# A Socio-Temporal Cache Prefetching Policy for the Multi-access Edge Computing Architecture

Cleomar Márcio Marques de Oliveira,
Igor Monteiro Moraes, Célio Albuquerque
*Laboratório MídiaCom*
*Instituto de Computação*
*Universidade Federal Fluminense*
Niterói - RJ - Brazil
cleomarmarques@id.uff.br, igor,celio@ic.uff.br

José Jerônimo de Menezes Lima
*PPGAdm*
*Programa de Pós-Graduação em Administração*
*Universidade do Vale do Rio dos Sinos*
Porto Alegre - RS - Brazil
jeronimol@unisinos.br

***Abstract*—In this paper, we propose a cache prefetching policy for the Multi-access Edge Computing architecture (MEC) that considers the temporal and social behaviors of mobile users. The key point of our policy is to prefetch new contents based on the popularity of their categories and not on the popularity of each content individually. Our goal is to increase the Quality of Experience (QoE) of mobile users by increasing the cache hit ratio. Simulation results show that HASSAN provides a hit ratio up to 65% for a cache size of 16 MB.**

***Index Terms*—Cache Hierarchy, Caching, Multi-access Edge Computing, Prefetching, Quality of Experience, Wireless Access Networks.**

## I. Introduction

The growing number of mobile devices promotes a dramatic increase in the demand for resources in wireless access networks. We also observed a massive increase of multimedia content traffic on the Internet. In recent years, most of the traffic carried by the Internet core is due to the multimedia content generated and retrieved from these access networks. Mobile devices generate approximately 83% of IP traffic on the Internet through multimedia video content distribution [1].

To deal with the increasing traffic demand, there are several proposals in the literature that employ prefetching and caching techniques to reduce the amount of contents retrieved from servers in the Internet [2]–[5]. The idea is to store content in servers in the wireless access network and thus avoid traffic exchanging with the Internet. With contents stored in servers nearer to users, we also reduce the content retrieval time and increase the Quality of Experience (QoE) of mobile users. Several prefetching and caching techniques have emerged in the literature focusing on the social behavior of mobile users, alone or in association with others. The temporal behavior is an example. The social behavior can be obtained from the analysis of the interaction history of mobile users with the network. Thus, we can establish the individual profiles of mobile users, as explained in the Section IV and also observed in the literature [2]–[6]. In this paper, we propose a prefetching policy, called HASSAN, that selects contents to store in advance based on the social behavior of mobile users. We consider that the Multi-access Edge Computing architecture (MEC) is adopted by the wireless access networks. With MEC, we have edge nodes that employ content caching as services, and are able to perform content prefetching and enforce caching policies according to established criteria. HASSAN considers the popularity of content categories and not the popularity of each content individually as the criterion to prefetch contents. Results show that HASSAN provides a hit ratio up to 65% for a cache size of 16 MB.

This paper is organized as follows. Section II describes some concepts of wireless access networks. We briefly discussed the MEC architecture and cache hierarchy. Section IV introduces our prefetching policy, HASSAN. Section V presents simulation results. Section III discusses related works. Finally, Section VI concludes this paper.

## II. Wireless Access Networks Issues

Figure 1 shows a typical example of a wireless access network with different mobile devices connected to the access points (APs). Those APs can be connected through a core network that has a backhaul access to the Internet.

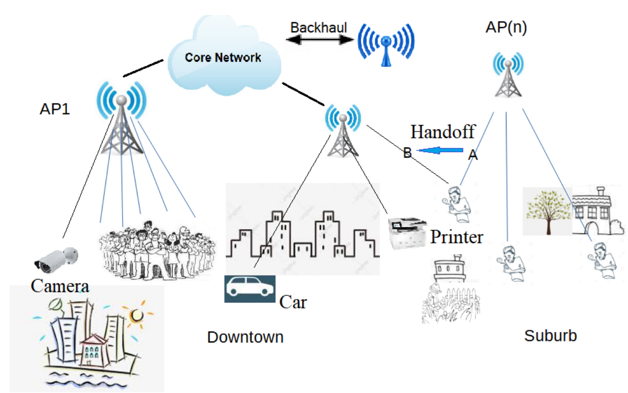


Fig. 1. Wireless Access Network Example.

In recent years, we have observed both the increasing number of mobile devices and the increasing demand of multimedia content retrieved from those devices. The consequence is that the backhaul access of wireless access networks can be overloaded because contents are stored in servers placed on the Internet [1]. In this case, mobile users may have low QoE[1] perception because of network congestion, disconnections, interruptions or temporary resource unavailability, among many other negative factors [14].

In the case of video distribution applications and high network latency, for example, users may suffer with interruptions in video playback, gaps between or inside frames, and freezing frames. With low QoE perception, users often drop their video sessions. Frequent interruptions and reconnections to wireless access networks are also common and may result in non-compliance with mobile application requirements for video playback [14].

To deal with the negative effects caused by the overload of backhaul access, several studies propose to increase the contention window of the media access control protocols, to enhance packet forwarding, to employ more effective routing protocols, in addition to deploying network equipment with more networking and computational power. All these solutions imply financial costs, and greater demand for network infrastructure resources.

The Multi-access Edge Computing (MEC) architecture, as illustrated in Fig. 2, is proposed to reduce mobile data traffic on networks with backhaul [15]. With MEC, we have edge nodes instead of simple APs. These edge nodes, also called MEC servers, employ content caching as services, and are able to perform content prefetching and enforce caching policies according to established criteria. The goal is to store content requested by mobile users in advance, based on recorded context information such as location, destination and content category popularity, for example.

MEC servers can employ cache hierarchy, as illustrated by Fig. 3. In this case, a MEC server has different memory levels to store contents. The higher the memory level in the hierarchy, the more storage capacity this level has to have. The idea is that higher cache levels can serve requests from more users that are from larger geographic regions. In the example of Fig. 3, we have different levels to satisfy national, regional, and metropolitan regions.

In this paper, we consider the use of MEC [15] to evaluate our proposed prefetching policy, HASSAN, and other policies found in the literature. HASSAN also employs a two-level cache hierarchy. Our goal is to increase the cache hit ratio by using the social and temporal behavior of mobile users [10] to prefetch contents. The higher the cache hit ratio, the lower the number of contents retrieved from servers on the Internet. A cache hit means that the content is stored in the local MEC server and, therefore, the user will receive this content from a server that is just one hop away from their mobile device. In this case, possible congestion or other network issues are avoided, with data transmission being only susceptible to the quality conditions of the mobile device's connection to its MEC server. Thus, the QoE perception of users requesting content can increase.

[1]The Quality of Experience (QoE) is a subjective concept adopted by ITU in 2016, [7]. QoE represents the user satisfaction or dissatisfaction degree with a service on the network. It is due to several Quality of Service, QoS [8] factors. QoE perception can be different for each individual. It can be influenced by several factors such as the user's gender, age group, cultural level and economic class. Other social and/or temporal aspects observed in these users in their interactions with the networks, through their mobile devices, also play an important role in QoE perception [2], [9]–[13].

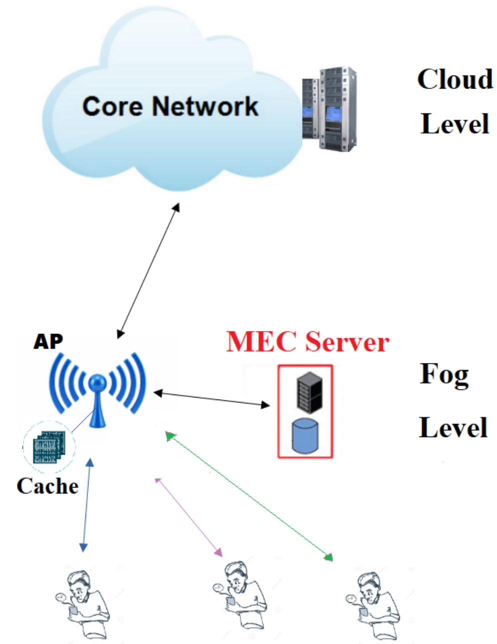


Fig. 2. MEC Architecture.

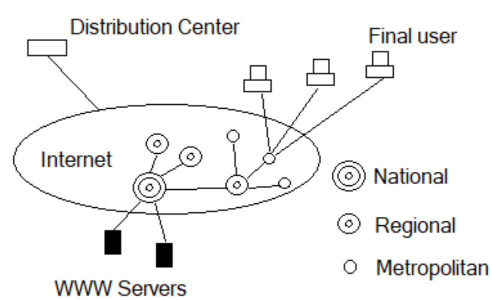


Fig. 3. A general cache hierarchy example.

## III. Related Work

We found in the literature few works related to our HASSAN policy.

Yan *et al.* [4] consider the growing interest in the popularity of online video providers. They found that some users visit different websites in search of the videos they prefer and others visit only a specific website. The work presented in the article was an attempt to model user preference for videos on six websites in China by processing access logs at a large ISP in that country. They proposed a Multi-site Probabilistic Factoring (MPF) generator model, to capture user preferences both on different sites accessed in search of content and on specific and exclusive sites only. From these observations, validations of recommendations were carried out and the performance results were superior to other current factoring models. They concluded from the work that the model presented can allow collaboration between video providers on the Internet.

Baccour *et al.* [3] introduce an approach that deals with storing and prefetching blocks of content on specific MEC servers added to more and working collaboratively in a fog infrastructure, which lightens the load on only one server. The authors considered the growing demand for multimedia content on the network, such as video on demand (VoD). Then, they proposed using multiple servers on the edge networks in a MEC Architecture, for mutual collaboration in the pre-selection and storage of content. The objective is to alleviate demand on the network's core and improve performance on edge access networks. Content chunks with different types of encoding are stored in a given server only. The idea to store only chunks optimizes the fog server limited resources about power processing requirements. This idea contributes to a higher overall performance of the edge network.

They designed a probabilistic model through the users' visualization pattern of the videos study. Them the probabilistic model selects the chunks most likely to be requested. Based on this model, they have developed two policies related to the popularity of content—one for prefetching and the other for replacing content in the cache. The performance results obtained compared to traditional cache management policies were superior. HASSAN policy may work associated with this approach due to the high content volume available at a provider. Previously they stored the multimedia content considered the most critical and impactful in an edge network on different servers in the Fog Computing infrastructure. Whether on servers with greater computing power or storage capacity.

Li *et al.* [5] consider the benefits that can be obtained with access networks to meet the demands of new terminal applications and timely content delivery services. These have increasing network performance requirements in addition to greater processing power. To meet these demands, some challenges were identified to avoid unsatisfactory responses from the network. The idea was to use the caching strategy, associated with the management of contents in cache memories through the LRU policy, associated with a mechanism based on priorities for the replacement of contents. Priority queues were implemented, with weights being assigned to their contents. Those without new requests were given lower weights and considered candidates for replacement in the cache. These weights were recalculated with each new request. Content with lower final weight was replaced. To improve the performance of the network with the strategy, prefetching was used, based on the theory of Bayesian networks. Network nodes with less data load were selected to store the prefetching files. Evaluations of the strategy were carried out in an operational network at a university and the results were the best in benckmarketing tests in the metrics of hit rates and memory consumption.

Rim and Kang [16] introduces an approach that also uses the prefetching strategy with D2D communications to alleviate the demand on APs close to mobile devices. The content selection criterion is based on popularity and demand, based on your preference predicted by users in the near future. The probability of a content being rated as preferred depends on each content. It is considered that in a large cache, which stores the most popular content, the hit rate is likely to be high. To implement this idea, a mobile device with large memory capacity was established to provide the contents requested by other users in its vicinity. The simulations showed that prefetching considering the prediction for a near future, obtained better results than for a more distant future.

## IV. The Proposed Prefetching Policy: HASSAN

The proposed HASSAN policy is considered of low computational complexity. However, as shown in Section V, it presents good performance results, compared to those with similar models that do not use social factors and do not consider the popularity of the content categories, but the popularity of the content.

The HASSAN prefetching policy selects contents to store in advance based on the social and temporal behaviors of mobile users. Our basic assumption is that mobile users have the habit of requesting their content in the same places and at the same period of the day, or close to them [10]. Trestian *et al.* [10] show that mobile users tend to request content from the same AP, at or near the same time and in the same geographic region. This behavior is observed for mobile users uploading files. In our work, we assume that this behavior is also observed for mobile users downloading their requested files. HASSAN also considers that mobile users have preferred video categories and request videos from these categories of their own preference, i.e, the prefetching criterion of a content is based on the popularity of the category that this content belongs to and not on the popularity of this content.

The more accurate the prediction of content request is, the higher the cache hit ratio is. A prediction model is necessary because human behavior is not mathematically equitable. In literature, heuristics are developed to establish behavior patterns close to the real behaviors of these users. In this way, user profiles can be established based on the historical record of their requests in the MEC server, as in this proposed policy, or by some other strategy or approach in literature. Depending on the degree of technological development of the computational tools used to establish such profiles, in addition to other scientific resources, the prediction of requests can be more or less accurate. The exploratory model used by HASSAN considers principles of Zipf's Law and other Power Laws. Due to the amount of data and number of variables considered, descriptive statistics showed good results to assist in the prefetching content that is more likely to be requested by users connected to a MEC server.

HASSAN defines an observation window $w_i$, where $i = 1 \ldots \infty$ and a time slot $t_j$, where $j = 1 \ldots W/t_p$ and $W$ is the size of the observation window and $t_p$ is the duration of the time slot, both in units of time. The observation window $w_i$ is composed of $W/t_p$ time slots. Let us assume the current observation window is $w_i$ and the current time slot is $t_j$. HASSAN records the history of content requests in the time slot $t_j$ in one MEC server. This history of request is used to calculate the popularity of content categories in this time slot $t_j$, which is given by $p_{c,j} = r_{c,j}/R_j$, where $p_{c,j}$ is the popularity of category $c$ in the time slot $j$, $r_{c,j}$ is the number of requests of contents that belongs to category $c$ in the time slot $j$, and $R_j$ is the total number of content requests in the time slot $j$. At the end of the time slot $t_j$ in the observation window $w_i$, all the contents in cache are removed and new contents are prefetched based on the popularity of each category calculated in the time slot $t_j$ in the observation window $w_(i-1)$. For example, HASSAN defines the time slot duration as one hour and the observation window as one day. Thus, every day at 3 PM, for example, HASSAN will remove all the content currently in the cache of a MEC server and new contents will be stored based on the history of requests made on the previous day between 3 PM and 4 PM. The duration of both the time slot and the observation window may vary depending on the application the prefetching policy will apply.

HASSAN employs a two-level cache hierarchy. With more data storage capacity than the lower level, the higher level cache complements the content stored in the former. Content chunks are stored in cache memories and referenced by network users' requests. This operation occurs primarily in the low-level cache memory on the MEC servers. Another practical aspect of HASSAN is to optimize caching capacity. Low-level caching only stores a single chunk of each different prefetched content. The other chunks that make up a piece of content are stored in the top-level memory of the cache hierarchy. The results obtained in the evaluations carried out allow a performance analysis with more cache levels through extrapolation of the results obtained and presented in this study.

## V. Performance Evaluation

We developed a simulator to evaluate our HASSAN prefetch policy. This simulator consists of algorithms to assemble the data sets that allow the simulations to be carried out. It also has the algorithms related to the compared policies, in addition to the one referring to the HASSAN policy guidelines. It also has algorithms for counting hits and for generating statistics in performance evaluations. All these algorithms are separated and logically grouped in three modules. This facilitates the reading and the sequence of execution of the evaluations.

We compare HASSAN with classical caching policies: FIFO (First In, First Out), LRU (Least Recently Used), LFU (Least Frequently Used), and random. We consider two performance metrics: the cache hit ratio (HR) and the efficiency of prefetching (EP). The EP metric is defined as the result of dividing the total number of prefetching content by the number of cache hits. The amount of content loaded by prefetching is greater than the storage capacity of the cache memory. The content exceeding this cache capacity are stored on the MEC server associated with the AP. The evaluation scenario considers one MEC server. All the experiments were performed with real data from the data set of the Internet video provider MovieLens [17]. This data set contains ratings of movies. We consider one video rating as one content request. In fact, we consider five slices of the Movielens data set in our evaluation. Each slice contains the video ratings of one month. In previous work [2], we only have considered synthetic data and one metric to evaluate the policies.

### A. The Cache Hit Ratio Analysis

The goal of the first analysis is to show the increase in the number of cache hits with HASSAN and, consequently, a better perception of QoE by mobile users in their requests.

QoE is an abstract concept and its perception may vary depending on several factors, such as age, sex, intellectual level and socioeconomic factors. In this sense, increasing the hit rate can provide better satisfaction to the mobile user because it can avoid problems arising from the level of congestion in the network core, for example, if there is no hit in your content request.

In Fig. 4, colored bars show the average hit ratio (HR) obtained for each compared policy for each size of cache capacity. The cache capacity ranges from 512 kB to 16 MB. The standard deviation can be seen in the graph as well.

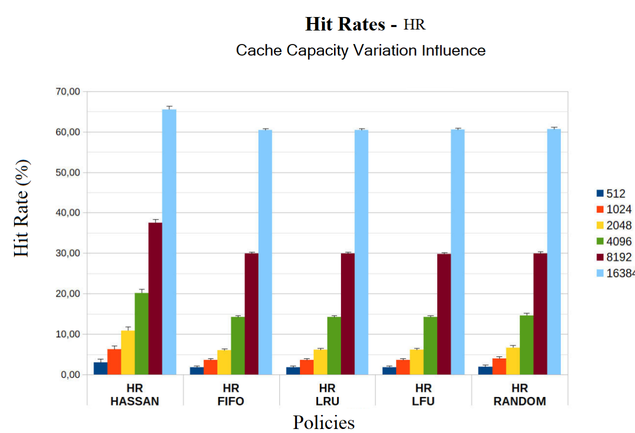


Fig. 4. The cache capacity impact on the policies' cache hit ratio.

Results confirm our expectation of the good performance of HASSAN in relation to the number of cache hits. HASSAN outperforms the four policies in terms of the cache hit ratio regardless the cache capacity. With 16 MB of cache capacity, HASSAN provides a cache hit ratio of 65%. This result corroborates that is a good choice to consider the social and temporal behaviors of mobile users as HASSAN does. The prefetching criterion of HASSAN, , considering the content categories popularity, is the main difference for the other policies. FIFO, LFU, LRU, and random consider the same file of prefetched content in the cache, obtained with the criterion of individual popularity of each video.

In the following analysis, we introduce a weight to the contents in the historical record of requests. The aim was to improve fairness in computing the popularity of content categories. Content categories with more recent requests are more likely to have a new request for content from these categories.

The distribution used for the records of requests was the Distribution in Quartiles. This distribution is considered suitable for the type and volume of data, being widely used in data science applications for data analysis. The weights are assigned to the contents in each Quartile, according to its timestamp, with values of 1.0 for the first, oldest Quartile, being incremented by 0.25 as one advances to the next Quartile, up to the Quartile most recent, with 1.75 weights assigned to its content. When calculating the popularity of categories, the most recently requested content counts more points for its respective category, as these categories is more likely to have new requested content, as this gives more fairness to the process.

In Fig. 5, the results obtained in the third policy performance experiments are graphically shown. You can observe different cache memory capacity values, 5 different policies and two performance metrics. The data presented are the average obtained from the processing of 5 slices of data from the data set used. The standard deviation can be observed.

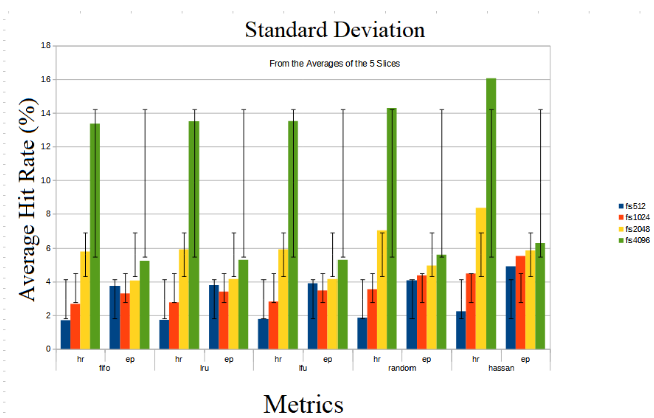


Fig. 5. The standard deviation of the cache hit ratio.

## B. Analysis of Variance of Evaluation Output Data set

We analyzed the variance to show that the set of videos that were cached with HASSAN and counted hits is different from the set of cached videos that count hits from the other cache policies. Descriptive statistics is a term that refers to data analysis that portrays data meaningfully. From them, patterns can be obtained. It contributes to their analysis, but does not allow to affirm the validity or not of a given hypothesis exclusively from them [18].

This test basically compares the means between the groups and determines if any of these means are significantly different from each other. ANOVA is a test that can tell which group is significantly different from each other.

The output of the analysis of variance of Tab. I shows in the second column the values of F. It is a value produced by ANOVA, F-statistics or F-ratio. It is the ratio between the model and its error, resulting from the ANOVA F tests. F statistics are based on the mean squares ratio. It is an estimate of variance. The last column shows the pvalue output number. These values were obtained from the analysis of variance between the respective input sets indicated in the first column with HASSAN policy data set.

They indicate the probability that the input sets are the same or different from each other. When the pvalue is less than 0.005 (0.5%) we reject the null hypothesis. When pvalue is greater than 0.005 (0.5%) we do not reject the null hypothesis. Null hypothesis means that the analyzed sets may not be different from each other. The data obtained are compared using the pvalues, which allow an evaluation of the null hypothesis. If the values of P are low enough, the possibility of the data constituting a true null is unlikely [18], [19].

By associating the results of the analysis of variance in Tab. I to the results shown in Fig. 6, it can be seen that the proposed policy performed better. HASSAN policy HR and EP rate values are higher.

**Slice1**

| | | fifo | lru | lfu | random | hassan |
|---|---|---|---|---|---|---|
| fs512 | hr | 1,55 | 1,55 | 1,6 | 1,65 | 2,1 |
| | ep | 3,4 | 3,4 | 3,51 | 3,62 | 4,61 |
| fs1024 | hr | 2,25 | 2,2 | 2,25 | 3,55 | 4,55 |
| | ep | 2,77 | 2,71 | 2,77 | 4,37 | 5,6 |
| fs2048 | hr | 4,9 | 5 | 5 | 6,2 | 8 |
| | ep | 3,44 | 3,51 | 3,51 | 4,35 | 5,62 |
| fs4096 | hr | 13 | 13,1 | 13,15 | 14,4 | 15,85 |
| | ep | 5,1 | 5,14 | 5,16 | 5,65 | 6,22 |

Fig. 6. The variance table.

As final consideration of the evaluations carried out, we can corroborate that the proposed policy can contribute to improve the perception of QoE of mobile users in the presented scenario. The improvement in the perception of QoE will occur through this higher rate of HR hits in your requests.

TABLE I
THE VARIANCE OUTPUT.

| Variance - Frame Size 512 - Slice 1 | | |
|---|---|---|
| Policy | F_onewayResult: statistic=array | pvalue=array |
| FIFO | 11.2447439 | 0.00128354 |
| LRU | 11.2447439 | 0.00128354 |
| LFU | 11.95203766 | 0.00091992 |
| RANDOM | 11.2447439 | 0.00128354 |

## C. HASSAN Policy Data Savings Simulation

In the illustration of Fig. 7, NetFlix CDN architecture is shown. In order to simulate the data savings provided by the Hassan policy, this simulation considered an AP from that CDN, with only 1 hop.

The results obtained from the hit rates of slice1 data from the third performance evaluation performed with the data set for the LFU and for the HASSAN policy were considered in Fig. 8.

Only the results of the cache hit ratio of LFU and HASSAN policies of Slice 1 were considered, for simplicity.

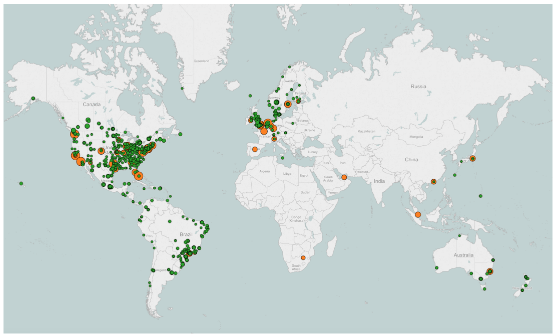


Fig. 7. The NetFlix worldwide CDN.

Fig. 7 shows the NetFlix CDN on all continents. This CDN was used as an example for the simulation of savings generated with the use of the Hassan policy in a node of this network.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | Simmulation Parameters | | | | | |
| | | Requested Content – RC | | 2000 | | | |
| | | Content Size - CS | | 300 MB | | | |
| | SLICE 1 | | | E | F | G | H |
| A | B | C ( % ) | D ( % ) | D * RC / 100 | E * 300 ( MB ) | D – C | G * 300 ( MB) |
| Cache Size | Metric | LFU | HASSAN | | | | |
| fs512 | hr | 1,6 | 2,1 | 42 | 12600 | 0.5 | 150 |
| | ep | 3,51 | 4,61 | | | | |
| fs1024 | hr | 2,25 | 4,55 | 91 | 27300 | 2.3 | 690 |
| | ep | 2,77 | 5,6 | | | | |
| fs2048 | hr | 5 | 8 | 160 | 48000 | 3 | 900 |
| | ep | 3,51 | 5,62 | | | | |
| fs4096 | hr | 13,15 | 15,85 | 317 | 95100 | 2.7 | 810 |
| | ep | 5,16 | 6,22 | | | | |
| | | | | Sum ( MB ) : | 183.000 | | 2.550 |

Fig. 8. The accumulated data savings.

Approximately 93 GB did not travel through the network in the simulation, for cache size 4 MB for example, as shown in Fig. 8, since they were on the same AP where the mobile users requested the content, just a hop away from the users' mobile devices.

## VI. Conclusions

In this work, we have proposed a prefetching policy, called HASSAN, that selects contents to store in advance based on the social behavior of mobile users. We assume that mobile users have the habit of requesting their content in the same places and at the same period of the day, or close to them [10]. We also considers that mobile users have preferred video categories and request videos from these categories of their own preference. As a consequence, the prefetching criterion of HASSAN is based on the popularity of the category that this content belongs to and not on the popularity of this content.

We have compared HASSAN with classical caching policies: FIFO, LRU, LFU, and random. We have considered two performance metrics: the cache hit ratio (HR) and the efficiency of prefetching (EP). All the experiments were performed with real data from the data set of the Internet video provider MovieLens [17]. Simulation results show that HASSAN provides a hit ratio up to 65% for a cache size of 16 MB.

As future works, we intend to use artificial intelligence associated with machine learning techniques to determine the behavior of mobile users and, thus, enhance the performance of our prefetching policy.

## References


[1] Cisco, "Cisco annual internet report (2018 - 2023) white paper," Internet, https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html, Mar. 2020, accessed in 2022/03/01.

[2] C. M. M. de Oliveira, I. M. Moraes, and C. V. N. de Albuquerque, "A cache prefetch policy based on users' temporal-and-social behavior for content management in wireless access networks." *9th Latin American Network Operations and Management Symposium (LANOMS)*, Sep. 2019.

[3] E. Baccour, A. Erbad, A. Mohamed, K. Bilaly, and M. Guizani, "Proactive video chunks caching and processing for latency and cost minimization in edge networks," *IEEE Conference on Wireless Communications and Networking*, 2019.

[4] H. Yan, C. Yang, D. Yu, Y. Li, D. Jin, and D. M. Chiu, "Multi-site user behavior modeling and its application in video recommendation," *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[5] C. Li, M. Song, S. Du, X. Wang, M. Zhang, and Y. Luo, "Adaptive priority-based cache replacement and prediction-based cache prefetching in edge computing environment," *Journal of Network and Computer Applications*, vol. 165, Sep. 2020.

[6] N. Garg, M. Sellathurai, , V. Bhatia, B. N. Bharath, and T. Ratnarajah, "Online content popularity prediction and learning in wireless edge caching," *IEEE Transactions on Communications*, vol. 68, no. 2, Feb. 2020.

[7] I.-T. P. S. G. . Recommendations., "Executive summary of study group 12 meeting." Internet, https://www.itu.int/en/ITU-T/studygroups/2013-2016/12/Pages/exec-sum.aspx, Jun. 2016, accessed in 2022/03/22.

[8] A. Salama, R. Saatchi, and D. Burke, "Quality of service evaluation and assessment methods in wireless networks," *4th International Conference on Information and Communication Technologies for Disaster Management*, Dec. 2017.

[9] C. Li, L. Toni, J. Zou, H. Xiong, and P. Frossard, "Qoe-driven mobile edge caching placement for adaptive video streaming," *IEEE Transactions on Multimedia*, vol. 20, Apr. 2018.

[10] I. Trestian, S. Ranjan, A. Kuzmanovic, and A. Nucci, "Taming the mobile data deluge with drop zones," *IEEE/ACM Transactions on Networking (TON)*, vol. 20, pp. 1010–1023, Aug. 2012.

[11] K. ur Rehman Laghari, N. Crespi, B. Molina, and C. Palau, "Qoe aware service delivery in distributed environment," *Waina*, 2011.

[12] A. Raake, M.-N. Garcia, W. Robitza, P. List, S. Goring, and B. Feiten, "A bitstream-based, scalable video-quality model for http adaptive streaming: Itu-t p.1203.1," *ITU-T Study Group 12 - IEEE*, 2017.

[13] K. Fizza, A. Banerjee, K. Mitra, P. P. Jayaraman, R. Ranjan, P. Patel, and D. Georgakopoulos, "Qoe in iot: a vision, survey and future directions," *Discover Internet of Things*, Jan. 2021.

[14] D. V. C. Silva, G. M. B. Domingues, P. B. Velloso, and A. A. A. Rocha, "Analysis of mobile-live-users of a large cdn," in *IEEE Symposium on Computers and Communications (ISCC)*, 2018, pp. 946–951.

[15] ETSI, "ETSI - Multi-access Edge Computing - MEC," *Internet*, 2021, Accessed on 2022/03/01.

[16] M. Rim and C. G. Kang, "Content prefetching of mobile caching devices in cooperative d2d communication systems," *IEEE Access*, 2020.

[17] GroupLens, "Movielens 20m data set," Internet, https://grouplens.org/datasets/movielens/20m/, 2021, accessed on 2022/04/26.

[18] S. Madhavan, *Mastering Python for Data Science*, P. Publishing, Ed. www.packtpub.com: Packt Publishing Ltd, Aug. 2015.

[19] M. B. Editor, "How to correctly interpret p values," Internet, https://blog.minitab.com/en/adventures-in-statistics-2/how-to-correctly-interpret-p-values, 2014, accessed on 2022/may/1. [Online]. Available: https://blog.minitab.com/en/adventures-in-statistics-2/how-to-correctly-interpret-p-values