness value greater than δ_F. If no video rate satisfies these conditions, it picks the video rate that results in a switching less than δ_S and ignores the fairness condition. If still there is no feasible video rate, it selects the most sustainable video rate. In this work, we adopt the ECAA algorithm for a single-cell scenario such that the MEC allocates the video rates to the clients.

3) SARA

The SARA algorithm considers the sizes of the upcoming segments in addition to the throughput and buffer occupancy to predict the segment download time. The algorithm divides the playback buffer into multiple predefined thresholds: I, B_α, B_β where ($I < B_\alpha < B_\beta$). When the buffer occupancy is below I, the client selects the lowest bitrate. When the buffer level is between I and B_α ($I < B_{curr} < B_\alpha$), the bitrate is incremented in single steps. As the buffer level further increases and stays between B_α and B_β ($B_\alpha < B_{curr} < B_\beta$), the client selects the most suitable video rate that is greater than or equal to the current video rate. When the buffer level increases above B_β, the most suitable video rate for current throughput is selected. The request to download the next segment is only sent when the buffer level falls to B_β. The algorithm does not dynamically adjust the buffer thresholds as the number of available video rates, segment sizes, buffer level or buffer sizes vary.

4) NALD

The NALD algorithm selects the video rate based on estimated throughput and playback buffer level. The algorithm divides the playback buffer level into multiple buffer thresholds. When the buffer occupancy is less than 20% of the buffer size, the proposed algorithm selects the lowest available video rate. To increase the video rate in response to the increase in the throughput and the buffer level, the following two conditions should be satisfied; first, the selected video rate should be less than the estimated throughput. Second, for a client to select the k^{th} video rate, the buffer level should be higher than the buffer threshold. To decrease the video rate, the algorithm stays at the current video unless the buffer level decreases below the threshold. Once the buffer level decreases below the threshold, irrespective of the estimated throughput, the algorithm selects the highest video rate such that the buffer level at the download of the next segment does not decrease below 20% of the buffer size.

5) QLSA

The QLSA algorithm has two key features: insertion of intermediate video rate and mitigating the rate switches. The algorithm first calculates $\Delta = l_{tmp} - l_{prev}$ where l_{prev} defines the quality level of the previous segment and l_{tmp} calculates the video rate less than the available bandwidth. The algorithm selects two thresholds: Th_{low} and Th_{high}. If $\Delta < -Th_{low}$, the algorithm gradually drops the video quality. If $Th_{low} < \Delta < Th_{high}$, the algorithm selects the highest video rate which does not exceed the weighted average of the available throughput. If $\Delta > Th_{high}$, the algorithm selects l_{tmp} as the next video rate.

6) Insntant

The algorithm selects the highest available video rate less than the available throughput.

IV. PERFORMANCE EVALUATION

To achieve adaptive streaming, the HTTP server offers the client 12 levels of representation to adapt the video rates. These video rates are 184, 380, 459, 693, 1270, 1545, 2000, 2530, 3750, 5379, 7861 and 11321 kbps. The users are randomly distributed within the cell. The users move at a speed of 75 kmph. The simulation runs 200 s. The segment duration and buffer size are set to 4 and 15 seconds, respectively and 10 clients compete for the bandwidth unless otherwise stated. The clients' arrival time is uniformly distributed within first 30 seconds.

A. Impact of Buffer Size

Fig. 2 shows that the MEC-assisted algorithms achieve the highest fairness values and the lowest inefficiency values. The reason is that the MEC-assisted algorithms have the knowledge of all the competing clients. Therefore, the algorithms are able to fairly assign video rates to the clients. Fig. 3 shows that for a larger buffer size, the ECAA achieves the high video rate. However, it experiences playback interruptions. When the buffer is decreased to 15 seconds, MEC-assisted algorithms achieve the highest video rate while keeping a low frequency of video rate switches. Figs. 4 and 5 shows that when buffer size is decreased, the clients experience higher number of playback interruptions. When the bandwidth varies or a new client joins the streaming session, the clients have a small buffer to keep the segments, this increases the frequency of playback interruptions.

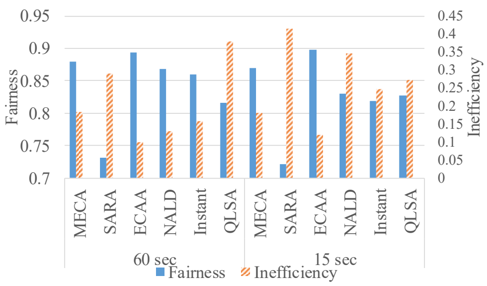


Fig. 2. Comparison of fairness and inefficiency values when buffer size is set to 60s and 15s

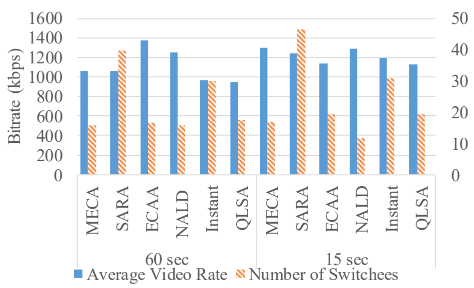


Fig. 3 Average video rates selected and average video rate switches experienced by the clients when buffer size is set to 60s and 15s

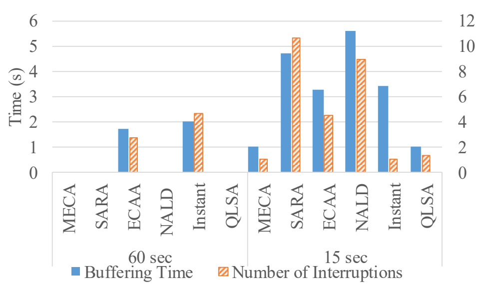


Fig. 4 Average rebuffering time and number of interruptions experienced by the clients when buffer size is set to 60s and 15s

B. Impact of Number of Clients

Fig. 6 shows that when the number of clients are decreased to 6, the fairness values increase and inefficiency values decrease. The MEC-assisted algorithms achieve the highest fairness values and they utilize the bandwidth efficiently. Fig. 7 shows that similar to previous experiments, ECAA algorithm achieves the highest video rate when the number of clients are set to 6. When the number of clients are increased to 10, the MECA and ECAA algorithm achieve the highest video rates. Furthermore, The MECA and ECAA minimize frequency of video rate switches. Figs. 8 and 9 show that the SARA and ECAA algorithms experience high number of playback interruptions. The SARA algorithm aggressively increases the video rates based on current throughput and playback buffer. As the throughput or number of clients vary, the SARA algorithm is unable to protect the buffer from draining.

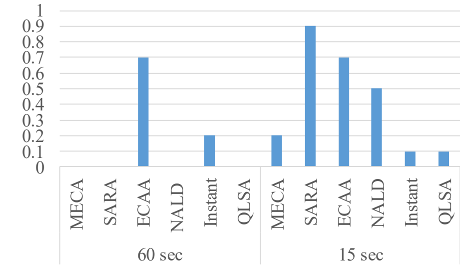


Fig. 5. The number of clients that experienced playback interruption when buffer size is set to 60s and 15s

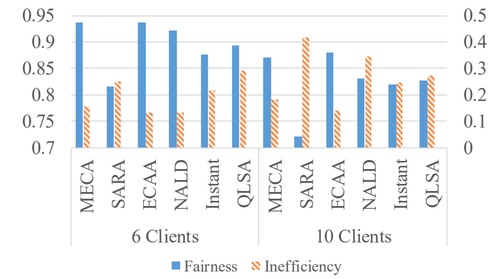


Fig. 6. Comparison of fairness and inefficiency values when the number of clients are change from 10 to 6 clients

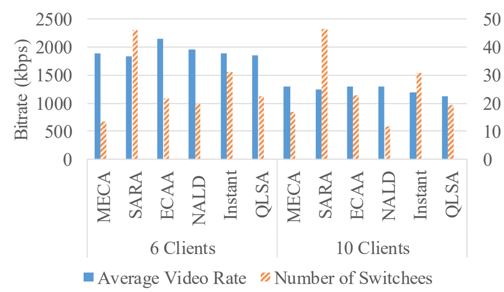


Fig. 7. Average video rates selected and average video rate switches experienced by the clients

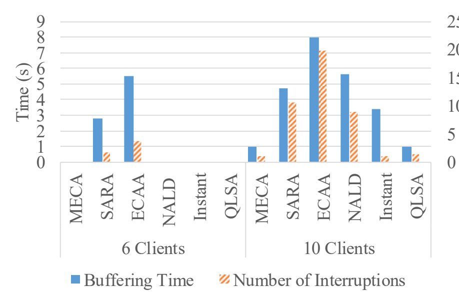


Fig. 8 Average rebuffering time and number of interruptions experienced by the clients

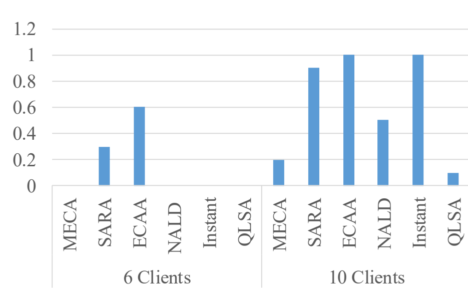


Fig. 9 The number of clients that experienced playback interruption

C. Impact of Segment Duration

In this experiment, we change the segment duration from 2 to 4 seconds. Similar to the previous experiments, Fig. 10 shows that the MEC-assisted algorithms achieve the highest fairness value and the lowest inefficiency value. Fig. also 11 shows that the MEC assisted algorithms are able to achieve the highest video rate while maintaining low number of video rate switches. Figs. 12 and 13 show that MECA algorithm is able to minimize the playback interruptions but the ECAA algorithm experiences high number of playback interruption when the segment duration is increased to 4 seconds. The MECA algorithm is able to mitigate the playback interruptions because it allows the client to suggest the upper bound on the feasible video rates based on the playback buffer level and available throughput at the client side.

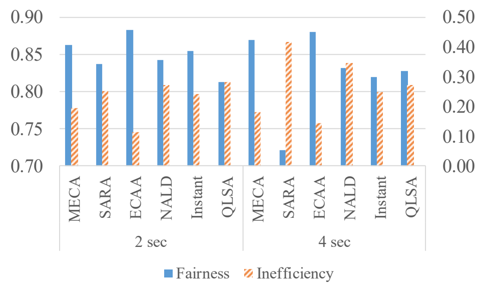


Fig. 10. Comparison of fairness and inefficiency values when the segment duration is changed from 4 seconds to 2 seconds

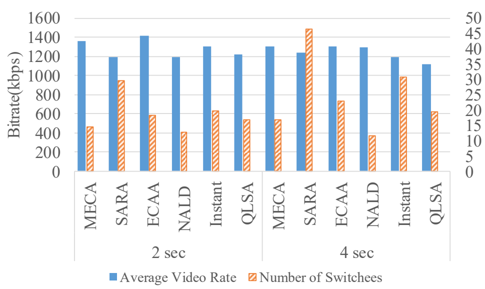


Fig. 11. Average video rates selected and average video rate switches experienced by the clients

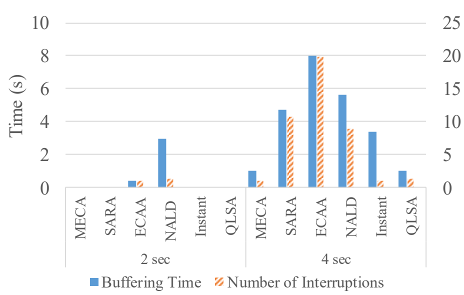


Fig. 12. Average rebuffering time and number of interruptions experienced by the clients

D. Impact of Clients Arrival Time

In this experiment, we compare the performance of the algorithms when all the clients simultaneously start streaming with the scenario when the clients randomly start the streaming session. Fig. 14 shows that along with MEC-assisted algorithms NALD algorithm is able to equitably allocate the video rate switches for the scenario when all the clients start streaming simultaneously. The MEC-assisted algorithms efficiently share the bandwidth. Fig. 15 shows that along with MEC-assisted algorithms, the NALD algorithm is also able to achieve high video rate while mitigating the video rate switches. The figure shows that the rest of the algorithm experience high number of video rate switches. As observed in previous experiments, Figs. 16 and 17 show that the ECAA algorithm continuously experienced playback interruptions for both scenarios.

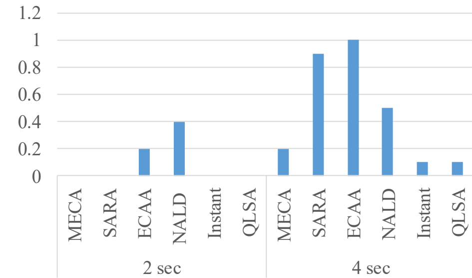


Fig. 13 The number of clients that experienced playback interruption

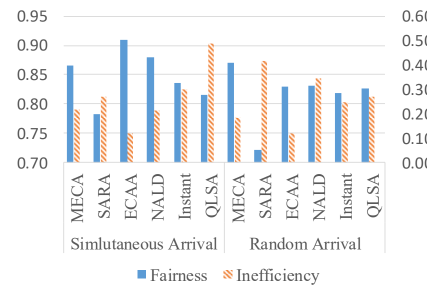


Fig. 14. Comparison of fairness and inefficiency values as the arrival time of the clients changes

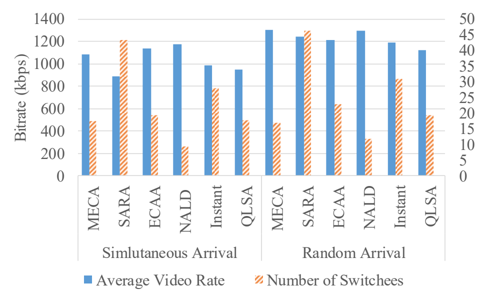
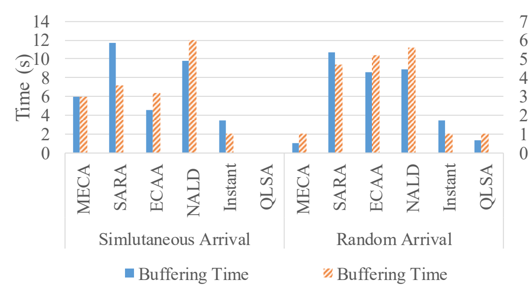


Fig. 15. Average video rates selected and average video rate switches experienced by the clients



Fig, 16. Average rebuffering time and number of interruptions experienced by the clients

E. Key Observations

The experiments show that the MEC-assisted algorithms are able to achieve high video rates compared to the greedy client based algorithms. Similarly, they are able to minimize the number of video rate switches. They assign equitable video rates to the clients and efficiently utilize the bandwidth.

The algorithms are able to utilize the knowledge of the competing clients and select high quality video rates equitably. However, the experiments show that the MEC-assisted algorithms experience playback interruptions. The MECA algorithm which takes a recommendation from the client on the selection of the video rates experiences fewer interruptions compared to the client side algorithms. However, ECAA algorithm achieves high video rate but at the expense of playback interruptions. The ECAA algorithm selects the highest video rate for the first segment irrespective of the available throughput. This depletes the playback buffer. Secondly, it does not consider the variations in the throughput and possibility of new clients joining the streaming session. It selects video rates aggressively. When the throughput suddenly drops or a new client arrives, it is not able to preserve the playback buffer. Although, the MEC-assisted algorithms jointly select the video rates, the algorithm should consider placing an upper bound on the set of feasible video rates that can be assigned to the clients based on the variations in the playback buffer level of individual clients. Moreover, the algorithms should take into consideration the clients throughput trend and mobility behavior.

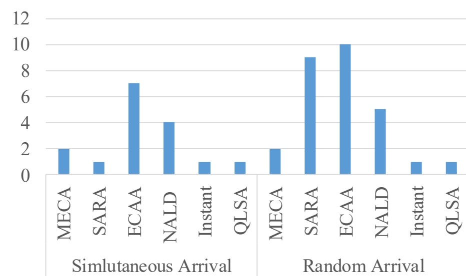


Fig. 17. The number of clients that experienced playback interruption

V. Conclusion

In this paper, we perform an in-depth evaluation of MEC-assisted quality adaptive algorithms under different client and server settings. We conduct extensive experiments of MEC-assisted adaptive algorithms with client based HTTP adaptive streaming algorithms and make several observations on the performance of these algorithms. The results show that the MEC-assisted algorithm are able to achieve high video rate, mitigate the video rate switches, select equitable video rates for the clients and efficiently utilize the bandwidth. However, MEC assisted algorithms are not able to protect the playback buffer from draining unnecessarily. Hybrid edge cloud and client adaptation algorithm experiences lesser playback interruptions compared to edge cloud adaptation algorithm.

In future, we plan to extend this work and evaluate the performance of MEC-assisted algorithms with more client based algorithms. Furthermore, we plan to evaluate the performances under varying client speeds, using different video data sets and network topologies.

References

[1] "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017-2022 White Paper," Feb 2019. Accessed: Apri. 27, 2019 [Online].Available:https://www.cisco.com/c/en/us/solutions/collateral /service-provider/visual-networking-index-vni/white-paper-c11-741490.html

[2] S. Egger, B. Gardlo, M. Seufert, and R. Schatz, "The impact of adaptation strategies on perceived quality of http adaptive streaming," In: Proceedings of ACM Workshop on Design, Quality and Deployment of Adaptive Video Streaming, pp. 31-36. Sydney, Australia (2014).

[3] Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A.C. Begen, and D. Oran, "Probe and adapt: Rate adaptation for HTTP video streaming at scale," IEEE J. Sel. Areas Commun. 32(4), 719-733 (2014).

[4] T. Y. Huang, H. Nikhil, H. Brandon, M. Nick, and J. Ramesh, "Confused, timid, and unstable: picking a video streaming rate is hard," In: Proceedings of ACM Int. conf. on Internet Meas., pp. 225-238. Boston, USA (2012).

[5] S. Akhshabi, L. Anantakrishnan, A.C. Begen, and C. Dovrolis, "What happens when HTTP adaptive streaming players compete for bandwidth?," In: Proceedings of ACM Workshop Netw. and Operating Syst. Support for Digital Audio and Video, pp. 9-14. Toronto, Canada (2012).

[6] W. U. Rahman and K. Chung, "A novel adaptive logic for dynamic adaptive streaming over HTTP," J. Vis. Commun. Image Representation. 49, 433-46 (2017)

[7] P. Juluri, V. Tamarapalli, and D. Medhi: SARA: Segment aware rate adaptation algorithm for dynamic adaptive streaming over HTTP. In: Proceedings of IEEE Int. Conf. on Commun. Workshop, pp. 1765-1770. London, United Kingdom (2015).

[8] M. Azumi, T. Kurosaka, M. Bandai, "A QoE-aware quality-level switching algorithm for adaptive video streaming," In: Proceedings of IEEE Global Commun., pp. 1-5. San Deigo, CA, USA (2015).

[9] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5G networks: New paradigms, challenges and scenarios," IEEE Communication Magazine. 55(4), 54– 61 (2017).

[10] S. Akhshabi, A. C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP," In: Proceedings of ACM Conf. on Multimedia Syst., pp. 157-168. California, USA (2011).

[11] C. Muller, S. Leder, C. Timmerer, "An evaluation of dynamic adaptive streaming over HTTP in vehicular environments," In: Proceedings of Workshop on Mobile Video, pp. 37-42, North Carolina, USA (2012).

[12] C. Liu, I. Bouazizi, and M. Gabbouj, "Rate adaptation for adaptive HTTP streaming," In: Proceedings of ACM Conf. on Multimedia Syst., pp. 169-174. California, USA (2011).

[13] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive," In: Proceedings of ACM Int. Conf. on Emerg. Netw. Experiments and Technol., pp. 97-108. Nice, France (2012).

[14] W. U. Rahman and K. Chung, "Chunk size aware buffer-based algorithm to improve viewing experience in dynamic HTTP streaming," IEICE Transactions on Commun. 99(3), 767–775 (2016).

[15] 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 SIGCOM Comput. Commun. Review. 44(4), 187-198 (2015).

[16] W. U. Rahman and K. Chung, "Buffer-based adaptive bitrate algorithm for streaming over HTTP," KSII Tran. Internet and Inform. Syst. 9(11), 4585-4622 (2015).

[17] W. U. Rahman, C. S. Hong and E. Huh, "Edge Computing Assisted Joint Quality Adaptation for Mobile Video Streaming," IEEE Access. 7(1), 129082–129094 (2019).

[18] A. Mehrabi, M. Siekkinen, and A. Ylä-Jääski, "Edge computing assisted adaptive mobile video streaming," IEEE Trans. Mobile Comput. 18(4), 1–17 (2018).

[19] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "BOLA: Near-optimal bitrate adaptation for online videos." In: Proceedings of IEEE International Conference on Computer Communications, pp. 1-9. San Francisco, USA (2016).

[20] J. Kua, G. Armitage, and P. Branch, "A survey of rate adaptation techniques for dynamic adaptive streaming over HTTP," IEEE Communications Surveys & Tutorials. 19(3), 1842-1866 (2017).

[21] D. Stohr, A. Frommgen, A. Rizk, "Where are the sweet spots? A systemic approach to reproducible dash playe rcomparisons," In: Proceedings of the ACM International Conference on Multimedia, pp. 1113-1121. Beijing, China (2017).

[22] R. Jain, D. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer system," Tech. Rep., Dec.1998.