

Improving Video Delivery with Fourier Analysis of Traffic in Multi-Access Edge Computing

Eryk Schiller¹, Remo Röthlisberger², Torsten Braun¹, and Mostafa Karimzadeh¹

Institute for Computer Science, University of Bern, Switzerland
¹{surname}@inf.unibe.ch, ²remo.r@gmx.ch

Abstract. This paper presents a new video delivery scheme in mobile networks using Multi-Access Edge Computing (MEC). Our goal is to improve the quality of video streaming experienced by the mobile video consumer. Our approach is based upon Dynamic Adaptive Streaming over HTTP. We present a novel algorithm, which uses information obtained from the Radio Network Information Service of MEC to provide the mobile user with a video quality matching the current radio link quality and channel capacity. We evaluate our approach using a real experiment performed on a Long Term Evolution (LTE) femto cell test-bed. Our algorithm displays enhanced adaptation of video rates in comparison to other state of the art solutions.

Keywords: Long Term Evolution (LTE) · Mobile Edge Computing (MEC) · Dynamic Adaptive Streaming over HTTP (DASH).

1 Introduction

The European Telecommunications Standards Institute (ETSI) Multi-Access Edge Computing (MEC) [1]¹ extends intelligence at the network edge through computing and storage facilities deployed in the close vicinity of the Radio Access Network (RAN). Due to MEC, video will greatly benefit from a low delay of the network edge, RAN-aware video content optimization, and adaptation to wireless network conditions. As ETSI mostly focuses on MEC video transcoding or content optimization for end-users [2], which requires a large number of computing resources at the MEC platform processing the video, other mechanisms are needed for better resource friendly streaming.

Dynamic Adaptive Streaming over HTTP (DASH) [3] opens up new opportunities in terms of MEC content optimization, as in DASH a video is already encoded with many representations at the content provider and video transcoding is not required at the MEC platform. The different representations are announced to the video consumer through a Media Presentation Description (MPD) file containing descriptors leading to locations of the video encoded with different qualities and, therefore, requiring distinct data rates for video

¹ Notice that in September 2016, ETSI Mobile Edge Computing group changed its name from Mobile Edge Computing to Multi-Access Edge Computing.

streaming. Typically, a higher representation requires more throughput than a lower one. The video consumer processes the MPD file and requests the video quality, i.e. representation, according to end-to-end measurements of the channel capacity.

In this work, we provide a MEC-based recommendation system for DASH (c.f., Sec. 3), which personalizes the video representations towards current channel conditions experienced by a User Equipment (UE). We observe the channel of every user by using a MEC Radio Network Information Service (RNIS) (c.f., Sec. 3.2) and compute the per-user radio channel capacity (c.f., Sec. 4) using novel Fourier-based traffic analysis. We then provide a dynamically generated MPD file towards the video consumer containing a limited set of video qualities matching the user channel capacity (c.f., Sec. 3.3). The client periodically fetches its personal MPD file containing suggested representations and regularly requests video segments using the set of representations adjusted by the video server to the experienced momentary channel capacity. Our solution does not change the DASH paradigm. We enrich the video system with personal MPD files. The MPD file is dynamically generated on the MEC-based video server (c.f., Sec. 3.1).

This work is organized in the following way. In Sec. 2, we survey the state of the art in MEC and video delivery. In Sec. 3, we describe the architecture of the system. The algorithm adapting MPD files for users is presented in Sec. 4. The performance evaluation and comparison against regular DASH is elaborated in Sec. 5. Finally, we conclude in Sec. 6.

2 Related Work

2.1 MEC

In ETSI MEC, the mobile ecosystem is enriched with a Multi-Access Edge Cloud (MEC) residing close to evolved Node Bs (eNBs) in LTE/4G or next generation Node Bs (gNBs) in 5G, which allows mobile users to contact MEC applications residing in the close vicinity of the UE [2]. MEC applications can be aware of the state of the air interface (e.g. capacity, congestion, radio signal quality, etc.) through RNIS implemented on top of MEC [2,4]. For example, FlexRAN [5] implements a flexible and programmable Software Defined-Radio Access Network (SD-RAN) platform, which separates the RAN control and data planes through a custom-tailored southbound API. It supports a real-time control channel that enables various degrees of coordination among RAN components.

2.2 DASH

Dynamic Adaptive Streaming over HTTP (DASH or 3GPP-DASH) [3] is a popular standard for video streaming over the Internet allowing for improved user experience in the presence of variable network conditions. Besides the conventional Hyper Text Transfer Protocol (HTTP), DASH consists of two main components, which are the Media Presentation Description (MPD) file and video segments residing on a HTTP server. The MPD file describes the characteristics of the stream. It contains information about the stream availability, segment

duration, video representations, and the resource identifiers for each segment. Typical DASH clients first request an MPD file and the first few segments of the video, in order to fill a buffer. When the buffer is filled, the player starts displaying the video to the client and the remaining segments are continuously fetched from the Internet. In conventional DASH Advanced Video Coding (AVC), the representation is selected by the client based on either buffer level or throughput-based algorithms. Buffer-based video streaming considers the buffer fill level to keep/improve the quality of the subsequent segment, if the necessary buffer re-fill level was experienced while downloading previous segments. Karagkioules et al. [6] surveys different adaptation algorithms. It is worth noting that there exist different kinds of video encodings and this work focuses on video delivery with DASH Advanced Video Coding (AVC). Many projects, especially studying improved caching strategies, work with DASH Scalable Video Coding (SVC) [7]. We work with DASH-AVC, because its implementation is simpler and therefore better adapted for resource-constrained end systems.

2.3 DASH improved with SDN and MEC

There are several different approaches improving video delivery in networks studied through simulations [8,9,10,11]. Cetinkaya et al. [8] uses DASH SVC and Software Defined Networking (SDN) to improve video streaming. The authors suggest routing video flows through the underlying infrastructure taking into consideration the capacity of the backhaul network. Li et al. [9] propose a Mobile Edge Computing (MEC) approach to improve fairness and overall video definition among UEs sharing the same channel. Lai et al. [10] propose a method for improved video delivery in heterogeneous networks with SDN. Fajardo et al. [11] propose a network-assisted HTTP streaming mechanism based on MEC. Their mechanism is able to adapt DASH streams to different channel conditions based on periodical measurements of Channel Quality Indicators (CQIs) and adaptation algorithms matching experienced CQIs to video definitions. Foukas et al. [5] prove a similar approach in a real experiment by using FlexRAN. In their use-case, CQI statistics reported by a UE are gathered at the FlexRAN controller. A DASH-based video streaming server uses information gathered at the controller to match the CQIs to video representations (i.e., bitrates).

2.4 Novelties of This Work

This work is composed of three innovations with respect to i) the architecture of an NFV-based MEC-assisted system for video delivery c.f., Sec. 3), ii) Fourier-based radio-channel assessment (c.f., Sec. 4), and iii) extensive measurements of a real system (c.f., Sec. 5).

We develop a MEC-based approach for video streaming based on real components containing a cloudified Core Network (CN), a MEC cloud, an eNB, and a UE. We demonstrate that a MEC-based video server (i.e., a MEC application) deployed on the MEC cloud can improve DASH-based video delivery for users connected to a wireless network.

Unlike other contributions, in the estimation of channel capacity [5,11], we do not use CQIs to assess the channel capacity. The problem of matching CQIs to channel capacity depends on technology, frequency, environment, vendor, radio scheduling, and hardware. Therefore, it is not feasible to derive exact tables matching CQIs and the desired rate of video delivery in a mobile system. Moreover, the use of CQIs does not apply to congested networks, in which the data rate is limited by the system capacity.

We do not work with absolute values such as channel capacity or representation bitrate. Therefore, our approach is fundamentally different from the typical knapsack problem, in which different representations requiring different bitrates are allocated within fixed channel capacity. In our approach, we observe patterns in channel consumption and wireless metrics. If the load is too high, we recommend the clients to lower their video representations, and when the load is too low, we recommend them to use higher video qualities. Moreover, we evaluate wireless metrics to predict the potential degradation of the channel capacity.

3 Architecture of the Video Delivery

3.1 Functional Architecture

Fig. 1 illustrates a MEC compliant network. It consists of RAN, a MEC cloud, and the CN. Since the MEC cloud is installed very close to the RAN, traffic originated by UEs can avoid traversing the CN and directly access MEC Applications (Apps). The MEC Apps benefit from the close vicinity of the eNB and thus experience low latency.

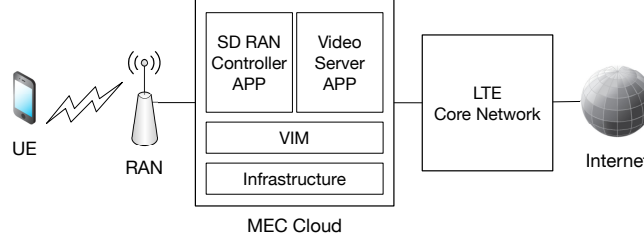


Fig. 1: A simplified depiction of the network architecture.

MEC Apps instantiated on a MEC cloud infrastructure through a Virtual Infrastructure Manager (VIM) receive traffic directly from the user plane (e.g., video service) or other MEC related services (e.g., SD-RAN platforms) dealing with radio control and management planes.

3.2 Platform Implementation

A 4G/5G mobile telecommunication platform developed in this work is depicted in Fig. 2. We use OpenAirInterface [12], which implements the Home Subscription Service (HSS), Mobility Management Entity (MME), and Serving/Package Gateway (S/PGW) as a minimal LTE CN and the eNB for RAN. OpenAirInterface provides the LTE mobile network and radio signal towards UEs. We

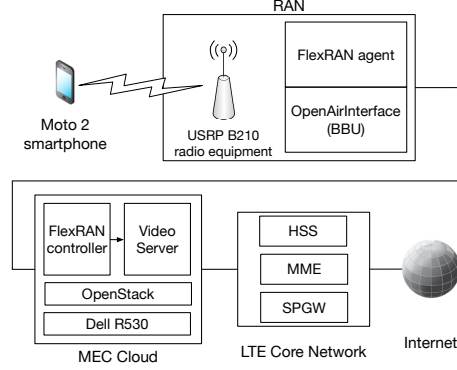


Fig. 2: The implementation of the experimental setup.

use the USRP B210 board², which provides an LTE Frequency Division Duplex (FDD) transmission in band 7 (2.5 GHz/2.6 GHz) using 5 MHz channels and the Single-Input Single-Output mode. We have a smartphone Moto 2³ connected to the OpenAirInterface network. As SD-RAN, we use FlexRAN [5]. FlexRAN consists of an agent co-located with the OpenAirInterface Base Band Unit (BBU)⁴, and an external SD-RAN controller, which is instantiated on the MEC cloud. Ubuntu-based MEC Apps (i.e., FlexRAN controller and Video Server) are both hosted as VNFs on an OpenStack based cloud with Dell R530 cloud workers (Intel Xeon 2.5 GHz CPU with 80 threads, 192 GB RAM).

3.3 Information Flow in the System

In order to provide the Video Service MEC App with information about the current radio link status (c.f., Fig. 2), we established an information flow between different components. Foukas et al. [5] implemented an SD-RAN controller (i.e., FlexRAN controller) that communicates with the eNB through a FlexRAN agent. The FlexRAN controller provides the MEC application with required radio link statistics. The SD-RAN controller is based on a publish/subscribe architecture and can periodically publish statistics about the per UE radio link quality, e.g., LTE CQI, Reference Signal Received Power (RSRP), etc.. CQI values are computed by the UE and reported on the uplink channel to the eNB. They are related to the current state of the Signal to Interference Plus Noise Ratio (SINR). CQI displays values between 0 and 15, where CQI of 15 denotes the best possible quality. RSRP is defined as the linear average over the power contributions (in [W]) of the resource elements (REs) that carry cell-specific reference signals within the considered measurement frequency band [13]. RSRP values reside between -44 dBm and -140 dBm [13].

Our video service subscribes with the FlexRAN controller and dynamically updates MPD files based on information provided by the controller. This leads to the information flow as described in Fig. 3. First, a UE starts accessing the

² <https://www.ettus.com/product/details/UB210-KIT>

³ <https://www.motorola.com/us/products/moto-z-play-gen-2>

⁴ the feature-68-enb-agent branch

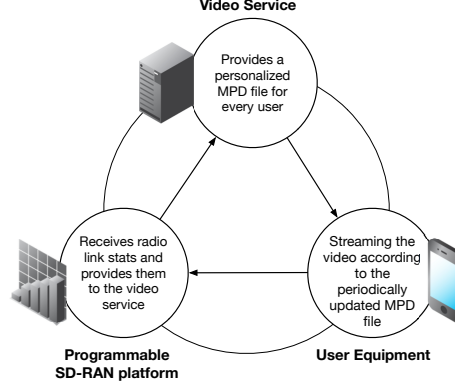


Fig. 3: The information and control flow.

video content from the MEC video service triggering traffic between the UE and the eNB. The radio signal quality is reported by the UE through the Radio Resource Control (RRC) to the eNB. eNB enriches the information received from UEs through the RRC channel with per-user traffic and RAN scheduling statistics, and sends this information to the FlexRAN controller using the FlexRAN agent. The FlexRAN controller publishes the information towards the subscribed video service, which in turn adapts (i.e., limits) available video qualities in the MPD file. Finally, the UE periodically downloads the MPD file, which closes the information loop between the system components.

4 Algorithm Selecting Video Representations

4.1 Characteristics of video traffic patterns

We assume that video streaming is the most dominant traffic of the user (i.e., other applications utilize only a small fraction of bandwidth on the mobile device) and also, that there is only one video stream running per user. It is not a strong assumption, while multiple video streaming is rare and users typically use one application on a UE at a time. Therefore, we assume minimal background traffic from other sources (i.e., instant messengers, email, etc.).

In our initial measurements, using a setup discussed in Sec. 3.2, we observe a specific periodical pattern, which is typical for video delivery with DASH AVC using buffer-based adaptation algorithms confirmed by other sources [5,6,14]. The traffic requested by the client appears as a periodical rectangle function (i.e., a channel is periodically occupied and idle with a certain frequency). This is caused by a periodical re-fill of the buffer. When the buffer level decreases below a certain threshold, a new segment is requested by the client. The periodicity of traffic peaks (i.e., rectangles) in the experienced goodput is approximately equal to the segment duration (e.g., 2 seconds) as the client consumes the video content in real time (a 2 second segment is consumed within 2 seconds).

Figs. 4a and 5a and related traffic plots reported by Augustin et al. [15] show that the DASH video pattern can be clearly distinguished from full-capacity downlink/uplink transmissions, which cause an approximately constant traffic

pattern limited by the channel capacity. We confirm the difference between the video stream pattern and full-capacity traffic in the time and frequency domain. Please notice a traffic peak at 0.5 Hz in Fig. 5b, which relates to the segment size $t_s = 2$ s.

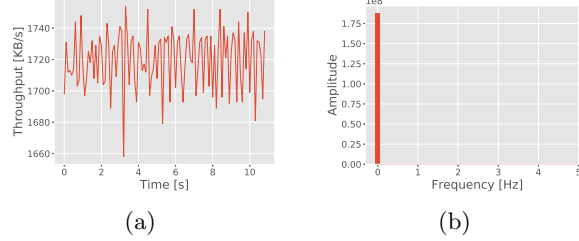


Fig. 4: Downloading a file using `wget`, results in constant throughput (a) and a FFT (b) with a spike at 0 Hz.

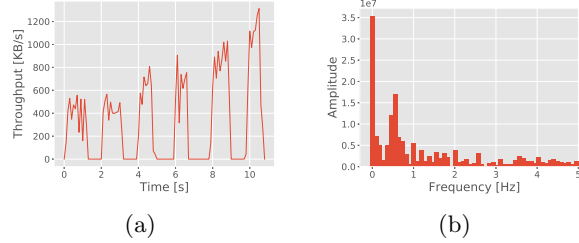


Fig. 5: Streaming a video reveals the DASH pattern in throughput (a) and a FFT (b) with a spike at 0 Hz and 0.5 Hz, corresponding to the segment size of two seconds.

Our idea is to use the Fast Fourier Transform (FFT) to discover traffic patterns on the downlink. If the link is saturated, it displays constant throughput behavior displayed in Fig. 4b. When we observe the saturation pattern on the link, we unload it by forcing the client to use lower video representations. Otherwise, if the link displays a regular video traffic with a significant peak correlated to the segmentation frequency, e.g., segments of 2 s cause peaks at 0.5 Hz, we can either keep the current definitions or force the client to use higher video representations. For more information about the use of Fourier Transforms, please consult [16] (c.f., p. 257).

We operate with sampling frequencies of around 10 Hz. There is no oversampling, as the periodicity of discrete packet-based transmission will be discovered with sampling rates of around 1000 Hz. For example, if a UE is exchanging packets of size $1500 \text{ B} = 12000 \text{ bits}$ with a throughput of 12 Mbps, data packets will be received at a frequency of 1000 Hz. Very high sampling rates of around 1000 Hz should be therefore avoided because of the risk of oversampling (i.e., moving spikes from 0.5 Hz to 1000 Hz).

4.2 Implementation

We specify a DASH *server side adaptation* mechanism, which also includes information about the quality of the wireless channel. Multiple Key Performance

Indicators (KPIs) exist to measure the channel quality in LTE. In our case, we use RSRP [13].

Algorithm 1 Simplified adaptation algorithm, based on radio link utilization and received radio signal power.

```

1: procedure MAIN LOOP( $T, i$ )     $\triangleright$  Duration  $T$  of video and interval  $i$  of sampling
2:    $t \leftarrow 0$                                  $\triangleright$  Variable running from 0 to  $T/i$ 
3:    $x \leftarrow$  segment duration
4:   delay  $\leftarrow$  time period (e.g., 1 sec)
5:    $max_r \leftarrow$  max tolerated RSRP drop (e.g., 5)
6:    $min_{\Delta} \leftarrow$  min margin (e.g., 0.04)
7:    $max_{\Delta} \leftarrow$  max margin (e.g., 0.4)
8:    $\triangleright$  As long as the video is playing
9:   while  $t \leq \frac{T}{i}$  do
10:     $S_t = \text{RNIS.getStats}()$ 
11:    if  $S_t.rsrp - S_{t-\frac{\text{delay}}{i}}.rsrp > max_r$  then
12:      Video rep. scale down; (c.f. Fig. 6a)
13:       $\triangleright$  Run a FFT over the last sampled data
14:       $\text{fft} = \text{runFFT}(S_{t-\frac{x}{i} \dots t}.throughput)$ 
15:      if  $\frac{\text{fft}[\frac{1}{x}]}{\text{fft}[0]} \geq max_{\Delta}$  then
16:        Video rep. scale up; (c.f., Fig. 6b)
17:      if  $\frac{\text{fft}[\frac{1}{x}]}{\text{fft}[0]} \leq min_{\Delta}$  then
18:        Video rep. scale down; (c.f., Fig. 6a)
19:       $t = t + 1$ 
20:      sleep(1)

```

Algorithm 1 presents a simplified implemented procedure. It limits available qualities (i.e., video representation scale-down) for the user, if the RSRP drastically decreases on average in the last second or a given FFT frequency ratio between video traffic at frequency $\frac{1}{t_s}$ and the constant traffic at 0 Hz is too small i.e., lower than min_{Δ} , indicating that the radio link is overloaded. On the other hand, it increases the video quality (i.e., video representation scale-up), if the given frequency ratio is above a certain threshold max_{Δ} .

The representations in the MPD file refer to the video qualities in ascending order, i.e., the lowest quality is provided by representation #1. In the scale-up, the server decides to replace the lowest representation with the next higher available quality. In scale-down, the highest representation is replaced with the next lower one available (c.f., Fig. 6).

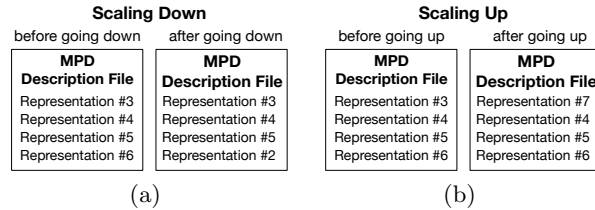


Fig. 6: The scaling down (c.f., Fig. 6a) and scaling up (c.f., Fig. 6b) procedures.

5 Evaluation of Video Delivery

The evaluation of the scheme is provided on a real LTE femto-cell testbed. Essentially, the physical layer, the noise, and competing video streams are taken into account in our measurements providing precise real-world measurements of our scheme.

5.1 Preparations of the Video Stream

In the experiment, we use the GPAC project toolset⁵, namely MP4Box, to encode and segment the video. MP4Client streams the video at the client side. In order to provide a DASH stream that could saturate the OpenAirInterface [12] wireless link (max. capacity of around 8 Mbps), we require a video with a high bit rate. Thus, we use a UHD 4K video. We encoded the video using MP4Box in 10 different bit rates, resulting in 10 different video representations requiring network capacity between 50 kbps up to 8 Mbps. The videos are divided into segments of $t_s = 2$ seconds, which is appropriately short for on demand video delivery in changing conditions of mobile networks.

5.2 Video Delivery Experiments

We use an OpenAirInterface-based [12] LTE setup (c.f., Sec. 3.2). We operate in a femto-cell scenario, where a user is moving inside a building (e.g., office space, train station, etc.). The eNB and video service are handling one UE.

In our experiment, we cover an office and a hallway with LTE signal and use a mobility pattern between different points of different radio signal qualities as shown in Fig. 7. We use the buffer-based adaptation technique (as implemented in the MP4Client). We compare our MEC-assisted approach, i.e., a video server receiving radio statistics from the SD-RAN controller (c.f., Sec. 3.1) and periodically updating the MPD file according to our algorithm specified in Sec. 4 against a native DASH solution (a video service providing static MPD files). Please notice that an MPD file is requested by the client once for the entire video stream or periodically every t_s for the regular and MEC-enabled DASH respectively. The algorithm parameters are $max_r = 5$, $min_\Delta = 0.04$, $max_\Delta = 0.4$. We compare two methods of video delivery residing very close to the user. We show the improvement of the video definition using the MEC-assisted server side adaptation.

Using a notebook connected to a smart phone Moto 2⁶, we run the video stream for approximately 90 seconds, while moving the phone according to the mobility pattern (c.f., Fig. 7). We stay at point *A* from 0 to 15 s, move from *A* to point *B* during 5 s, stay at point *B* between 20 s and 35 s, and move from *B* to *C* during 6 s. Then, we stay at *C* between 41 s and 60 s, and go back to *B* again during 6 s, and stay there between 66 s and 90 s. On the way between *A* to *C*, and *C* to *B*, we experience decreasing and increasing signal levels respectively.

5.3 Results for One UE: Buffer-based Adaptation

There are many different client-side adaptation mechanisms. However, in this work, we compare our MEC enabled video streaming technique with state-of-

⁵ GPAC, multimedia open source project: <https://gpac.wp.imt.fr/>

⁶ <https://www.motorola.com/us/products/moto-z-play-gen-2>

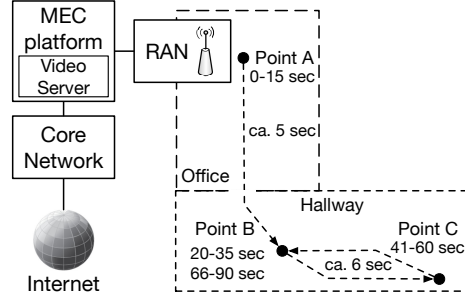


Fig. 7: Moving pattern in the office and hallway.

the-art, buffer-based mechanisms implemented in the GPAC client. Due to variable radio conditions, the comparison is based on a statistical basis repeating the same experiment 10 times. The measurements of requested representation qualities against segment number and client buffer fill levels against time in 10 experiments vary significantly.

Analysis over 10 measurements for each of the adaptation mechanisms, i.e., regular DASH vs. our MEC-assisted DASH, provides an appropriate statistical estimation of the experienced buffer level and video quality at the UE. Fig. 8a and Fig. 8b show the difference between the slower buffer-based adaptation and the faster MEC-assisted adaptation. Our algorithm provides a higher initial representation, c.f., Fig. 8a from segment #10 onwards, but then suffers from a quick drop in the buffer, as we move away from the eNB, c.f., Fig. 8b at around 20 seconds, when moving from point *B* to *C*. Due to the margin, the buffer stays at an appropriate level and we can keep the higher quality of the video. At point *C* (c.f., Fig. 7), which manifests the worst radio conditions, we need to decrease the video representation (segment #28). The buffer relaxes and performs better than regular DASH. Notice that in regular DASH, we experience a similar, but stronger drop in the buffer level and this affects the video quality in a negative way (c.f., Fig. 8a red line, at around segment #33-#40). With MEC-assisted DASH, we improved the video quality in the good signal conditions (between 10 and 20 seconds) up to 40%, while keeping the buffer level at an approximately 35% lower, but stable level.

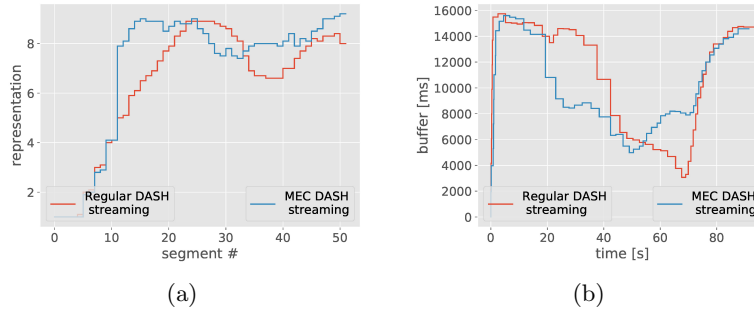


Fig. 8: (a) Requested video representation against segment no# averaged over 10 experiments for regular and MEC-assisted DASH. (b) Buffer level against time averaged over 10 experiments for regular and MEC-assisted DASH.

5.4 Results for Two UEs: Throughput-based Adaptation

We attach two laptops, i.e., machine #1 and machine #2. Every laptop is equipped with a Huawei E3276 dongle⁷. The laptops have fixed positions (i.e., RSRP displays -78 dB and -82 dB for machine #1 and machine #2 respectively). They simultaneously and concurrently stream the same video stream from the video server through the LTE network (c.f., Sec 3.2). In comparison to

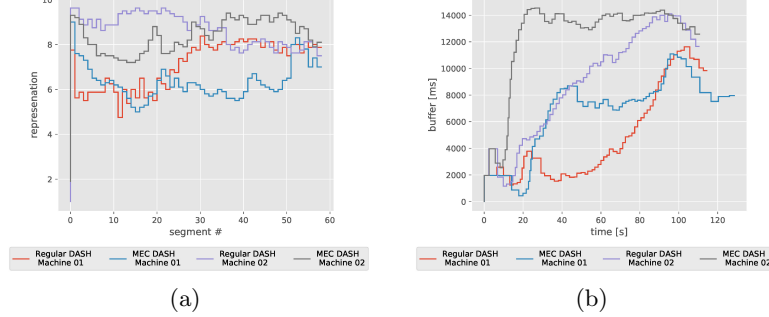


Fig. 9: (a) Average representations on machines #1 and #2. (b) Average buffer fill level on machines #1 and #2.

rate based regular DASH, our algorithm (i.e., mainly the Fourier channel assessment as the RSRP values remain stable throughout the experiment) provides slightly decreased representation qualities (c.f., Fig. 9a, machine #2), however, operates with a much more reasonable buffer fill level (c.f., Fig. 9b). This is beneficial in mobile scenarios, in which the connecting quality can quickly vary, so the buffer shall not be maintained at low levels. This proves that our algorithm also behaves appropriately in multi-user scenarios.

6 Conclusions

We provide a proof of concept improving MEC capabilities in regular, “off-the-shelf” DASH AVC that controls video qualities available for the client depending on various metrics. In particular, we provide our own MEC DASH adaptation algorithm and compare it against regular buffer-based DASH. We experience much faster adaptation to good radio conditions as well as better experience in terms of worse signal quality, when the buffer remains at the higher level. We are convinced that MEC is beneficial for video streaming so that DASH can profit from improved performance in spite of mobility.

References

1. Giust, F., Verin, G., Antevski, K., Chou, J., Fang, Y., Featherstone, W., Fontes, F., Frydman, D., Li, A., Manzalini, A., Purkayastha, D., Sabella, D., Wehner, C., Wen, K.W., Zhou, Z.: MEC Deployments in 4G and Evolution Towards 5G. ETSI White Paper No. 24 (2018)
2. Patel, M., Hu, Y., Hédé, P., Joubert, J., Thornton, C., Naughton, B., Ramos, J.R., Chan, C., Young, V., Tan, S.J., Lynch, D., Sprecher, N., Musiol, T., Manzanares, C., Rauschenbach, U., Abeta, S., Chen, L., Shimizu, K., Neal, A., Cosimini, P.,

⁷ <https://consumer.huawei.com/en/mobile-broadband/e3276/>

- Pollard, A., Klas, G.: Mobile-Edge Computing – Introductory Technical White Paper. ETSI White Paper (2014)
3. 3rd Generation Partnership Project (3GPP): Transparent end-to-end Packet-switched Streaming Service (PSS); Progressive Download and Dynamic Adaptive Streaming over HTTP (3GP-DASH). Tech. rep., 3GPP (2015)
 4. Chang, C.Y., Alexandris, K., Nikaein, N., Katsalis, K., Spyropoulos, T.: Mec architectural implications for lte/lte-a networks. In: Proceedings of the Workshop on Mobility in the Evolving Internet Architecture. pp. 13–18. MobiArch '16, ACM, New York, NY, USA (2016)
 5. Foukas, X., Nikaein, N., Kassem, M.M., Marina, M.K., Kontovasilis, K.: Flexran: A flexible and programmable platform for software-defined radio access networks. In: Proceedings of the 12th International on Conference on Emerging Networking EXperiments and Technologies. pp. 427–441. CoNEXT '16, ACM, New York, NY, USA (2016)
 6. Karagioules, T., Concolato, C., Tsilimantos, D., Valentin, S.: A comparative case study of http adaptive streaming algorithms in mobile networks. In: Proceedings of the 27th Workshop on Network and Operating Systems Support for Digital Audio and Video. pp. 1–6. NOSSDAV'17, ACM, New York, NY, USA (2017)
 7. Müller, C., Renzi, D., Lederer, S., Battista, S., Timmerer, C.: Using scalable video coding for dynamic adaptive streaming over http in mobile environments. In: 2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO). pp. 2208–2212 (Aug 2012)
 8. Cetinkaya, C., Ozveren, Y., Sayit, M.: An sdn-assisted system design for improving performance of svc-dash. In: 2015 Federated Conference on Computer Science and Information Systems (FedCSIS). pp. 819–826 (Sept 2015)
 9. Li, Y., Frangoudis, P.A., Hadjadj-Aoul, Y., Bertin, P.: A mobile edge computing-assisted video delivery architecture for wireless heterogeneous networks. In: 2017 IEEE Symposium on Computers and Communications (ISCC). pp. 534–539 (July 2017)
 10. Lai, C., Hwang, R., Chao, H., Hassan, M.M., Alamri, A.: A buffer-aware http live streaming approach for sdn-enabled 5g wireless networks. *IEEE Network* **29**(1), 49–55 (Jan 2015)
 11. Fajardo, J.O., Taboada, I., Liberal, F.: Improving content delivery efficiency through multi-layer mobile edge adaptation. *IEEE Network* **29**(6), 40–46 (Nov 2015)
 12. Nikaein, N., Knopp, R., Kaltenberger, F., Gauthier, L., Bonnet, C., Nussbaum, D., Ghaddab, R.: Demo: Openairinterface: An open lte network in a pc. In: Proceedings of the 20th Annual International Conference on Mobile Computing and Networking. pp. 305–308. MobiCom '14, ACM, New York, NY, USA (2014)
 13. 3GPP: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer; Measurements. Technical Specification (TS) 36.214, 3rd Generation Partnership Project (3GPP) (03 2017), version 14.2.0
 14. Han, B., Qian, F., Ji, L., Gopalakrishnan, V.: Mp-dash: Adaptive video streaming over preference-aware multipath. In: Proceedings of the 12th International on Conference on Emerging Networking EXperiments and Technologies. pp. 129–143. CoNEXT '16, ACM, New York, NY, USA (2016)
 15. Augustin, B., Mellouk, A.: On traffic patterns of http applications. In: 2011 IEEE Global Telecommunications Conference - GLOBECOM 2011. pp. 1–6 (Dec 2011)
 16. Smith, S.W., et al.: The scientist and engineer's guide to digital signal processing (1997)