Consolidated Caching with Cache Splitting and Trans-rating in Mobile Edge Computing Networks

Shashwat Kumar and Antony Franklin A.
Department of Computer Science and Engineering
Indian Institute of Technology Hyderabad, India
Email: {cs15resch11011, antony.franklin}@iith.ac.in

***Abstract*—The contribution of video content in mobile data traffic is predicted to increase over the coming years. If the network is not managed efficiently, this huge amount of data will increase the congestion in the cellular network. Caching at the edge of the cellular network is one of the prominent solutions to mitigate this issue. By deploying Mobile Edge Computing (MEC) at eNodeB (eNB), distributed edge caching can be realized in the cellular network. In this work, we propose a consolidated caching scheme with cache splitting and trans-rating in MEC network. Cache consolidation uses the collaborative capability of MEC servers to avoid replication of videos in the MEC cache network thus increasing the caching capacity. Without replication, access delay may increase. To maintain low access delay, we split the cache storage into two logical parts; full video cache, and initial video cache. Full video cache stores the complete videos without any replication in the MEC cache network and initial video cache stores only initial segments of the videos. Initial segments of the videos are replicated in the cache network to reduce the access delay. Extensive simulations results show improvement in initial access delay, cache hit ratio, and external traffic load compared to the existing solutions.**

I. Introduction

According to Cisco mobile data traffic forecast, global data traffic reached 7.2 Exabytes per month at the end of 2016 and expected to touch 49 Exabytes per month by 2021 [1]. The video traffic is accounted for 60% of the total smartphone data traffic in 2016 and predicted to increase up to 78% by 2021. There is immense pressure on telecom operators to scale up their network capacity to cope up with this vast amount of traffic. The video traffic accessed over the cellular network is intermittent and usually very high during the peak hours compared to non-peak hours. This sporadic behavior of video traffic makes efficient network management difficult for the telecom operators. Edge caching is a prominent solution to resolve this problem [2]. By caching at the edge of the network (i.e., base station), as the content is stored near the users, end-to-end delay and external traffic load can be reduced significantly. External traffic load is the amount of data that need to be fetched from the Internet to fulfill the user requests.

Mobile Edge Computing (MEC) [2] is a new paradigm that brings the cloud infrastructure to the edge of the cellular network. MEC servers can be deployed at eNBs in LTE-A network enabling the deployment of services, which requires computation and storage, at the edge of the network. MEC servers can be used to deploy content caching services for video traffic. By caching popular videos on the MEC servers access delay and the external backhaul traffic can be reduced. Different users might request different bit-rate versions of a video based on their devices' capabilities, network conditions, and preferences. For example, users with a fast network connection can get the video in high-resolution without any delay while the users with the low-bandwidth connection may get high-quality videos with high access delay which leads to rebuffering. Adaptive Bit Rate (ABR) streaming techniques [3] have been used to improve the users' quality of experience for video streaming. In ABR streaming, the quality (bit-rate) of the streaming video is adjusted according to the capability of the user device, network connection speed, and user preference. Existing video caching techniques often treat each users' request equally and independently, whereby each bit-rate version of the video is offered as a disjoint stream (data file) to the user. As the storage capacity of the MEC servers is limited, it is not efficient to store different bit rate versions of the same video on the edge cache. Higher bit-rate versions of a video can be trans-rated to lower bit rate version in real time using the processing power at the MEC servers. The MEC architecture consists of distributed MEC servers that can collaborate among them in real time. The access delay for fetching the content from the cache on MEC servers is low compared to the access delay for fetching the content from the origin server over the Internet.

Using these capabilities of MEC architecture, we propose the following two-fold solution to improve the access delay, cache hit ratio, and external traffic load.

- *Cache Consolidation:* As the MEC servers can collaborate with each other to share the stored content, it is unnecessary to replicate the same content over different MEC servers. One copy of the content can be shared by all the MEC servers. Thus, with cache consolidation more videos can be cached in the MEC network.
- *Cache Splitting:* With cache consolidation, the video is transferred from other MEC server (if video is not cached on the serving MEC server) which leads to significant increase in access delay. To reduce access delay, we split the cache into two logical parts. First part is used to store the complete videos without any replication and second part used to store the initial segments of the videos. Initial segments of the videos are replicated in the MEC cache network to reduce the access delay.

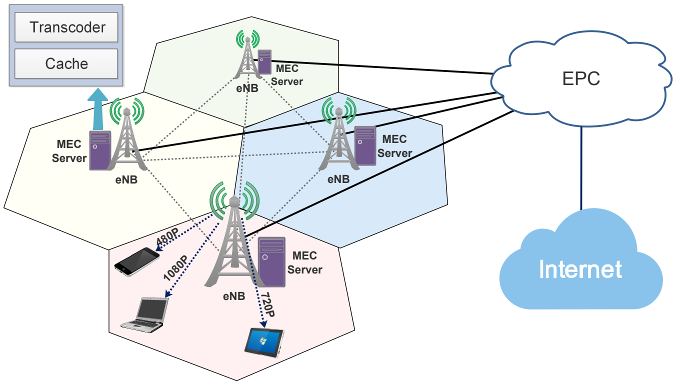

Fig. 1: Video caching system architecture where the edge-cache is deployed on the MEC server located at eNBs.

The remainder of the paper is organized as follows. Section II provides the related work. Section III explains the system architecture of the edge caching using MEC servers. The proposed cache consolidation and cache splitting are discussed in Section IV. Section V evaluates the performance of the proposed solution and show the comparison with existing methods. We conclude the paper in Section VI.

II. Related Work

Content caching in wireless networks is explored in [4] and [5] to reduce the access delay. For collaborative caching on servers at Base Stations (BSs), backhaul link between BS has been utilized [6], [7]. In [8] and [9], caching and processing for multi-bitrate video streaming is proposed. However, they do not consider the collaborative scheme of multiple caching/processing servers. Furthermore, the proposed technique in CachePro [9] solves the optimization problem for every new request, that will result in re-directing large numbers of pre-scheduled requests. On the other hand, the heuristic solution in [10] requires the knowledge of the content popularity, which may be hard to estimate accurately in practice. JCCP in [11], propose a collaborative caching and trans-rating approach where MEC servers collaborate with each other to share the cached content and perform the trans-rating if the higher bit-rate version in available in the cache. In [12], a coordinated data assignment algorithm is proposed to minimize the network cost with respect to the pre-coding matrix and cache placement matrix in a Cloud-RAN (C-RAN). In [8] and [13], various techniques to transcode a video from higher bit-rate to lower bit-rate version are discussed. Compressed-domain based approaches, such as bit-rate reduction and spatial resolution reduction are the most favorable among these techniques.

III. System Architecture

As shown in Fig. 1, an MEC network consists of multiple MEC servers connected via backhaul links. Each MEC server is deployed side-by-side with the eNB in a cellular RAN, providing computation, storage, and networking capabilities to support context-aware and delay-sensitive applications near the users. Processing and storage capabilities of MEC servers are used for video trans-rating and caching. These MEC servers can collaborate to share their computing and storage resources. Based on received video request, the MEC server can serve the video content from its cache (if available) or download the content from the Internet and serve the user while caching the same content for future access. If a higher bit-rate version of the requested video is available in the cache then MEC server trans-rates the available video to requested lower bit-rate version to serve the user. Video trans-rating, i.e., compressing a higher bit-rate video to a lower bit-rate version, can be done by various techniques given in [8] and [13]. Video trans-rating is a computation-intensive task and the computation cost can be measured as consumed CPU cycles for trans-rating on the MEC server. Following are the possible events that might happen when a user requests for a video.

1) The video is obtained from the MEC cache of the connected eNB.
2) A higher bit-rate version of the video, from the cache of the connected eNB, is trans-rated to the desired bit-rate version and delivered to the user.
3) The video is retrieved from MEC cache of a neighboring eNB or the origin content server.
4) A higher bit-rate version of the video, from MEC cache of the neighboring eNB, is trans-rated using the co-located trans-coder and then transferred to the connected eNB.
5) Similar to (4), but the trans-rating is done at the MEC server of the connected eNB.
6) Video is not cached and needs to be downloaded from the content server over the Internet.

IV. Proposed Work

Most of the existing work on video caching, which are not ABR-aware, mainly rely on the store and transmit mechanism without any processing. Proposed solution tries to utilize both caching and processing capabilities at the MEC servers to satisfy the user requests for videos of different bit-rate versions. MEC servers can trans-rate a video to lower bit-rate, using its processing capabilities, to fulfill the user requests. If enough processing power is available to trans-rate a video from higher bit-rate version to a lower bit-rate version, there is no need to cache lower bit-rate video when a higher bit-rate version of the same video is already cached. We extend the collaborative caching paradigm by consolidating the cache. Using the trans-rating and MEC collaboration, we propose the following solutions to reduce the access delay and external traffic load:

A. Cache Consolidation

By exploiting the collaboration among the MEC servers, cache at the edge of the network can be consolidated. In a collaborative environment where MEC servers can share the data, there is no need to replicate the same video on a different MEC servers. Instead of replicating the same content on the MEC servers, requested content can be transferred from one MEC server to another. Through cache consolidation, more videos can be cached collectively on the MEC servers and thus a significant improvement in hit ratio and external traffic load can be achieved. Even though access delay among the MEC servers is very low compared to the delay between a content server and MEC servers, as cache consolidation does not have content replication it may lead to increase in the access delay (i.e. when a popular video is cached on an MEC server, it needs to be transferred each time a user requests for it on another MEC servers). To address such scenarios, we propose cache splitting as given below.

B. Cache Splitting

In video streaming, initial access delay depends on the time player takes to download the initial segments of the video. The user does not require to download the complete video to start watching a video, as soon as the video player buffers the initial segments of the video, it starts the playback and rest of the segments downloads over the time. Caching initial segments are sufficient to reduce the access delay of the video. It also leads to better hit ratio as more number of videos can be cached in the MEC servers compared to when only complete videos are cached. However, if all cache storage is used to cache only initial segments of the videos then, for each hit, rest of the video needs to be downloaded from the content server over the Internet. To balance the external traffic load and the access delay, we propose a logical splitting of cache storage. One part of the cache storage is used to cache complete videos and another part to cache only initial segments of the videos.

In cache splitting, it is critical to choose the level of the splitting of cache as reducing the delay and external traffic load are conflicting objectives. To thoroughly understand it, we theoretically analyze the access delay and external traffic load in the following sub-section.

C. Access Delay and Backhaul Load Analysis

Let V be the video library containing N videos. The size of the cache at each MEC server is C_s. With cache split at x $(0 \leq x \leq 1)$, allocated cache size to store the full videos and initial videos segments is xC_s and $(1-x)C_s$ respectively. Let the average video size be $b\ bytes$ and only a fraction (n, $0 < n < 1$) of all video segments need to be stored in the initial video cache. These initial fractions of the video are sufficient to start the video playback. The number of videos that can be stored in the full video cache is $C_f = \frac{xC_s}{b}$ and number of videos that can be stored in the initial video cache is $C_i = \frac{(1-x)C_s}{bn}$. The total number of videos in the cache is,

$$C_t = \frac{C_s(x(n-1)+1)}{bn} \tag{1}$$

Assuming a uniform popularity distribution, cache hit-ratio H can be given as,

$$H = \frac{C_s(x(n-1)+1)}{nbN} \tag{2}$$

Let d_m be the access delay if there is a miss and d_h is the access delay if there is a hit ($d_m > d_h$). Then the average access delay is given by,

$$D = Hd_h + (1-H)d_m \tag{3}$$

using the value of H from (2)

$$D = d_m + (d_h - d_m)\frac{C_s(x(n-1)+1)}{nbN} \tag{4}$$

As $(d_h - d_m) < 0\ and\ (n-1) \leq 0$, the value of access delay D will be minimum if $x = 0$. To minimize the access delay, all the cache storage should be allocated to store initial part of the videos.

The hit ratio for initial video cache is given by,

$$H_i = \frac{C_s(1-x)}{nbN} \tag{5}$$

So, the total external traffic load would be,

$$\begin{aligned} T &= H_i(1-n)b + (1-H)b \\ &= \frac{C_s(1-x)}{nbN}(1-n)b + (1 - \frac{C_s(x(n-1)+1)}{nbN})b \\ T &= b - \frac{C_s}{N} \end{aligned} \tag{6}$$

From 6, T does not depend on x when video requests follow a uniform distribution. So, for any division of the cache, value of T will be same for given values of b, C_s, and N.

In the real world, the video requests follow some popularity distribution instead of uniform distribution. Popular videos will be requested more number of times compared to non-popular videos. Here we use Zipfs' law for popularity distribution, which gives the probability of an incoming request for i^{th} popular video as,

$$p_i = \frac{i^{-\alpha}}{\sum_{j=0}^{N} j^{-\alpha}} \tag{7}$$

where α is Zipf parameter. When video requests follow Zipfs' popularity distribution, cache hit-ratio can be derived using the Cumulative Distribution Function (CDF) of the Zipf distribution given by,

$$CDF = \frac{H_{k,s}}{H_{N,s}} \tag{8}$$

where $H_{N,s}$ is the N^{th} generalized harmonic number.

Considering that most popular videos are cached. The cache hit-ratio H can be derived as,

$$H = \frac{\sum_{n=1}^{C_t} \frac{1}{n}}{\sum_{n=1}^{N} \frac{1}{n}}$$

The sum of the n^{th} harmonic number can be approximated by the integral $\int_1^n \frac{1}{x}dx$, whose value is $ln(n)$. So,

$$H = \frac{ln(C_t)}{ln(N)} \tag{9}$$

The hit ratio of initial video cache is given by,

$$H_i = \frac{\sum_{n=1}^{C_t} \frac{1}{n} - \sum_{n=1}^{C_f} \frac{1}{n}}{\sum_{n=1}^{N} \frac{1}{n}} = \frac{ln(C_t) - ln(C_f)}{ln(N)} \tag{10}$$

The overall access delay D is,

$$D = d_m - (d_m - d_h)\frac{ln(C_s) + ln(1 - x(1-n)) - ln(nb)}{ln(N)} \tag{11}$$

From (11), the access delay will be minimum (for $x = 0$) when all the cache storage is used for caching the initial segments of the videos. The total external traffic load T is given by,

$$\begin{aligned} T &= H_i(1-n)b + (1-H)b \\ T &= b - \frac{b}{ln(N)}[(1-n)ln(x) + nln(1 - x(1-n)) + \\ &\quad ln(C_s) - ln(b) - nln(n)] \end{aligned} \tag{12}$$

From (12), the total external traffic load will be minimum when $x = 1$. So, to minimize the external traffic load, all cache storage should be used to cache full videos.

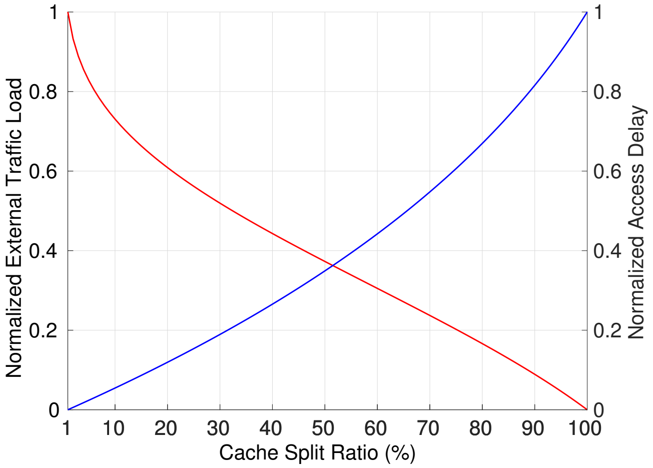

Fig. 2: Effect of cache split ratio on access delay and external traffic.

Fig. 2 shows the change in access delay and external traffic load with varying cache split. It is clear from the plot that the average access delay increases when we allocate more cache to store complete videos, as the number of videos that can be stored in the cache reduces and hence the hit-ratio. The external traffic load reduces when more storage is allocated to cache complete videos, as each cache hit in full video cache leads to zero external traffic. Choosing a proper cache split ratio is tricky as reducing access delay and external traffic load are conflicting objectives. Fig. 2 reflects that splitting the cache around 50% provides an equivalence point that balances access delay and external traffic load. The network operator may choose a suitable cache split ratio based on the allowable QoE constraints.

Algorithm 1 Caching and Replacement Algorithm

```
For each video request $v_l$ arriving at eNB $j$, proceed.
if $v_l \in C_j^F$ then serve the user from $C_j^F$.
else if $v_l \in C_j^I$ then serve the initial segments from $C_j^I$.
    if $v_h \in C_j^F$ and $P_j^* + p_l \le P_j$ then
        trans-rate the remaining video segments from $v_h$ to $v_l$ and serve the user.
    else if $v_l \in \cup_{j \neq k, k \in K} C_k^F$ then
        $f = C_k^F$ for $k \in K$ and $k \neq j$ s.t. $v_l \in C_k^F$
        fetch the remaining video segments from $C_k^F$ and serve the user.
    else if $v_h \in \cup_{j \neq k, k \in K} C_k^F$ and $P_k^* + p_l \le P_k$ then
        $f = \min_{k \neq j, k \in K} d_{jk}$ s.t. $v_h \in C_k^F$
        trans-rate the remaining video segments at eNB $f$ and serve the user.
    else
        fetch the video from origin server and cache in $C_j^F$ using LRU; remove $v_l$ from the $C_j^I$.
else if $v_h \in C_j^F$ and $P_j^* + p_l \le P_j$ then
    trans-rate the video $v_h$ to $v_l$ and serve the user.
else if $v_l \in \cup_{j \neq k, k \in K} C_k^F$ then
    $f = C_k^F$ for $k \in K$ and $k \neq j$ s.t. $v_l \in C_k^F$
    fetch the video from $C_k^F$, serve the user, and cache initial segments of $v_l$ in $C_f^I$ using LRU.
else if $v_h \in \cup_{j \neq k, k \in K} C_k^F$ then
    $f = \min_{k \neq j, k \in K} d_{jk}$ s.t. $v_h \in C_k^F$
    if $P_f^* + p_l \le P_f$ then
        trans-rate the video on MEC server at eNB $f$, serve the user, and cache initial segments of $v_l$ in $C_j^I$ using LRU.
    else if $P_j^* + p_l \le P_j$ then
        fetch the video $v_h$ from $C_f^F$, trans-rate video $v_h$ to $v_l$ on MEC server at eNB $j$ and serve the user, cache initial segments of $v_l$ in $C_j^I$ using LRU.
else if $v_l \in \cup_{j \neq k, k \in K} C_k^I$ then
    $f = C_k^I$ for $k \in K$ and $k \neq j$ s.t. $v_l \in C_k^I$
    fetch the initial segments of video from $C_f^I$, serve the user, and fetch the remaining segments from origin server and cache in $C_j^F$ using LRU.
else
    fetch the video from origin server and cache it on $C_j^F$ using LRU. Cache initial segments of replaced video from $C_j^F$ in $C_j^I$ using LRU.
```

D. Cache Placement and Replacement Algorithm

Replicating the complete video over the MEC network reduces the access delay, but less number of videos can be cached which directly affects the hit ratio. By caching the initial segments of the video, the access delay can be reduced with no extra cost in external traffic load as the complete video is already cached on one of the MEC servers. So, to reduce the access delay, initial segments of the videos are replicated in

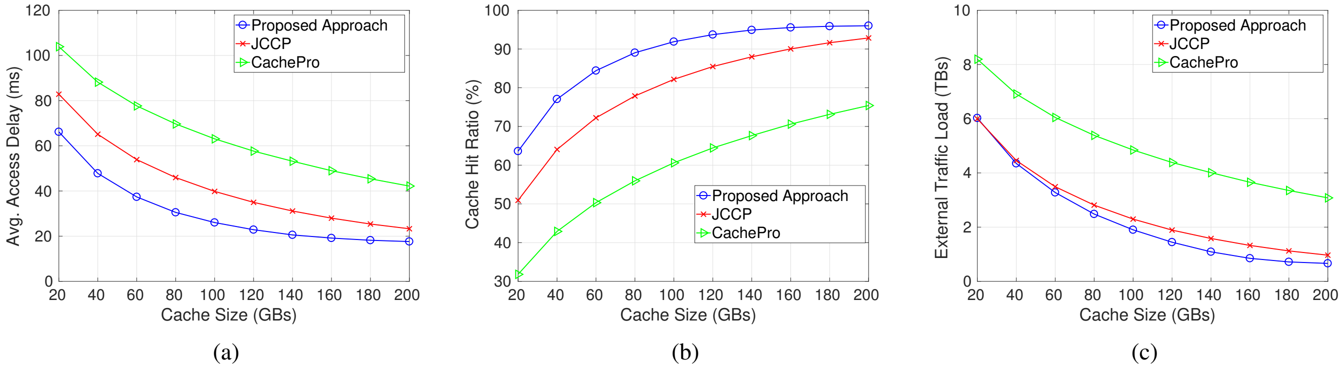

Fig. 3: Comparison of caching schemes for different cache size at each MEC server; cache split ratio $x = 75\%$; processing power at each MEC server $C_j = 50Mbps$.

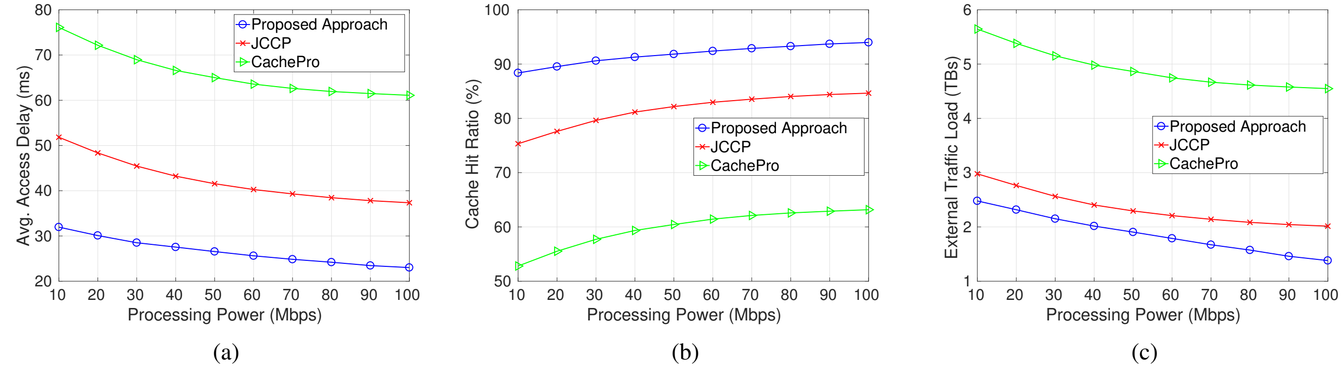

Fig. 4: Comparison of caching schemes for different processing power at each MEC server; cache split $x = 75\%$; Cache size at each MEC server $C_j = 100GB$.

the network which costs less towards storage and provides the same benefit regarding the access delay. Least Recently Used (LRU) cache replacement policy is used to replace the full videos in the cache. Initial segments of the replaced full videos are moved to the initial cache to keep the access delay low. Algorithm 1 shows our caching and replacement algorithm. In the algorithm, C_j^F and C_j^I represent the set of videos stored in full video cache and initial video cache respectively on MEC server at eNB j. Processing power and current load on MEC server at eNB j is given by P_j and P_j^*. p_l is the required processing power to trans-rate higher bit-rate video v_h to lower bit-rate version v_l. d_{jk} is the access delay between MEC server j and k. K is the set of all MEC servers.

V. Performance Evaluation

In this section, we evaluate the performance of the proposed scheme by varying cache size and processing power. In simulations, we consider an Urban Macro (UMa) cell model with four eNBs, each eNB using a transmission power of $46dbm$ and $20Mhz$ of channel bandwidth. Each eNB is serving 50 active users, which are uniformly distributed in the cell of $5km$ radius. An MEC server is deployed at each eNB, as shown in Fig. 1, to provide the caching and processing resources. We assume that the video library V consists of 2000 videos following a Zipf popularity distribution with exponent value of 0.8. Playtime of each video is 10 minutes and a video can be served in any of the 4 bit-rate variants (0.4 Mbps, 1.2 Mbps, 2.5 Mbps, and 5 Mbps, for 360p, 480p, 720p, and 1080p video resolutions, respectively). The sizes of different bit-rate variants of the video are 50 MB, 80 MB, 100 MB, and 150 MB. Each user generates video requests independently following the Poisson process with mean inter request interval of $8min$. The users request video following the Zipfs' popularity distribution. To determine the requested bit-rate variant, we use the UMa path loss model with Line of Sight (LOS) conditions as specified in 3GPP TR36.814 V9.2.0 [14]. Using the channel model we calculate the available Down-Link (DL) throughput at UE, and based on the available throughput UE selects the bit-rate of the video. The end-to-end latency for fetching the video content from local eNB is randomly assigned between [5,15] ms following a uniform distribution, [20, 60] ms for neighboring MEC servers, and [100, 200] ms for origin content server or CDN on the Internet. The storage capacity of the MEC server for caching the video content is [20-200]GB which is low compared to the video library size. Cache is split at 75% to keep the external traffic load low. Available processing power at MEC server is 50

Mbps which represents the number of encoded bits that can be processed per second. In our evaluation, we considered the following performance metrics:

- Hit ratio - fraction of requests fulfilled from the cache.
- Average access delay - the average time to download initial fragments of the video (sufficient to start the video playback) from cache or CDN/content server to the user device.
- External traffic load - the amount of data fetched from CDN/content server to fulfill the user requests.

In the simulation results, we compare our solution with the following existing approaches:

- *CachePro:* a joint caching and trans-rating scheme with the collaboration among the MEC servers [9].
- *JCCP:* a joint collaborative caching and processing approach with collaboration among the MEC servers [11].

A. Effect of Change in Cache Storage Size

Fig. 3a shows the effect of MEC server cache storage size on access delay. Increasing the cache size reduces the access delay as more videos can be stored in the cache at the MEC servers. Fig. 3b shows that hit-ratio increases with cache size as the number of cached videos at MEC server increases. Fig. 3c reflects that external bandwidth cost also reduces when the storage capacity of the cache is increased. The proposed consolidated caching scheme provides 53% decrease in access delay, 49% increase in hit ratio, and 54% decrease in external traffic load compared to *CachePro*. As compared to *JCCP*, proposed scheme provides 29% decrease in access delay, 11% increase in hit ratio, and 12% decrease in external traffic load.

B. Effect of Change in Available Processing Power

Fig. 4 shows the effect of available processing power for trans-rating at the MEC server. Fig. 4a shows that increase in the processing power decreases the access delay as more the processing power available at the MEC servers, more number of videos can be trans-rated simultaneously. This increases the number of users being served from local MEC server. Fig. 4b shows that with more processing power available, hit-ratio increases. Fig. 4c shows that the external traffic load decreases as processing power increases, as more video requests can be served from one of the MEC servers after trans-rating without any external traffic. For caching capacity of $100GB$, our consolidated caching scheme provides 60% decrease in access delay, 53% increase in hit ratio, and 61% decrease in external traffic load compared to the *CachePro*. As compared to *JCCP*, proposed scheme provides 37% decrease in access delay, 13% increase in hit ratio, and 20% decrease in external traffic load.

VI. Conclusion

In this paper, we propose a two-fold solution for caching at the edge of the mobile network using MEC. By exploiting the collaborative nature of MEC servers, cache consolidation is introduced. With cache consolidation, video content is not replicated on different MEC servers, and thus more videos can be stored collaboratively on the MEC servers at eNBs. More videos cached at MEC servers results in higher hit ratio and reduction in external traffic load. Without replication of videos at the edge of the network, access delay increases. To reduce the average access delay, we introduce splitting of the cache into two logical parts, one part of the cache is used to store full videos, and another part of the cache is used to store initial segments of the video. Replication of initial segments on MEC servers at eNBs reduces the access delay. We combine cache consolidation and cache splitting, to reduce average access delay and external traffic load. Simulation results show that the proposed scheme reduces the access delay and external traffic load compared to *CachePro* and *JCCP*.

VII. Acknowledgment

This work was supported by the project ”Low Latency Network Architecture and Protocols for 5G Systems and IoT”, SERB, Govt. of India.

References

[1] Cisco, “Cisco Visual Networking Index, Global mobile data traffic forecast update, 2016-2021,” *white paper*, February 2017.

[2] ETSI, “Mobile-Edge Computing - Introductory Technical White Paper,” *white paper*, September 2014.

[3] T. Stockhammer, “Dynamic adaptive streaming over HTTP: Standards and design principles,” in *Proceedings of the Second Annual ACM Conference on Multimedia Systems*, ser. MMSys ’11, 2011, pp. 133–144.

[4] E. Bastug, M. Bennis, and M. Debbah, “Living on the edge: The role of proactive caching in 5G wireless networks,” *IEEE Communications Magazine*, vol. 52, no. 8, pp. 82–89, August 2014.

[5] H. Ahlehagh and S. Dey, “Video-aware scheduling and caching in the radio access network,” *IEEE/ACM Transactions on Networking*, vol. 22, no. 5, pp. 1444–1462, October 2014.

[6] A. Gharaibeh, A. Khreishah, B. Ji, and M. Ayyash, “A provably efficient online collaborative caching algorithm for multicell-coordinated systems,” *IEEE Transactions on Mobile Computing*, vol. 15, no. 8, pp. 1863–1876, Aug 2016.

[7] 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*, vol. 52, no. 2, pp. 131–139, February 2014.

[8] B. Shen, S.-J. Lee, and S. Basu, “Caching strategies in transcoding-enabled proxy systems for streaming media distribution networks,” *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 375–386, April 2004.

[9] H. A. Pedersen and S. Dey, “Enhancing mobile video capacity and quality using rate adaptation, RAN caching and processing,” *IEEE/ACM Transactions on Networking*, vol. 24, no. 2, pp. 996–1010, April 2016.

[10] H. Zhao, Q. Zheng, W. Zhang, B. Du, and Y. Chen, “A version-aware computation and storage trade-off strategy for multi-version VoD systems in the cloud,” in *Proceedings of 2015 IEEE Symposium on Computers and Communication (ISCC)*, July 2015, pp. 943–948.

[11] T. X. Tran, P. Pandey, A. Hajisami, and D. Pompili, “Collaborative multi-bitrate video caching and processing in mobile-edge computing networks,” *CoRR*, vol. abs/1612.01436, 2016. [Online]. Available: http://arxiv.org/abs/1612.01436

[12] S. Mosleh, L. Liu, H. Hou, and Y. Yi, “Coordinated data assignment: A novel scheme for big data over cached Cloud-RAN,” in *Proceedings of 2016 IEEE Global Communications Conference (GLOBECOM)*, December 2016, pp. 1–6.

[13] A. Vetro, C. Christopoulos, and H. Sun, “Video transcoding architectures and techniques: an overview,” *IEEE Signal Processing Magazine*, vol. 20, no. 2, pp. 18–29, Mar 2003.

[14] 3GPP, “TR36.814 v9.2.0: Evolved Universal Terrestrial Radio Access (E-UTRA); Further advancements for E-UTRA physical layer aspects,” March 2017. [Online]. Available: www.3gpp.org