

# QoE-Driven DASH Video Caching and Adaptation at 5G Mobile Edge

Chang Ge, Ning Wang, Severin Skillman, Gerry Foster, Yue Cao  
Institute for Communication Systems, Department of Electrical and Electronic Engineering  
University of Surrey, Guildford, UK  
{c.ge, n.wang, s.skillman, g.foster, y.cao}@surrey.ac.uk

## ABSTRACT

In this paper, we present a Mobile Edge Computing (MEC) scheme for enabling network edge-assisted video adaptation based on MPEG-DASH (Dynamic Adaptive Streaming over HTTP). In contrast to the traditional over-the-top (OTT) adaptation performed by DASH clients, the MEC server at the mobile network edge can capture radio access network (RAN) conditions through its intrinsic Radio Network Information Service (RNIS) function, and use the knowledge to provide guidance to clients so that they can perform more intelligent video adaptation. In order to support such MEC-assisted DASH video adaptation, the MEC server needs to locally cache the most popular content segments at the qualities that can be supported by the current network throughput. Towards this end, we introduce a two-dimensional user Quality-of-Experience (QoE)-driven algorithm for making caching / replacement decisions based on both content context (e.g., segment popularity) and network context (e.g., RAN downlink throughput). We conducted experiments by deploying a prototype MEC server at a real LTE-A based network testbed. The results show that our QoE-driven algorithm is able to achieve significant improvement on user QoE over 2 benchmark schemes.

## Categories and Subject Descriptors

D.7 [Networks]: Network Services; D.3.3 [Networks]: Network Components—*Middle boxes / network appliances*

## 1. INTRODUCTION

The delivery of mobile content, in particular of ultra-high definition (UHD) videos with 4K resolution, has been envisaged as a prominent use case in the context of future 5G network development. It has been widely recognized that 5G is not only about increasing network capacity, but also about assuring service quality through enabling necessary network intelligence in complex and dynamic environments. With the recent development of new networking paradigms, in particular Information Centric Networking (ICN [6–8,13])

and Mobile Edge Computing (MEC [4]), it is envisaged that content awareness and intelligence can be embedded at the mobile network edge (e.g. at eNodeBs, or eNBs in LTE networks) in order to achieve desirable user quality of experiences (QoE) against various dynamicity and uncertainty in the network ecosystem. On one hand, ICN paradigms advocate the principle of putting content objects as the focus of networking landscape, with routing, forwarding and caching operations purely based on their names for efficiency. This is in contrast to the traditional host-centric approach where the physical locations of content sources need to be identified before an application-layer request (such as HTTP) is issued targeting the resolved content source by the Domain Name System (DNS). Such a networking philosophy enables content awareness by the underlying network platform, in the sense that the network is able to understand specific content delivery requirements and conditions when performing content handling operations.

In recent years, the concept of MEC has also been proposed by ETSI (European Telecommunications Standards Institute), with the initial aim of developing content-oriented network intelligence at the mobile network edge in 5G. Concerning mobile content management, such a paradigm can be gracefully built on top of an ICN underlay that offers native name-based representation, routing and forwarding functionalities. Traditionally, the ICN design normally does not cater for user QoE in content delivery applications. However, concerning video streaming applications, one distinct technical challenge is to deal with the dynamicity of radio resource availability at Radio Access Network (RAN), which can substantially impact user QoE during video playback. In addition, there is also uncertainty in the network conditions at the mobile backhaul and also in public Internet if the content source resides remotely. In order to tackle such issues, one possible approach is content localization, i.e., to bring the content close to content consumers. In LTE networks, this typically means caching content at eNBs. The benefits of such a strategy are twofold. First, the availability of content objects at the network edge eliminates the risk of suffering from backhaul and public Internet network condition deteriorations during video streaming sessions. Second, the mobile edge is able to directly capture the RAN conditions and perform prompt guidance on video adaptations accordingly, without necessarily relying on end-to-end adaptation at either users or content sources, which is slow in responding to the quickly-changing RAN conditions.

In this paper, we introduce an MEC-enabled ICN-based content handling framework at the mobile network edge,

which realizes context-aware content localization in order to enhance user QoE in video distribution applications. We specifically focus on DASH (Dynamic Adaptive Streaming over HTTP) based video applications, in which a video content is divided into multiple segments that can be independently requested during a video streaming session. Furthermore, each video can be encoded into different qualities<sup>1</sup> in terms of resolution and bitrates.

In conventional DASH video streaming applications, user devices (i.e., DASH clients) adaptively request different video qualities according to dynamic network conditions in order to maintain satisfactory user QoE. In contrast to this over-the-top (OTT) approach, which is slow in reacting to the highly fluctuating RAN conditions, our proposed scheme relies on the knowledge collected at the network edge to serve locally cached video segments at their right representations according to MEC-predicted RAN conditions. It is worth mentioning that such a solution is not designed on top of any specific ICN scheme, as the main interoperability issue is the understanding by the underlying ICN network on the URL-based naming of DASH video segments [9].

While the majority of the content caching schemes in the literature mainly focus on system efficiency (e.g. cache hit statistics) through popularity-based decision-making logic, the main focus of this paper is to provide assurance to user QoE against dynamic RAN conditions, which is achieved through context-aware caching and serving of DASH segments at their appropriate quality levels. Towards this end, we propose a two-dimensional QoE-driven content caching and adaptation scheme, which holistically takes into account the popularities of both video segments and their representations. Given limited content caching storage at the mobile edge, it can be inferred that a multi-objective optimization for caching / replacement decision is required to offer the best trade-off between content popularity and representation popularity. This is of particular importance to popular video codecs such as H.264 and H.265, where different representations of a video segment are stored in independent files. Given the changing RAN condition and user device heterogeneity, multiple files may need to be cached for a single segment, which in turn may reduce the number of distinct segments that are cached. To tackle such a challenge, our proposed QoE-driven caching / replacement algorithm caters for optimized balance between the two objectives in order to best support user QoE in dynamic network conditions. We believe that the proposed scheme will empower the future content-aware 5G mobile network with embedded network intelligence for comprehensively improving user QoE in video streaming applications.

## 2. BACKGROUND REVIEW

HTTP-based video streaming applications (e.g., MPEG-DASH) have become increasingly popular in the Internet. In DASH, a video is divided into multiple segments so that clients can progressively download and buffer future segments during video playback. Furthermore, each segment is encoded into multiple qualities, so that when a DASH client requests a segment, it can choose from a number of video qualities (i.e., bitrates) to adapt to its perceived dynamic network condition. In the literature, significant re-

<sup>1</sup>We use the terms “quality” and “representation” interchangeably in this paper.

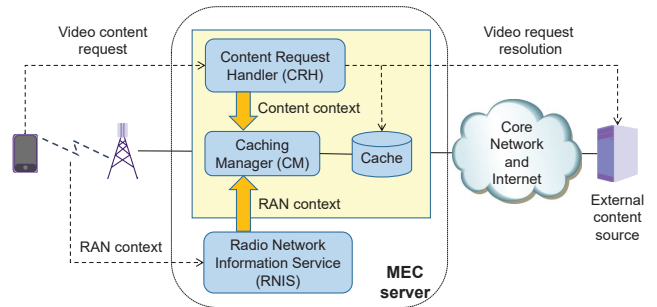


Figure 1: System overview

search efforts have been invested on optimizing DASH video adaptations at the client side [10, 11, 14].

In-network content caching has been regarded as a key feature in ICN designs. With such a network function, incoming content requests can be resolved to a nearby content cache router which is able to directly serve the content consumer without involving the original server. Traditionally, researches on in-network caching have been mainly conducted in the context of fixed ISP networks. However, there have also been recent proposals that are made towards caching at the edge of mobile cellular networks [12] or at the device side [3].

With the advent of MEC, content adaptation and caching can be empowered by smart network functions embedded at the network edge, based on which content user QoE can be substantially enhanced under dynamic network conditions. In [4], a MEC server at the mobile edge can directly provide its knowledge on the RAN conditions as feedback to a remote content server when an end user is consuming video content. Such knowledge can be used to assist the remote content server to promptly adapt TCP parameters (e.g., congestion window) to improve TCP throughput maintain assured user QoE under dynamic RAN conditions. This is in contrast to the traditional reactive approach where the content server relies on mobile client perceived network conditions in order to make adaptations which lacks agility and efficiency. In [5], the authors proposed a MEC-based platform that is able to perform video adaptation for multiple user groups with differentiated QoS classes.

## 3. SYSTEM OVERVIEW

Figure 1 presents the overall system architecture of our proposed scheme. A dedicated ICN network function, known as content request handler (CRH), is typically maintained at the network edge (e.g. eNBs or aggregation points to the north of them). CRH is responsible for resolving incoming video content requests to either remote servers or the MEC server’s local cache. Without loss of generality, CRH takes similar content representation functions as existing ICN schemes, such as an NDN (Named Data Networking) access router in [1] or a content server in [2], in which case content requests are directly resolved and routed towards best-selected sources hop-by-hop.

The caching manager (CM) acts as the decision-maker for caching / replacing content (e.g., video segments) at the local cache. As previously mentioned, such decision making needs to take into account the context information on both

content popularity characteristics and dynamic RAN conditions, so as to determine which are the most appropriate representations of individual segments to be cached for future requests. Towards this end, the CM needs to collaborate with the Radio Network Information Service (RNIS), which is an intrinsic functional component of the MEC paradigm. Specifically, RNIS is responsible for capturing up-to-date RAN conditions and report back to CM. Meanwhile, CRH can also report necessary context information on content, segment and representation popularity to CM. The CM then takes both context knowledge as input and determines which segments / representations are the best to be cached locally. As such, it can be inferred that an intelligent algorithm for context-aware caching / replacement decision-making is needed by CM. Such an algorithm will be presented in the next section.

Concerning naming for individual DASH-like video segments with multiple representations, the authors of [9] have proposed a mechanism to map them to the NDN naming system. Specifically, a DASH segment at a given representation is named as:

*[Video\_content\_name].[Representation\_ID].[Segment\_ID]*

With such a naming scheme, a DASH client can adaptively make segment requests based on the above naming structure, according to its perceived network throughput. Once the requested content segments arrive at the MEC server, its CM function will apply the caching / replacement algorithm to each segment with specific representation, which is described in the next section.

## 4. CONTEXT-AWARE CACHING AND REPLACEMENT OF CONTENT

In this section, we present our proposed context-aware caching and replacement algorithm, which is driven by the following two context factors: a) the **popularity** (i.e., total number of times being requested) of each content, each video segment and each representation of it; and b) the **network conditions** that affect each user’s downlink throughput, such as bandwidth, latency, packet error, etc.

The objectives of the proposed algorithm are twofold. First, it aims to provide enhanced user QoE in video streaming applications in terms of video quality assurance. Second, it ensures that such QoE enhancement is provided to as many clients as possible.

### 4.1 Caching Strategy

The key principle of our proposed caching strategy is to ensure that the bitrates of cached video representations match the network’s downlink capacity that is experienced by individual users. For example, if a large number of users have been streaming video in a small cell, then video segments with high bitrates will not be cached at the eNB. This is because even if they are cached, each user will not be allocated with enough network (radio) resource to stream them without experiencing frozen playback. In the meantime, more lower-bitrate segments can be cached so that more distinct video requests can benefit from them.

As shown in Figure 1, such caching strategy is enforced by the caching manager (CM) function, which uses context knowledge provided by RNIS component in the MEC server.

If CM decides to cache a video segment based on the strategy above, the replacement algorithm in the next subsection will decide which currently-cached segments need to be deleted to make space for the new incoming segment.

### 4.2 Cache Replacement Strategy

There are two cache replacement strategies that have been traditionally adopted, which are Least Frequently Used (LFU) and Least Recently Used (LRU). If they are directly applied to DASH video segments that follow the naming structure as discussed earlier in Section 3, the cached segments that belong to the same video may be scattered during the replacement process, hence causing a video’s cached segments to be discontinuous. If a user streams a video with one segment cached locally and the next one not cached, the user QoE could be very different when watching these 2 segments. Furthermore, significant change in playback quality further degrades user QoE.

In order to ensure user QoE in terms of a) maximizing streamed video quality through content localization; and b) minimizing playback quality fluctuation through ensuring segment continuity in a cached content, we introduce a context-aware **QoE-driven** video segment replacement algorithm. Same as the caching strategy, such an algorithm is enforced at the CM function at the MEC server.

---

#### QoE-Driven Cache Replacement Algorithm

---

**Input:** *spaceToBeFreed* number of bytes to be freed

**Output:** the video rep<sup>2</sup> that are deleted

```

1: begin: freedSpace ← 0
2: sort cached content with ascending popularity
3: for each cached content
4:   sort its cached segments in ascending popularity
5:   for each cached segment
6:     // Keep at least 1 rep in each segment
7:     if segment has 1 cached rep, skip
8:   else
9:     sort its cached reps in ascending popularity
10:    for each cached rep
11:      delete the rep
12:      freedSpace += rep’s size
13:      if freedSpace >= spaceToBeFreed end
14: end
15: // All cached segments now have 1 cached rep
16: for each cached segment
17:   delete segment and its 1 cached rep
18:   freedSpace += rep’s size
19:   if freedSpace >= spaceToBeFreed end
20: end

```

---

In a nutshell, the algorithm above determines what existing cached segment(s) / representation(s) should be deleted to make room for a newly cached segment. The algorithm first sorts all cached contents based on their *content-level* popularity, i.e., how many times has each video been streamed. Then, starting from the least popular content, the algorithm sorts all cached segments based on their *segment-level* popularity, i.e., how many times has each segment been requested in all streaming sessions. Afterwards, the algorithm performs a 2-stage cache replacement operation:

---

<sup>2</sup>Short for representation

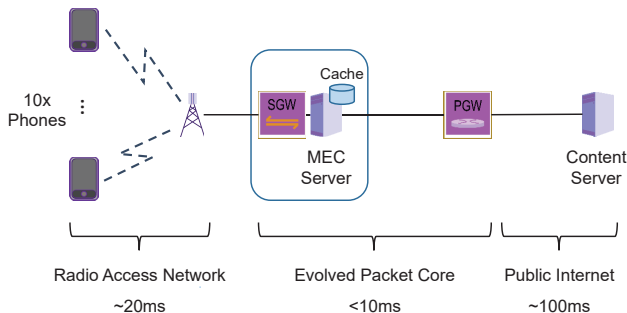


Figure 2: Experiment setup on the LTE-A testbed

- Stage 1: iterate through all cached segments, but skip the segments with only one representation cached. For the segments with more than one representations cached, delete the representations starting from the least popular one. This stage ensures that as many distinct segments are cached as possible for each video.
- Stage 2: if the algorithm reaches this stage, it means that all the remaining cached segments each have only one representation cached. Hence, in this stage, the segments (and their one cached representation) are deleted in ascending order of popularity until enough space is freed.

There are two key principles of our proposed cache replacement algorithm. First, it ensures that if a video is cached, all of its cached segments are consecutive to each other. Second, it maximizes the number of distinct segments that is cached for each video. Through these two principles, our algorithm is able to maximize user QoE by ensuring constant playback video quality (through ensuring segment continuity) that is as high as possible (through maximizing the number of cached segments).

Compared with traditional cache replacement strategies such as LFU / LRU, which treats video segments / representations as individual files, our algorithm is expected to reduce the side effects of ad-hoc cache replacement that may cause QoE fluctuation during video streaming. This is further evaluated in the next section.

## 5. PERFORMANCE EVALUATION

### 5.1 Experiment Setup

The performance evaluation is carried out by experiments that are performed on the LTE-A C-RAN wireless testbed that is hosted by the 5G Innovation Center at University of Surrey. Figure 2 illustrates the experiment setup. As shown in the figure, a MEC server prototype has been deployed at the mobile network edge. More specifically, it is deployed between the Service Gateway (S-GW) and the Packet Gateway (P-GW). The CRH, CM and RNIS functionalities, as presented in Section 3, have been implemented on the MEC server. It is also equipped with a content cache with the size of 800MB.

Regarding network latency, the round-trip time (RTT) between each UE and eNB on the air interface is around 15 to 20ms. The latency in the EPC is relatively low (less than

Table 1: Encoded Video Representations

Bitrate	Resolution
0.1, 0.4, 0.7Mbps	480x360
1.5Mbps	1280x720
2.5, 4, 8Mbps	1920x1080
15, 20Mbps	3840x2160

Table 2: Rules on Generating Streaming Sessions

Phone ID	Requested Video	Playback Session Duration (Random)	Playback Session Interval
1 to 7	Video 1	2:30 to 4:00	10s to 60s (random)
8 to 9	Video 2	2:00 to 3:00	
10	Video 3	1:30 to 2:30	

10ms), which means that if a requested content is cached at the MEC server, the latency between the UE and the content is at the order of 20ms. The latency between the P-GW and the content server is around 100ms with a 10ms fluctuation, which is emulated with Linux tc netem toolbox on the content server.

3 pieces of video content, which are 3 identical copies of the video Big Buck Bunny<sup>3</sup>, were used in the catalog. The video itself is 9 minutes 56 seconds long. Each copy of the video is encoded from raw YUV format into 9 different representations using H.264 codec, and each representation is divided into 120 segments where each segment is 5 seconds long. The details on the encoded video representations are listed in Table 1.

We used 10 Huawei Nexus 6P phones as UEs to perform the experiments. All of them are connected to an indoor eNB via Band-41 LTE-A network. The Band-41 network is TDD-based and has the downlink capability of approximately 100Mbps, where the radio resource is shared among the active UEs. Software wise, each phone runs Android 6.0.1 operating system, and uses Google Chrome browser to run the DASH reference player v2.1.1, which is a JavaScript-based DASH client<sup>4</sup>.

On each phone, the video streaming sessions are generated based on the rules in Table 2. These rules are designed to ensure that a) the content popularity follows zipf distribution; and b) the arrivals of user requests follow Poisson distribution. For this paper, the experiment ran for 32 minutes and 8 seconds (1,928 seconds in total), where videos 1, 2 and 3 were streamed in 45, 14 and 7 sessions respectively (66 sessions in total). Overall, approximately 2,900 segment requests were made.

### 5.2 Performance Metrics and Reference Schemes

We analyze the following KPIs to evaluate the benefits of our proposed QoE-driven cache management scheme.

- **Video quality:** this refers to the quality of the video segments that are requested during each streaming session. The video quality is defined as in bitrate (Mbps)

<sup>3</sup>Available at <https://peach.blender.org/download/>

<sup>4</sup>Available at [github.com/Dash-Industry-Forum/dash.js/](https://github.com/Dash-Industry-Forum/dash.js/)

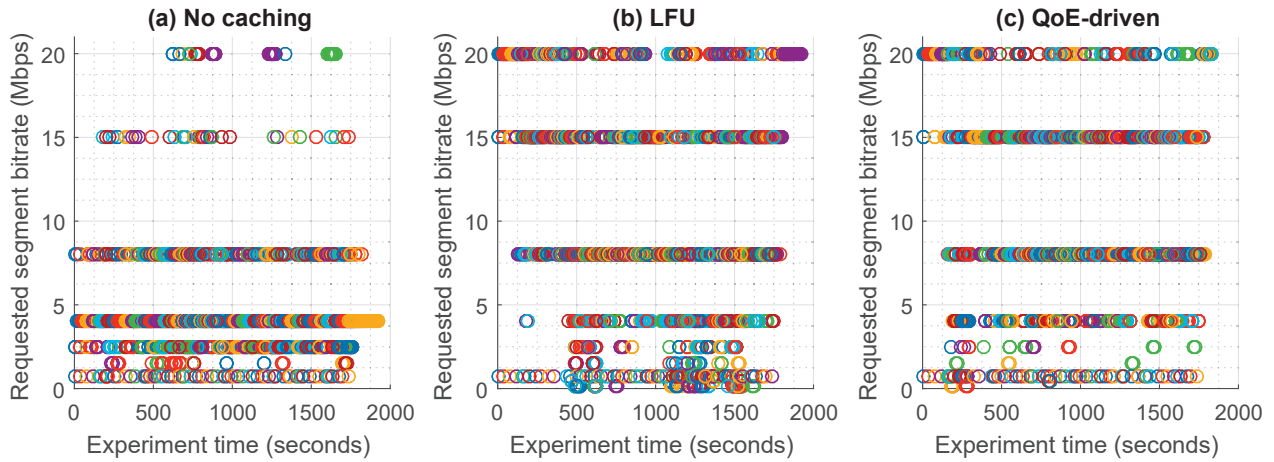


Figure 3: Video bitrate that each DASH client requests for each segment: comparison among cache replacement schemes

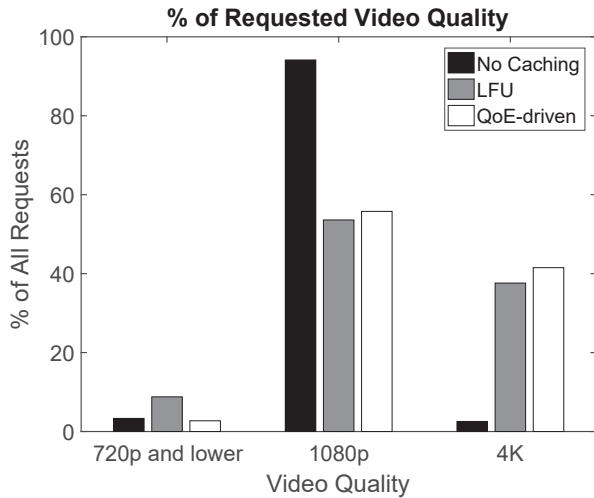


Figure 4: Breakdown of requested video quality among all video segment requests

as well as resolution (360p, 720p, 1080p and 4K).

- **Video quality switching statistics:** this refers to the events when video quality changes during a playback session. While this is common as a DASH client performs video quality adaptation, user QoE can be degraded by frequent quality switching events, especially the ones where there is a big change in video quality (e.g., from 4K to 360p).

We compare our QoE-driven caching scheme against 2 reference schemes.

- **No Caching:** the UEs stream videos directly from the content server with no MEC server in the middle. In this case, the latency between each UE and its requested video is always around 120ms.
- **LFU:** the MEC server performs standard LFU caching,

Table 3: Statistics on Video Quality Switching

	Total	Total Number of Switched Quality Levels	Switching across 720p: Total/Average-per-Session
No Caching	720	1029	68 / 1.03
LFU	683	1415	135 / 2.08
QoE-driven	567	939	75 / 1.14

which is based on the ICN naming structure proposed in [9].

### 5.3 Results on Video Quality

Figure 3 presents the bitrates of each requested DASH video segment under our proposed QoE-driven caching scheme and the two benchmark schemes. The results involve 66 video streaming sessions under each scheme.

First, it is directly observed that without any caching at the network edge, the video bitrates are much lower than LFU and QoE-driven schemes with caching. Numerically, the average bitrate is 4.76Mbps, 9.83Mbps and 10.56Mbps for no caching, LFU and QoE-driven schemes respectively. This is due to the RTT of around 120ms between the UE and the content server, and the long latency and packet errors on the path (especially at RAN air interface) significantly reduce TCP throughput.

In Figure 4, we classify the video qualities based on their resolutions (as defined in Table 1), and analyze the breakdown of requested quality among all video segment requests. It is observed that the no caching scheme has very little opportunity to stream video at 4K quality (2.5%). In contrast, both LFU and QoE-driven schemes were able to stream 4K segments for more requests (37.6% and 41.5% respectively).

Two conclusions can be drawn from the analysis above. First, without caching content at the network edge, the user-experienced video quality is significantly worse (over 50% less in bitrate on average). Second, even simple caching

scheme like LFU can substantially improve the *average* quality of video streaming, which highlights the significance of content localization.

## 5.4 Results on Video Quality Switching

Besides average bitrate, frequent changes in video quality during playback can also degrade user QoE. In this subsection, we analyze 2 sets of metrics on this subject, where the results are presented in Table 3.

First, we analyze the total *number* of video quality switching events, as well as the total *levels* of video quality that is switched every time. The 9 video quality levels are defined as in Table 1. It is observed that while the 3 schemes have similar number of quality switching events, the number of switched quality levels under LFU is over 50% higher than the other two schemes. This is mainly because that LFU does not ensure the continuity of cached video segments, and considers each file to be isolated from each other.

Next, we specifically evaluate the number of quality switches that are across the 720p (1.5Mbps) threshold. The reason is that during our experiments, we (and many visitors) feel that if the video quality drops below the 720p mark, the degradation of QoE (e.g., obvious compressed artifacts) is especially significant. It is seen in Table 3 that such events are also more frequently experienced under LFU (135 times), which are 98.5% and 80% higher than under no caching (68 times) and QoE-driven (75 times) schemes respectively.

Combining the results in the two subsections above, we can conclude that a) content localization can substantially improve user QoE in terms of average video streaming quality by approximately 50%; and b) continuity of cached video segments is important in reducing video quality switching events and maintaining desirable user QoE.

## 6. CONCLUSION

In this paper, we presented an ICN-based edge caching framework which aims to improve user QoE in future 5G network environments for DASH-like video streaming applications. Supported by the Mobile Edge Computing paradigm (MEC), the system is able to harvest context information on both the content characteristics and RAN conditions. Such knowledge is then fed into the proposed QoE-driven caching and replacement algorithm at the caching manager side, which determines which are the best video segments / representations to be cached at the network edge. We conducted experiments on a real LTE-A wireless testbed with multiple users and concurrent DASH video streaming sessions. First, the results quantified the benefits in user QoE by caching content at the network edge. Second, they also showed that by performing context-aware cache replacement instead of traditional schemes such as LFU, user QoE can be further assured by minimizing frequent switching of video playback quality.

## 7. ACKNOWLEDGMENTS

This work is funded by EPSRC KCN project (EP/L026120/1) and HEFCE Innovation Gateway project.

The authors would also like to acknowledge the support of the University of Surrey's 5G Innovation Centre (5GIC) (<http://www.surrey.ac.uk/5gic>) members for this work.

## 8. REFERENCES

- [1] Named Data Networking. Available at <https://named-data.net/>.
- [2] W. K. Chai, N. Wang, I. Psaras, G. Pavlou, C. Wang, G. G. de Blas, F. J. Ramon-Salguero, L. Liang, S. Spirou, A. Beben, and E. Hadjioannou. CURLING: Content-ubiquitous resolution and delivery infrastructure for next-generation services. *IEEE Communications Magazine*, 49(3):112–120, Mar. 2011.
- [3] G. Chandrasekaran, N. Wang, and R. Tafazolli. Caching on the move: Towards D2D-based information centric networking for mobile content distribution. In *Proc. IEEE 40th Conf. Local Computer Networks (LCN)*, pages 312–320, Oct. 2015.
- [4] ETSI. Mobile Edge Computing: A key technology towards 5G (Whitepaper). Available at [http://www.etsi.org/images/files/ETSIWhitePapers/etsi\\_wp11\\_mec\\_a\\_key\\_technology\\_towards\\_5g.pdf](http://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp11_mec_a_key_technology_towards_5g.pdf).
- [5] J. O. Fajardo, I. Taboada, and F. Liberal. Improving content delivery efficiency through multi-layer mobile edge adaptation. *IEEE Network*, 29(6):40–46, Nov. 2015.
- [6] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard. Networking named content. In *Proc. CoNEXT'09*, pages 1–12, New York, NY, USA, 2009. ACM.
- [7] P. Jokela, A. Zahemszky, C. Esteve Rothenberg, S. Arianfar, and P. Nikander. LIPSIN: Line speed publish/subscribe inter-networking. *SIGCOMM Comput. Commun. Rev.*, 39(4):195–206, Aug. 2009.
- [8] G. Kamel, N. Wang, V. Vassilakis, Z. Sun, P. Navaratnam, C. Wang, L. Dong, and R. Tafazolli. CAINE: a context-aware information-centric network ecosystem. *IEEE Communications Magazine*, 53(8):176–183, Aug. 2015.
- [9] S. Lederer, C. Mueller, C. Timmerer, and H. Hellwagner. Adaptive multimedia streaming in information-centric networks. *IEEE Network*, 28(6):91–96, Nov. 2014.
- [10] 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 Journal on Selected Areas in Communications*, 32(4):719–733, Apr. 2014.
- [11] D. J. Vergados, A. Michalas, A. Sgora, D. D. Vergados, and P. Chatzimisios. FDASH: A fuzzy-based MPEG-DASH adaptation algorithm. *IEEE Systems Journal*, 10(2):859–868, June 2016.
- [12] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. M. Leung. Cache in the air: exploiting content caching and delivery techniques for 5G systems. *IEEE Communications Magazine*, 52(2):131–139, Feb. 2014.
- [13] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos. A survey of information-centric networking research. *IEEE Communications Surveys Tutorials*, 16(2):1024–1049, 2014.
- [14] M. Zhao, X. Gong, J. Liang, W. Wang, X. Que, and S. Cheng. QoE-driven cross-layer optimization for wireless dynamic adaptive streaming of scalable videos over HTTP. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(3):451–465, Mar. 2015.