QoE-Driven Interest-Based Video Caching and Adaptation at 5G Mobile Edge Network

Chih-Ho Hsu, Yao Chiang, Yu-Hsiang Chao and Hung-Yu Wei

Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan

***Abstract* —With the emerging demand of video requests in mobile networks, Multi-access Edge Computing (MEC) techniques were implemented in many of the 5G network architecture designs. Also, as social networking serves as a huge role in most of the mobile applications, utilizing the social influence factors into caching mechanism would provide better networking services. In this paper, we proposed a video content caching framework intended to provide non-delay playback services for future requests of the same video content. In this way, the users in the same Radio Access Network (RAN) that shares the same base station having a MEC Server and cache storages should benefit from the caching mechanism to significantly reduce transmission overhead and optimize caching resources while maintaining the best quality of experience (QoE) for video viewing as possible. Finally, the experiments show the proposed caching mechanism method offers much better performance than traditional random, most popular and least frequent used (LFU) caching algorithms on the average hit ratio and QoE.**

Index Terms—multi-access edge computing, quality of experience, mobile social network, mobile video delivery.

BACKGROUND

With the development of mobile devices and social media, the network traffic has grown exponentially in recent year and it is expected to increase by 40-fold over the next five years, as a white paper of Cisco System notes [1]. Also, the video traffic will account for 82% of all the Internet traffic by 2020. These mobile video stream and social network traffic will exert a significant burden on the backhaul network. Facing this, user's QoE will sharply decline due to the traffic congestion.

To cope with a large amount of mobile video requests and ensure user's QoE under limited backhaul capacity, proposed by ETSI, Multi-access Edge Computing (MEC) has become one promising solution [2] in which the video contents are localized at the network edges within the radio access network (RAN). In this way, the delivery delay can be significantly reduced because it usually brings a large delay to provide video contents through content delivery network (CDN) or remote cloud services, which will end up alleviating the traffic congestion on the backhaul network. On the other side, there are several studies focusing on the Adaptive Bit Rate (ABR) technique to make it better for caching in MEC to enhance the QoE on the user end [3], [4].

Since the capacity of cache nodes is limited, accurate content popularity prediction is important to effectually utilize the cache capacity. From another point of view, since massive users tend to be influenced by the trends in social community and mainstream media, the performance of content caching will become more effective if we could predict the popularity of contents in the social community beforehand. The interests of different users are utilized when developing the caching strategy [5], [6]. In addition, the contact frequency between users is considered when determining the social relationship of users [7], [8].

However, the mentioned works are all based on the assumption that the popularity distribution of the contents is known as the prior information such as the Zipf distribution, which may be less accurate within small population region such as school [9], office, where members may have a closer connection with each other and are willing to share interesting content with their acquaintances.

To address the challenging problem mentioned above, we determine to focus on the similarity and relationship between the users within the small population region when considering the influence of social community. We proposed a framework of the MEC mobile network that utilizes the caching mechanism and social influence to provide a better method to cache video contents.

Also, in order to achieve a better quality of experience with limited bandwidth capacity, we aimed at providing non-delay video playback for every subsequent requests of the same video in the same Radio Access Network (RAN) by caching portion of the video. In this way, as long as the video is first requested by a user and then cached in the MEC server, the other users will benefit from the caching results by retrieving data from the base station without further transmission overhead. This would save much of the networking resources and caching storages to provide more caching options for more videos, which will end up increasing user's QoE significantly. Specifically, our contributions in this paper can be summarized as follows:

1) We investigate the interest of users in the edge community and propose a novel algorithm to proactively cache the videos based on MEC architecture and social influence.
2) We consider two kinds of caching strategy to deal with different scenarios at the same time, instead of caching contents using a monotonous method, which focuses on the physical constraints in present and the popularity of contents, respectively.
3) A transcoding-based video delivery method is designed for providing non-delay playback by taking the user's varying wireless channel capacity and backhaul bandwidth into account, which could increase user's QoE significantly.

The rest of this work is organized as follows. First, we introduce the detail of our system and the proposed caching algorithm toward maximizing the QoE. In the end, a comprehensive experiment is conducted to verify our assumption about the social influence and the performance of the proposed caching mechanism.

PROPOSED METHOD

The structure of this section is presented as the following: First, subsection A gives an overview of our scenario, including the architecture of our MEC caching server and the way we qualify the influence of users in the social community. Subsection B formulates our objective in terms of QoE and physical constraint. Also, our approach to deal with these optimization problem is discussed in subsection B. Finally, Subsection C elaborate the proposed caching mechanism with its design of transcoding and caching lifetime.

A. System Framework

This subsection presents how content caching is performed in our proposed system architecture, and implemented frameworks (as shown in Fig. 1) in this system are twofold.

1) MEC Server: The MEC Server is located at the base station (BS), comprising of three parts to perform the caching process:

Cache Engine, Cache Storages and Cache Agent. The Cache Engine is responsible for the execution of the proposed algorithm which is confined to our caching policies, and so that we may achieve non-delay video playback for all users within the same RAN. The Cache Storages are two storages that can store cached video contents with different lifetime. Upon the decision made from the Cache Engine, a portion of the video will be cached in the Initial Cache Storage (Cache I). By storing this calculated portion of the video, we can guarantee that even if the rest of the video should be downloaded from the cloud again from the backhaul network, we can still transmit the whole video content seamlessly through the downlink to the user end. When a video playback request is made by a user, the first portion of that video will be retrieved from Cache I and sent through downlink by the Cache Agent, and meanwhile, the rest of the video content would be requested through the backhaul. While the first portion of the video was completely transmitted, the rest of the video content requested would be sent to the Cache Agent on time to catch up with the video streaming to reduce significant delays. As for the second cache storage, i.e. Temporary Storage (Cache II), it is used when massive identical requests are sent from the user end. For example, if an instance news video became very popular in a short period of time, and many users requested the same video at that moment, instead of retrieving data from the backhaul repeatedly, Cache II would cache the rest of the video content at the base station so that most of the delays due to redundant backhaul transmissions can be reduced. Since the purpose of this storage is for abrupt requests, the lifetime of the contents would be much smaller than that of Cache I, and they would be released after timeout. In this way, the total capacity of the Cache Storages can be better utilized. Once the video contents were retrieved from the storages, Cache Agent would process transcoding to meet the current downlink wireless network conditions. Our goal is to provide as higher Quality of Experience (QoE) to the user as possible while the non-delay playback can still be fulfilled.

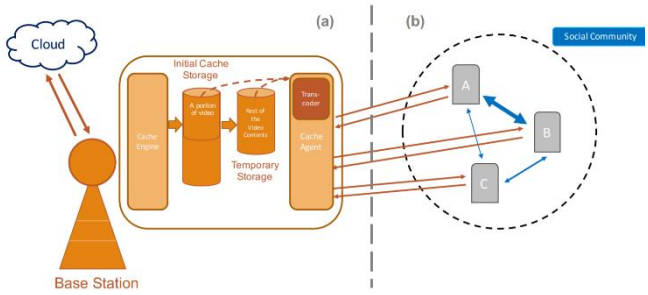

Fig. 1 System architecture composed of two parts: (a) Backhaul with MEC Server located at the base station and (b) Fronthaul with Social Community defined within the same RAN.

2) Social Community: In this paper, we intend to parameterize the social influence of each user so that when a user request for a video, the MEC Server would know how influential was this user and take it into consideration. Without generality, we assume that whenever a user with larger social influence made a request, the probability of other users requesting the same video in the same social community, i.e. the same RAN in this case, would be larger, and therefore the priority for this video content to be cache at the base station should be higher. However, the question is that we could barely learn the actual influence of each user with limited information.

To qualify the social influence, we first gives that there are a set of users U = $\{u_1, u_2, ..., u_k\}$ in the social community, where ul $\subseteq$ U denotes the l-th user , and a library of video F = $\{f_1, f_2, ..., f_L\}$ with $\{V_1, V_2 \ldots, V_L\}$, where $f_1 \subseteq$ F denotes the l-th video and $V_1 \subseteq$ F denotes the storage size of the l-th video. After that, we further assume the interest similarity between users could imply their strength of social relationship. Then, we consider the interest similarity $I_{a,b}$ of two users with item-item collaborative filtering (i.e. the sum of video contents in the watching history of both users divide by the length of watching history for both users), which can be expressed as:

$$I_{a,b} = \frac{\sum_{j=1}^{L} \theta_a^{(j)} \cdot \theta_b^{(j)}}{\sqrt{\sum_{j=1}^{L} \left(\theta_a^{(j)}\right)^2} \sqrt{\sum_{j=1}^{L} \left(\theta_b^{(j)}\right)^2}} \quad (1)$$

Where L is the number of the videos and $\theta_a^{(j)} \in \{0,1\}$ is an indicator function to signify whether user 'a' has seen the video j or not. For example, $\theta_a^{(j)}$=1 if the user 'a' has seen the video j, otherwise, $\theta_a^{(j)}$=0.

Finally, we define the social influence S of a user at time t within the same community as eigenvector centrality in the weight network, which can be calculated as follow:

$$S_a^{(t)} = \sum_{b=1}^{k} I_{a,b} S_b^{(t-1)} \quad (2)$$

Where k is the number of users within the community and $S_b^{(t-1)}$ is the social influence S of user b at time t-1.Note that the value of the social influence S of each user should be recalculated in every time slot.

B. Problem Formulation

To formulate the problem of caching for each of the cache storages, we took different factors into consideration since they were intended to serve for different purposes. For Cache I, we should consider storing as much of the video content as possible while maintaining the non-delay playback streaming of videos. For Cache II, we should consider the potential surge in popularity of the video in the impending future to decide whether we should also cache rest of the video to reduce the burden of redundant identical requests. Therefore, the problems are formulated as the following:

1) Cache I: To determine the optimal portion ρ of the video j should be cached in Cache I that can maximize the QoE of users, we consider the optimization problem (3), which means the sum of QoE of total requests for videos within n time slots. Note that we assume each user can only watch one video at the same time and the parameters are defined as in Table I.

$$max \sum_{t=0}^{N} \sum_{i=0}^{k} \sum_{j=0}^{L} R_{i,j}^{(t)} Q_i(\rho_j)$$

Subject to:

$$C1: \sum_{j=0}^{L} \rho_j V_j \leqq C_{edge}, \forall j \in \mathrm{N}$$

$$C2: b_i \leqq BW_i, \forall i \in \mathrm{N}$$

$$C3: \rho_j \frac{1}{p} \leqq (1-\rho_j)\left(\frac{1}{Th_B} + \frac{1}{p}\right), \forall j \in \mathrm{N}$$

$$C4: 0 \leqq \rho_j \leqq 1, \forall j \in \mathrm{N} \quad (3)$$

In the above equation (3), Q_i is the QoE function of user i and $R_{i,j}^{(t)} \in \{0,1\}$ is an indicator function denoting whether user i request for video j at time t. For instances, $R_{i,j}^{(t)}$=1 if the user I request for video j at time t. Otherwise, $R_{i,j}^{(t)}$=0. As pointed out by

several research works in DASH [3] [4] [9], the main factors that affect the QoE in dynamic adaptive video streaming are video quality, video stalling, bitrate switching rate and initial buffering time. Straightforwardly, as the videos are streamed with high resolution, the QoE will increases. While it might become negatively if bitrate switching rate is high. Moreover, long initial buffering time and stalling time will also negatively impact the QoE. In this paper, we adopt the definition of QoE proposed by [10].

TABLE I
NOTATION LIST

Notation	Definition
$I_{a,b}$	The interest similarity between user a and user b
${\theta_i}^{(j)}$	Indicator denoting whether user i has seen the video j
${S_a}^{(t)}$	Social influence of user a at time slot t
$Q_{i,j}$	The QoE function of the user i to the video j
C_{edge}	The cache storage size of the MEC server
ρ_j	The portion of the video j be cached in cache I
${R_{i,j}}^{(t)}$	Indicator denoting if user i request for video j at time t
b_i	Bitrate that the base station transmit to user i
BW_i	The wireless downlink channel capacity of user i
Th_B	Available throughput of backhaul network
p	Processing rate for video transcoding in MEC server
k	The number of users in the social community
L	The number of videos in the video library

Besides, constraint C1 describes that the total size of all stored portions of all video requests should not exceed the actual cache storage size. To deliver the video smoothly without delay, we should choose the proper bitrate version of the video to transmit according to C2, and so that the chosen bitrate should be less than the actual downlink bandwidth of the wireless network. Further, in order to seamlessly transmit the rest of the video from the backhaul to fronthaul, constraint C3 describes the total time of the stored contents in Cache I to be processed by Cache Agent shouldn't be greater than the time of the rest of the contents to be fetched from the backhaul and also be processed by Cache Agent. If the equation holds, it means the first bit of the rest of the portions would be ready at the transmitter end waiting to be sent to the user when the last bit of the stored contents were to be transmitted. Otherwise, due to the limited storage size (i.e constraint 1), the insufficient caching portion of the video makes it failed to fulfil the smooth transmission. Therefore, our goal is to do our best to prevent the inequality from happening by optimizing the storage utilization. The last constraint C4 is simply a notation of ρ being a value between 0 and 1 while ρ = 1 representing the whole video.

However, we have no way of knowing what requests for which videos will come in the next few time slots. Instead, by using heuristic method, we transform the original objective function (3) into (4).

$$max \sum_{i=0}^{k} \sum_{j=0}^{L} {S_i}^{(t)} {R_{i,j}}^{(t)} Q_{i,j}(\rho_j) \tag{4}$$

Which means maximizing the sum of QoE of total requests at given time t while taking the social influence of each requesting user into account. As mentioned in the previous section, we assume that when a user with larger social influence made a request, the probability of other users requesting the same video in the same social community would be larger. Thus, the decision of our caching engine might be more statistically accurate in the next few time slots if we weight each request with the social influence of its requesting user at present.

To further simplify the problem, by adopting the concept of DASH, we assume that the bitrate transmitted to the user is equal to its wireless downlink channel capacity (i.e $b_i \leqq BW_i$). Then, the original problem becomes a linear programming problem.

To obtain the optimal solution to the subproblem (8), we consider the feasible solution zone as a polytope in space and the vertex of it are candidates of the optimal solution. First, we randomly pick up a vertex as a pivot. Then, we compare the value of the overall QoE at the pivot with that of its adjacent vertex. If the value of the overall QoE at the pivot is the maximum within its neighborhood, then we say that the set of ρ at the pivot is the optimal solution and that the iteration should be terminated. Otherwise, we pick up the vertex that has the greatest value of overall QoE in the neighborhood as the new pivot. Repeat the process until we find out a vertex that its value of the overall QoE is the maximum within its neighborhood. Finally, we get the optimal set of ρ. The workflow of the process is depicted in Fig. 2.

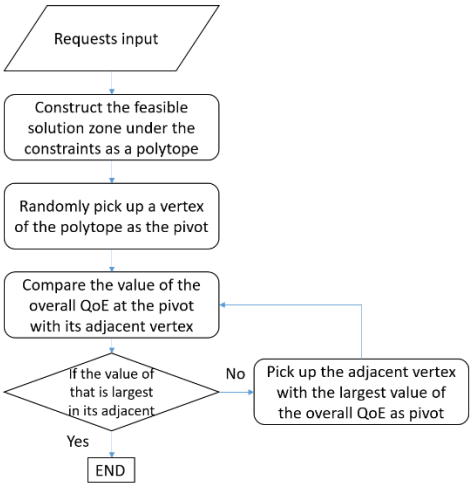

Fig. 2 Workflow of determining the optimal set of ρ at cache I.

2) Cache II: To determine whether we should also cache rest of the video, we consider the event that massive identical requests being made in a short period of time, whose traffic might exceed the available throughput of backhaul network in the impending future. However, to forecast growth in popularity of a video, we need to take the effect of social community into account. We assume that users with similar preference for videos tend to watch the same video. Thus, the growing in popularity of a video can be considered the as the number of users with similar interest in the community, which can be calculated based on the definition we mention above (1). Given the set of source user that request for video j at time t, the probability that an arbitrarily user i affected by the other users with similar interest request for the video j can be deduced as follow:

$$P_i = 1 - \prod_{y=0}^{k} \left(1 - {R_{i,j}}^{(t)} I_{i,y}\right) \tag{5}$$

Finally, by summing the possibility that a user request the video j within the community (9), the decision of caching the rest of the video contents in Cache II should be made when:

$$b_j \sum_{i=0}^{k} P_i \geqq Th_B \tag{6}$$

Where b_j is the original bitrate of video j obtained from the content server, and Th_B is the available backhaul throughput at time t. In this way, whenever the repeated requests create redundant traffic is likely to exceed the available bandwidth capacity of the backhaul

network in the impeding future, the rest of the video will be cached in Cache II, reducing the transmission delay caused by the possible congested network link.

C. CACHING MECHANISM

The caching mechanism of the purposed method can be divided into two steps: first time caching of the video and future requests. For the first time when the video was requested from the user, Cache Engine would first perform the algorithm to decide how much (i.e ρ) of the video should be cached in Cache I. Further, if equation (9) was satisfied for this scenario, then rest of the video contents should be cached in Cache II. Otherwise, they would be discarded after this request was successfully sent to the user. Then after caching, the entire origin video would be delivered to the Cache Agent to dynamically perform transcoding so that the transmitted bitrate can meet the network conditions for the wireless downlink. When a future request was made, two subsequent scenarios should be considered:

1) Contents were still stored in Cache II: In this scenario, the Cache Agent should retrieve the video data from Cache I and Cache II directly, then transmit it through downlink after transcoding was performed.

2) Lifetime of the contents in Cache II had expired: There would be two different situations for future requests when the lifetime of rest of the requested video contents had expired. The two situations are: (a) rest of the contents needed not to be stored and (b) needed to be stored in Cache II. If rest of the contents did not need to be stored Cache II according to (9), then after the first portion of contents were successfully retrieved from Cache I, the rest of the contents should be directly delivered. If rest of the contents needed to be stored in Cache II again, then the caching process should be performed just as first time request. Then after every requests, the caching algorithm should decide whether the data cached in both storages should be cleaned.

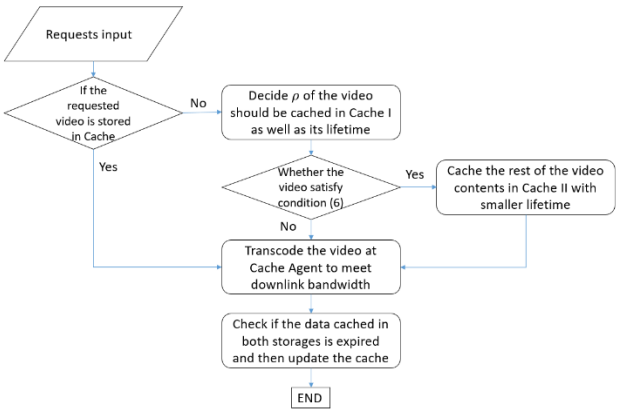

Fig. 3. The overall workflow of the proposed caching mechanism

The overall workflow of our caching mechanism is shown in Fig. 3. First, we collect and analyze all the requests at the beginning of the time slot. If the requested videos are still stored in the caching server, the cached video will be directly transmitted to the user after transcoding. Otherwise, we perform the optimization process of subproblem (8) to determine the optimal portion of each requested video that should be cached and an appropriate lifetime. After that, we check the requested video if it satisfies the equation (9). If equation holds, the caching server will cache the rest of the video in Cache II with relatively small lifetime. Then, we again transcode the video to maximize each user's QoE based on wireless channel capacity. Finally, we clean the data stored in the cache I and II if their lifetime are expired and update the cache list. By performing the algorithm again, we can optimize the cache storage utilization after every requests were made and that we would not be potentially spending more time on caching when a new request is to come.

RESULTS

In this section, we evaluate the performance of the proposed caching algorithm compared with other three traditional caching algorithms: the Most Popular Content (MP) caching, the Random caching and least frequent used (LFU) caching. In the MP caching, the caching server stores the most popular contents according to its viewing count. In the Random caching, the caching server store each requested video and discards the cached video at random if its cache space is full. In the LFU caching, the caching server also stores each requested video but it discard the least frequent used video in cache list if its cache space is full. To further validate the performance of the proposed approach, we utilized dataset of real world streaming traffic in the UMass campus to conduct our simulation. Collected from YouTube network traces by [11], the datasets contain 16337 users request for 86402 different videos within 611968 requests.

In the simulation setup, we set the interval of a time slot to be one hour, run trip time (RTT) of the fronthaul network to be 15ms and RTT of the backhaul network to be 50ms. Furthermore, the lifetime for videos in cache I and cache II are 5 and 2 time slots, respectively. The bandwidth of backhaul network is assumed to be 100Mbps. The specifications of initial buffering time and stalling time is reported in [12]. Also, the video length is set to be normal distribution with an average of 11.7 and a standard deviation of 4 minutes, which is the average video length on Youtube in 2018. Besides, the resolution of each video is set as 1080p with bitrate of 1MB/s. Finally, we adopt the hit ratio, initial latency and QoE as the performance metric. Note that the QoE is calculated by weighted sum of bitrate, initial latency and stalling time in our experiment.

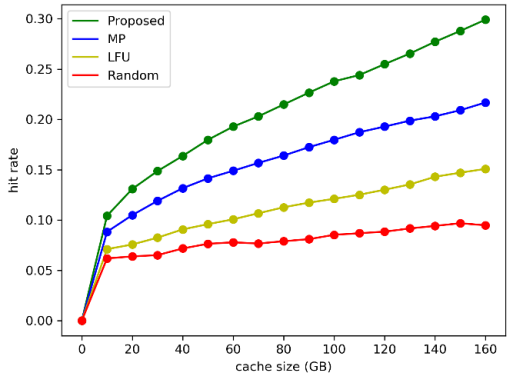

Fig. 4. The hit ratio vs. the cache size (in GB)

Fig. 4 shows the simulation results of the hit ratio with respect to the caching storage size at the MEC server. It is shown that, generally, the performance of the proposed is higher than MP, LFU and random caching. We observe that the hit rate of the proposed method is much higher than the traditional method. This could result from 2 perspectives: 1) We consider the effect of social community in terms of interest, which can extract more hidden information from the ordinary traffic that the traditional method. 2) The way we cache the video is to store partial of the video as long as it can satisfy the condition of non- delay playback instead of caching entire video, which will be more obvious as the cache size grows larger.

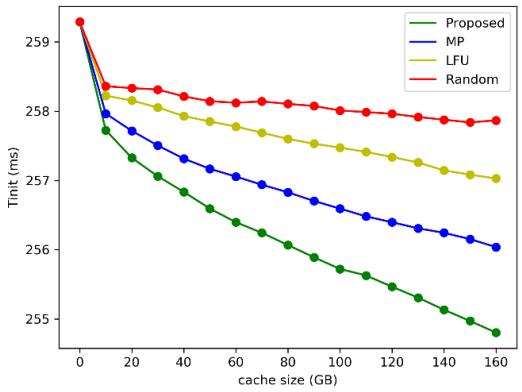

Fig. 5. The initial latency (in ms) vs. the cache size (in GB)

The experimental result of initial latency respect to the caching size of the MEC server is presented in Fig. 5. Observe that the initial latency of the proposed method is significantly low compared with the other traditional methods. This is mainly because caching at the edge server can provide much less latency in data transmission than directly retrieve videos from the contents servers through the backhaul network, and the hit ratio of the proposed method at the MEC caching server is the highest among 4 methods.

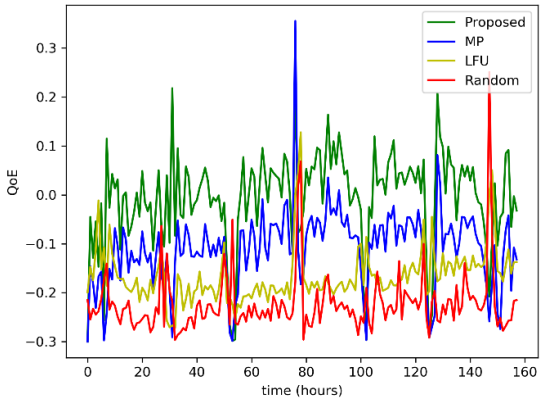

Fig. 6. The QoE vs. time slot (in hour) at cache size = 100 GB

Fig. 6 illustrates the experiment result of the QoE respect to the time slot. We can learn from the chart that the QoE is time dependent and will be degraded significantly whenever there is a surge in traffic. Though there are some fluctuations in the performance of the proposed method in terms of QoE due to the dynamic nature of social community, the performance of the proposed method is higher than MP, random and LFU in general. We observs that the MP method might not perform well in the first few time slots compared with the proposed method. This could stem from lack of samples for the MP method to capture the distribution of video's popularity while the proposed method can determine the video's popularity in the next few time slots based on the influence of social community even if their view counts are not the highest at the beginning. Also, note that the performance of the LFU method is continuously getting better as the time pass by. This reflects the nature that it discards the least frequent requested videos in cache at the end of every time slot and will eventually approach the performance of MP.

In conclusion, the simulation results show that the proposed method offers much better performance, in terms of average hit ratio and QoE, than traditional random, most popular and least frequent used(LFU) caching algorithms, which might imply that the effect of social community is not negligible and caching at the MEC server can improve user's QoE.

REFERENCES

[1] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013-2018, Cisco, San Jose, CA, USA, 2012. [Online]. Available:http://www.cisco.com/c/en/us/solutions/collateral/serviceprovider/visual-networking-index-vni/white_paper_c11-520862.pdf

[2] ETSI GS Multi-access Edge Computing (MEC) Framework and Reference Architecture 003 V2.1.1 (2019-01) [Online].Available: https://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/02.01.01_60/gs_MEC003v020101p.pdf

[3] W. Zhang, Y. Wen, Z. Chen, and A. Khisti, "Qoe-driven cache management for http adaptive bit rate streaming over wireless networks," IEEE Trans. Multimedia, vol. 15, no. 6, Oct. 2013.

[4] C. Ge, N. Wang, S. Skillman, G. Foster, and Y. Cao, "QoE-driven DASH video caching and adaptation at 5G mobile edge," in ACMICN'16Proceedings of the 3rd ACM Conference on Information-Centric Networking, pp. 237–242, Sep. 2016.

[5] N. Zhang, J. Guan, C. Xu and H. Zhang, "A dynamic social content caching under user mobility pattern," 2014 International Wireless Communications and Mobile Computing Conference (IWCMC), Nicosia, pp. 1136-1141, 2014.

[6] Y. Wang, M. Ding, Z. Chen and L. Luo, "Caching Placement with Recommendation Systems for Cache-Enabled Mobile Social Networks," in IEEE Communications Letters, vol. 21, no. 10, pp. 2266-2269, Oct. 2017.

[7] Zhou, Hao et al. "A Physical-social-based Group Utility Maximization framework for Cooperative Caching in Mobile Networks." 2018 10th International Conference on Wireless Communications and Signal Processing (WCSP), pp. 1-7, Oct. 2018.

[8] B. Wang, Y. Sun, S. Li, Q. Cao, Y. Chen and J. Xu, "Hierarchical Matching with Peer Effect for Latency-Aware Caching in Social IoT," 2018 IEEE International Conference on Smart Internet of Things (SmartIoT), Xi'an, pp. 255-262, 2018.

[9] H. Ahlehagh and S. Dey, "Video caching in Radio Access Network: Impact on delay and capacity," 2012 IEEE Wireless Communications and Networking Conference (WCNC), Shanghai, pp. 2276-2281, 2012/

[10] A. Mehrabi, M. Siekkinen and A. Ylä-Jääski, "QoE-Traffic Optimization Through Collaborative dge Caching in Adaptive Mobile Video Streaming," in IEEE Access, vol. 6, pp. 52261-52276, 2018.

[11] YouTube Traces From the Campus Network [Online].Available: http://traces.cs.umass.edu/index.php/Network/Network

[12] M. Shefkiu, S. Cakaj and A. Maraj, "Quality of experience (QoE) improvement by video caching implementation," *2017 South Eastern European Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*, Kastoria, pp. 1-5, 2017.

[13] 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 Commun. Mag., Feb. 2014.

[14] X. Xu, J. Liu, and X. Tao, "Mobile edge computing enhanced adaptive bitrate video delivery with joint cache and radio resource allocation," IEEE Access Special Section on Mobile Edge Computing, vol. 5, Aug. 2017.

[15] H. A. Pedersen and S. Dey, "Enhancing mobile video capacity and quality using rate adaptation, RAN caching and processing," IEEE/ACM Trans. Networking, vol. 24, no. 2, Apr. 2016.

[16] M. S. Elbamby, M. Bennis, and W. Saad, "Proactive edge computing in latency-constrained fog networks," in European Conference on Networks and Communications (EuCNC), Jun. 2017.

[17] Y. Zhou, F. R. Yu, J. Chen, and Y. Kuo, "Resource allocation for information-centric virtualized heterogeneous networks with in-network caching and mobile edge computing," IEEE Trans. Veh. Technol., vol. 66, no. 12, Dec. 2017.

[18] X. Wang, T. T. Kwon, Y. Choi, H. Wang, J. Liu, and S. Fraser, "Cloudassisted adaptive video streaming and social-aware video prefetching for mobile users," IEEE Wireless Commun., vol. 20, Jul. 2013.