New caching system under uncertainty for Mobile Edge Computing

Sarra Mehamel, Samia Bouzefrane, Khaled Slimani, Mehammed Daoui

New caching system under uncertainty for Mobile Edge Computing

1st Mehamel Sarra
LARI lab, University Mouloud Mammeri, Algeria
CEDRIC lab, Conservatoire National des Arts et Métiers,France
sarra.mehamel@yahoo.com

2nd Bouzefrane Samia
Conservatoire National des Arts et Métiers
CEDRIC lab,Paris,France.
samia.bouzefrane@lecnam.net

3rd Slimani Khaled
LARI lab,University Mouloud Mammeri, Tizi Ouzou
Algeria
khaledrmse@gmail.com

4th Daoui Mehammed
LARI lab, University Mouloud Mammeri, Tizi Ouzou.
Algeria
mdaouidz@yahoo.fr

***Abstract*—Edge caching is one of the most emerging technologies recognized as a content retrieval solution in the edge of the network. It has been also considered as enabling technology of mobile edge computing (MEC) that presents an interesting opportunity to perform caching services. Particularly, the MEC servers are implemented directly at the base stations (BSs) which enable edge caching and ensure deployment in close-proximity to the mobile users. However, the integration of servers in mobile edge computing environment (base stations) complicates the energy saving issue because the power consumed by mobile edge computing servers (MECS) is costly especially when the load changes dynamically over time. Furthermore, users with mobile devices arise their demands, introducing the challenge of handling such mobile content requests beside the limited caching size. Thus, it is necessary and crucial for caching mechanisms to consider the mentioned factors, meanwhile most existing studies focus on cache allocation, content popularity and cache design. In this paper, we present an energy-efficient fuzzy caching strategy for edge devices that takes into consideration four influencing features of mobile environment, while introducing a hardware implementation using Field-Programmable Gate Array (FPGA) to cut the overall energy requirements.**

***Index Terms*—Edge caching, Mobile edge computing, Energy efficiency, Fuzzy logic, FPGA.**

I. Introduction

Edge computing refers to the technologies that enable computation offloading and data caching to be performed at the edge of the network [1]. Caching popular content at the edge of the network has become a committed technology to facilitate storage and delivering in order to improve the quality of services for mobile users. Therefore, edge caching has initially been applied in mobile edge computing to enhance efficiency ranges and to reduce energy consumption of the systems. These enhancements are established so that the popular contents are likely to be requested several times from many users. The most conventional web caching methods are not efficient enough and may suffer from a cache pollution problem, since they consider just one factor and ignore other factors that may have an impact on the efficiency of web caching [2]. Many web caching policies have attempted to combine different factors to make a caching decision. These combinations can influence the performance of web caching. The challenge lies in making a caching decision and in choosing which content should be cached and which one should be evicted to make the best use of available cache space. Hence, reducing network traffic and server will be less overhead too as well as satisfying the mobile user. The studies proposed the idea of using machine learning techniques to cope with the problem of combining significant factors [3]. The model applied in [4] deploys supervised learning algorithm decision tree used to combine significant factors (frequency and size) and to predict contents that can be re-visited later. The results have revealed that the proposed algorithm significantly increased hit-ratio, byte hit-ratio and reduced the latency. Similar benefits can be achieved by using the remaining combination at the edge however it highlights the increased computational complexity regarding to the critical role of edge servers. The main motivation of using Fuzzy logic and FPGA in adopting edge caching approaches is the need to base the caching decision on both qualitative and quantitative information, as well as to move the workload from software to hardware in order to reduce the energy consumption and the server load of the edge devices. Fuzzy logic has been widely used in many applications because of its rapid preliminary study on different inputs, the easy adaptation and interpretation of the rules and the ability of using heterogeneous inputs which facilitate the combination between several factors in the most desirable way without using mathematical relations. The remaining parts of this paper are organized as follows. Section II introduces the background as well as the related works. In Section III, we illustrate the framework for the caching decision using fuzzy logic. Section IV presents the implementation of the proposed caching strategy and the evaluation of results. Finally, the paper concludes in Section V.

II. Background

Many studies [4]–[7] evaluated the performance of caching algorithms. A detailed comparative study of Leave Copy

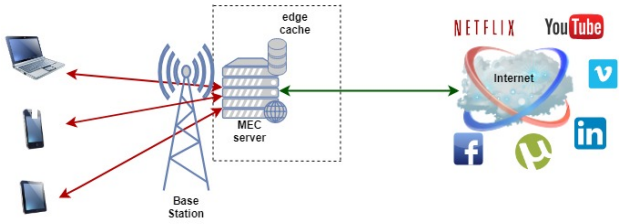

Fig. 1. Edge caching Architecture

Everywhere (LCE), Leave Copy Down (LCD), ProbCache, MAGIC was reported in [7], highlighting that each caching strategy is appropriate for one specific scenario with specific objectives and clarify that selecting the best strategy depends on these objectives. Least Recently Used (LRU) presented in [4] is a recency-based strategy since it depends on the recency of requesting a content as a main factor when removing the data contents from a full cache, and states that recently requested objects are more likely to be requested next. Another replacement strategy is Least Frequently Used (LFU) which is a frequency-based strategy. LFU evicts from the cache the content that was requested the fewest number of times and considers that the contents more likely to be requested again are the ones that have been requested more times. In [8], K. Cho et al. used the popularity of the content to determine the number of the content that should be cached. Ioannis Psaras et al. [9] calculate the caching probability of each content. In order to increase the cache hit ratio, the above studies are directed to be appropriate for different network topologies, but they consider neither the properties of the content itself like size nor the influencing factors and the end user characteristics such as cost and mobility. They only accumulate contents with past high popularity or high frequency which may no longer be useful over time. To adjust adaptively the varied content proprieties and its influencing factors, we present a fuzzy control caching system for edge servers which can select the more priority content to be cached based on well chosen factors. To decide whether to cache or evict contents, the decision making process is based on the: mobility, frequency, cache occupancy and cost of retrieval.

Mobility: since mobility is a fundamental feature of wireless systems, several analytical models are available [10], mobility-based caching policies have been investigated, such as in [11] and [12] where they used the mobility of users to predict the next base station in addition to the next requested content to be cached on it. The mobility can be modeled by several ways and the most used in literature is Markov chain. When user mobility is modelled by a Markov chain random walk, the optimal storage space is approximately solved by Poularakis and Tassiulas in [13] and the same model is used in [14] and [15]. The use of mobility in caching leads to insure its influencing role. Authors in [10] and [16] developed a mathematical model to explore the impact of user mobility on the performance of cache and it has been proved that user mobility can be turned into an advantage to enhance caching decision and minimize the incurred service cost.

Frequency: in the present context, the frequency defined as how often the contents are requested or intended to be requested, it is extracted from a function that establishes the popularity of every content. Content popularity is commonly modeled with a probability distribution function such as a Zipf or MZipf [17]. Authors in [18]–[20] used a Zipf popularity model with α parameter with $\alpha \in$ [0.6-2.5]. Other authors have extracted real traces from Content delivery network (CDN) [17]. Frequency based caching schemes store only popular content and keep count of the frequency of contents to determine the ones that are expected to become popular. The above studies have demonstrated how the frequency parameter can be fine tuned to obtain the best caching performance.

Cache Occupancy: The cache occupancy determines the size available in the cache. This size is usually expressed with an absolute value or a ratio. We express it by the following: CO = 1 means that the content occupies 100% of cache size. There are many related studies in the literature that propose and make use of the distribution of the content sizes [16]. In the literature, mostly, the size of cache is fixed and contents are used with homogeneous or heterogeneous sizes and gathered in a catalogue to represent the entire collection of contents in the network. Hence clarifying the strong correlation of size on cache performance can be determined.

Cost of Retrieval: In this paper, we take into account the cost of content retrieval, i.e., the cost associated to the external bandwidth needed by a MEC to retrieve the requested content to the end users. Classical caching strategies aim to maximize cache efficiency and propose solutions with high overall cost. To solve this mismatch, authors in [18] formulate a polynomial-time greedy algorithm to solve the optimization models that either minimize the cost or maximize the hit-ratio. Results show that significant cost savings are attainable. It is confirmed that the consideration of cost can impact the performance of caching systems.

III. Fuzzy inference system

Fuzzy logic is an extension of boolean logic dealing with the concept of partial truth which denotes the extent to which a proposition is true. Whereas the classical logic holds that everything can be expressed in binary terms (0 or 1, black or white, yes or no), fuzzy logic replaces Boolean truth values with a degree of truth. Degree of truth is often employed to capture the imprecise modes of reasoning that play an essential role in the human ability to make decisions in an environment of uncertainty and imprecision [19]. Fuzzy Inference Systems (FIS) consist of an input, a processing and an output phases. In the input phase, the inputs are mapped to an appropriate membership function with specific values. The processing stage consist of preforming each appropriate rule and generating a corresponding result. It then combines the results. Finally, the output phase converts the combined result back into a specific output value. The membership function of a fuzzy set represented by divided ranges defines how each

value in the input space is mapped to a membership degree. Our proposed model, as illustrated in the Figure 2, is made of the main FIS components.

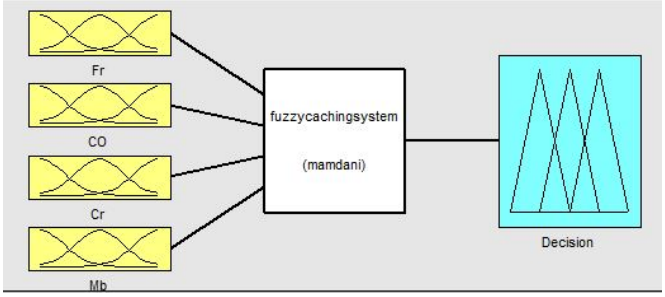

Fig. 2. Fuzzy inference system of caching decision

The preliminary step toward the construction of the fuzzy algorithm, is to analyze previous studies to understand the impact of each input and to control the output as a result of this analysis. We have chosen four input variables that describe each web content in term of: frequency, cache occupancy, mobility and cost of retrieval, as shown in table I. The output variable is the decision of caching, as depicted in table II.

TABLE I
INPUT VARIABLES DESCRIPTION

input	Description
Fr	Frequency of demand of each content in time period
Cr	The time incurred to retrieve the content
CO	Occupancy of content in the cache
Mb	The users proximity from the base station

TABLE II
OUTPUT VARIABLE DESCRIPTION

OUTPUT	Description
DECISION	The decision taken to cache contents

For each of these variables, we have defined:

- The fuzzy sets with membership functions describing the degree of membership of each variable to the corresponding fuzzy set as shown in Tables III and IV.
- The fuzzification of the four input variables and the output variables means establishing the linguistic variables.

Classification priority can be defined in the basis of recent part observations across the network. However from the perspective of approximate logic, the intention is defined on the observation and experience of end users. However, more precise linguistics can yield finer determination of membership grade. One of the popular methods to find such membership grades for the end user is the degree of similarity evaluation. It describes that if two or more observations are similar, then, grade of similarity is calculated [28]. Based on the degree of similarity, the partition of linguistics may be provided.

There are three membership functions associated with the defined linguistic variables: frequency (FR) : {Low, Medium, High}, referring to popularity of the content. Whenever the frequency of demanding a content is high, the popularity increases and vice-versa. Cache occupancy (CO) : {Small, Medium, Large} which represents the occupancy of a content in the cache expressed by dividing the content size on the remaining cache size. The closer CO value from 1 means that it occupies more size from the cache. Mobility (Mb) : {So-close, Close, Far} in reference to the distance between the base station where the cache is located and the end user. This proximity is updated each time the user moved from one base station to an other. Cost of retrieval (Cr) : {Short, Medium, Long} is the cost to retrieve the requested content to the end user in reference to the time incurred.

TABLE III
FUZZIFICATION OF INPUT VARIABLES

	Description	Ranges
Fr	Low, Medium, High	[0-0.2], [0.3-0.5], [0.6-1.0]
Cr	Low, Medium, Long	[0-0.3], [0.4-0.6], [0.7-1.0]
CO	Small, Medium, Large	[0-0.3], [0.4-0.6], [0.7-1.0]
Mb	So-close, Close, Far	[0-0.3], [0.4-0.7], [0.8-1.0]

To describe the output variable, we have chosen the following membership functions: {Low, Medium, High} which represent the priority of caching decision. After defining the membership function, we build the rule base. These rules are represented by IF-Then clauses in which the antecedents are the conditions of the mobile user and the properties of the content while the consequence is the caching decision. There is no general procedure to decide the number of fuzzy rules and the role of each involved factor in the decision making. In our case, the set of rules are based on the understanding of cache behaviour under different scenarios. Examples of some fuzzy logic rules are the following:

IF Fr is LOW and CO is SMALL and Cr is SHORT and Mb is SO_ CLOSE THEN DECISION is MEDIUM

IF Fr is LOW and CO is SMALL and Cr is SHORT and Mb is CLOSE THEN DECISION is MEDIUM

IF Fr is LOW and CO is SMALL and Cr is SHORT and Mb FAR THEN DECISION is LOW

IF Fr is LOW and CO is SMALL and Cr is MEDIUM and Mb is SO_CLOSE THEN DECISION is MEDIUM

TABLE IV
FUZZIFICATION OF OUTPUT

	Description	Ranges
Decision	Low, Medium, High	[0-0.2], [0.3-0.5], [0.6-1]

IV. Implementation and evaluation

This section presents the implementation and the evaluation of Fuzzy caching system for edge computing (FCEC) We divide the evaluation into two parts: first, we compare the software implementation of FCEC with the Least Recently Used (LRU) and First In First Out (FIFO) strategies. We used typescript with RxJS which is a reactive programming library for the software implementation, then we moved the algorithm into hardware using FPGA Xilinx virtex-5 LX50T-1156 board from DIGILENT coded with VHDL(VHSIC Hardware Description Language).

The basic performance characteristic of a cache is a hit ratio. The hit ratio is computed according to the following equation:

$$cacheHit = \sum_{N}^{i=1} hit_i / \sum_{N}^{i=1} hitsi + \sum_{N}^{i=1} miss_i \quad (1)$$

A cache hit occurs when the request is served from cache, otherwise it is a miss. Thus, the cache hit ratio is the number of cache hits divided by the sum of cache misses and hits (total number of requests) in a given time. A high hit ratio means that a cache performs well.

A low hit ratio means that the data in cache should not be cached or that the cache size is too small to stand temporal locality of all contents. In our software solution, we designed a client-server model with a cache as a proxy between the client and the server. We generated a set of resources (contents with heterogeneous sizes) and a set of requests, then we investigated how the cache hit ratio changes when the average cache size varies. In the admission phase (placement), we note that the fuzzy caching system gives each content in web a priority and places it in the cache according to this priority, unlike LRU and FIFO.

TABLE V
Notation

Notation	Description
S	cache size
Id_i	Id or Index of content
N	Number of contents
R	Number of Requests
P_i	Priority of caching decision for each Id_i

Algorithm 1 Fuzzy Caching strategy

Initialization: Initialize N, R with Zipf distribution function
Client sends Request R
Base station receives R
Apply fuzzy rules to set P_i
while $S > 0$ **do**
  if *content in cache* **then**
    $hit \leftarrow hit + 1$
    return Id_i
  else
    $miss \leftarrow miss + 1$
    get $Id_i fromserver$
    while $S = 0$ **do**
      check $P(Id_i)$
      Evict :Id with min{P(Id)}
    end
    Push in cache : Id_i
    return Id_i
  end
end

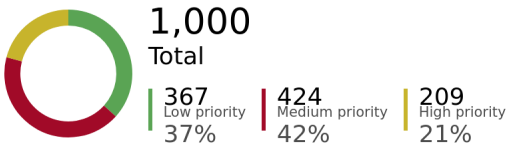

Fig. 3. Distinction between web object by giving each a priority

In Figure 3, we have a catalogue of 100 different contents with heterogeneous sizes. The requested contents were classified into Low, Medium and High priority in order to take the caching decision.

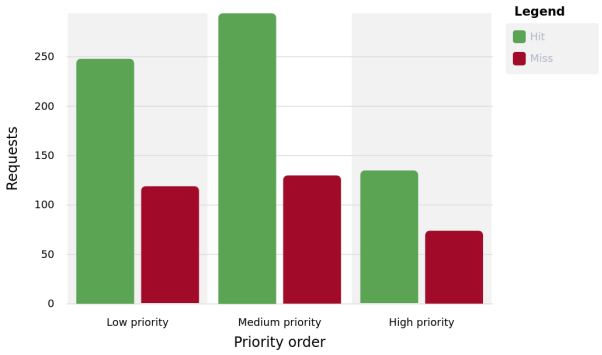

Fig. 4. Cache Hit ratio in High priority requests

In the eviction phase (replacement), we used the content priority and the content size to choose the object that should be replaced. The efficiency of our proposed technique appears when we launch random requests over time. We should

mention that it became significant that high-priority contents in the web are most likely to be requested followed by Medium and Low ones. In Figure 4, a hit rate of 65% appears in high priority content.

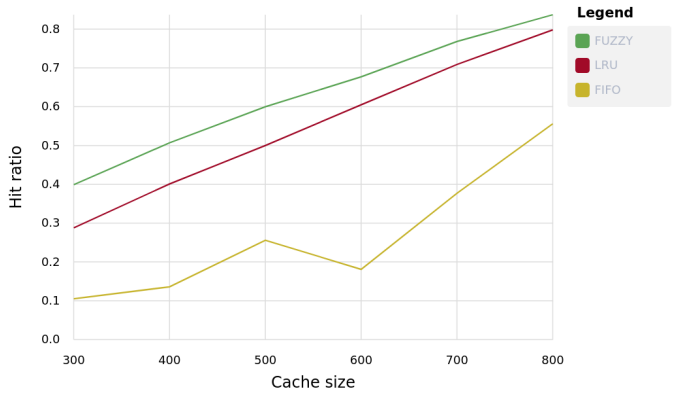

Fig. 5. Cache hit ratio of Fuzzy caching system, LRU and FIFO

To evaluate our proposal, we compared hits ratios with two caching strategies:

- **Least Recently Used (LRU) [21]:**
  In this strategy, the system caches the most popular requests and when the cache storage is full, the system evicts the least requested recently content, and replaces it by the new most requested one.
- **First In First Out (FIFO) [22]:**
  In this strategy, the system records the time of caching contents. In case of a full cache, the system evicts the content which was earliest stored and replaces it by the new content.

For the distribution of requests and contents, studies have confirmed Zipfs law as an appropriate distribution model for access pattern to contents on the Internet such as videos hosted on YouTube [23] or peer-to-peer file sharing systems (BitTorrent, Gnutella) [24] [25]. According to Zipfs law, a small fraction of popular web objects attracts most user requests, which is favourable for the efficiency of small caches [25]. Zipf distributions of finite support for a content catalogue of N contents with request probabilities $Z(r)$ have been determined for the objects popularity ranks $r \in \{1,2,3...,N\}$:

$$Z(r) = \alpha r^{\beta} with : r^{\beta} < 0; \beta = Z(1) = 1/\sum_{r=1}^{N} r^{\beta} > 0 \quad (2)$$

where β is an adaptive shape parameter and α a normalization constant. R represents the users requests with $R = 10^3$ that is generated according to the Zipf distribution function with α set as : $\alpha = 0.75$. A catalogue of $N = 10^2$ contents is then generated, where each content is accompanied with its size, frequency, cost and mobility. These factors are first initialized randomly and then change over time according to α.

We first study the stability of cache hit ratio over time to characterize how the hit ratio changes with the changing of contents distribution in order to study the long term cache performance. Then, we focus on the short-term performance by studying the relationship between cache size and hit ratio.

Regarding to the long term study, we express it by the variation of α Zipf parameter to have different request distributions in order to evaluate the capability of our proposed system against the above strategies in term of maintaining good performance over time. We distribute a data set based on the random changing of α Zipf parameter value. From Figure 6, we can observe that the proposed fuzzy caching strategy and LRU are stable over time unlike FIFO. We also notice that our proposed algorithm shows more advantage with cache size = 600 and a hit ratio approximating 0.8 %.

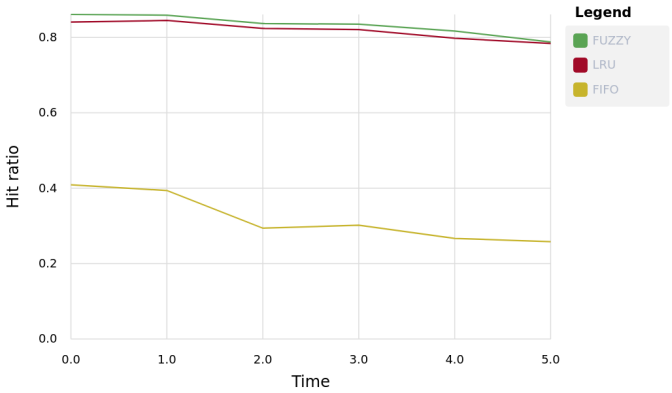

Fig. 6. Cache Hit ratio stability over time

For the short term performance, figure 5 shows hit ratio achieved by our proposed strategy and the other caching strategies introduced above. We can see how the hit ratio varies with cache size measured with UNIT. We noticed also that each time the cache size increases the hit ratio augments and the proposed FCEC system provides a higher cache hit rate for all cache capacity values. When the cache size is small, the performance of LRU is close to our proposed strategy. As the cache size increases, the gap between FCEC strategy and other two caching algorithms increases at first, and then gradually became almost similar with LRU and wobbling for FIFO. At cache capacity C = 700, the cache hit ratio of the three algorithms is increased at around 0.8%. At this point, the cache hit ratio achieved by the mentioned strategies tend to converge because the cache size is high enough to store all the contents.

For the hardware implementation, we migrated the main algorithm into hardware using FPGA. We noticed that the hardware consumes only 12 clock cycles, which can be considered as an advantage regarding to the critical role of the cache in the edge of the network. Figure 7 shows the measurement of energy consumption, the thermal properties, the voltage and the electric current during the run of the fuzzy caching decision system. We notice that a power of P =0.45W is consumed when making the caching decision. Comparing Fuzzy caching strategy with LRU that consumes P =0.9W

[27], it is notable that Fuzzy caching strategy consumes less power.

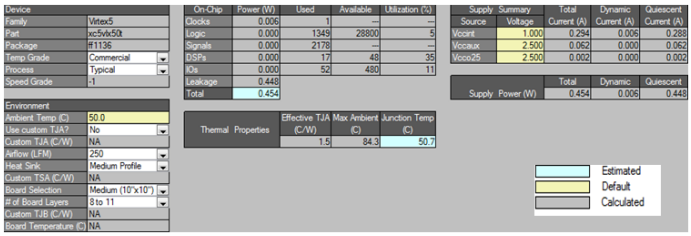

Fig. 7. The values of energy consumption, thermal properties, voltage and current.

V. CONCLUSION

We have presented a novel caching system for edge devices, that combines size, mobility and cost awareness and continuously optimized using fuzzy logic decision maker model of edge caching. The software implementation shows that the proposed mechanisms improves the hit ratio and the hardware implementation reduces the power utilization. As more different applications with various contents migrate into the internet, future mobile edge computing system will experience more variability in requests, content sizes and costs. We believe that the proposed caching strategy can address these challenges. As future work, we aim to extend the proposal to an other model using reinforcement learning with agents that learn with the dynamics of the environment, which will encompass the problem of stochastic dynamics and enhance the quality of caching decision.

REFERENCES

[1] W. Shi ; J. Cao ; Q. Zhang ; Y. Li ; L. Xu, Edge Computing: Vision and Challenges IEEE Internet of Things Journal, Volume: 3 , Issue: 5 , Oct. 2016 pp: 637 - 646

[2] W. Ali , S. M. Shamsuddin, A. s. Ismail, A Survey of Web Caching and Prefetching, Int. J. Advance. Soft Comput. Appl., Vol. 3, No. 1, March 2011 PP:2074-8523.

[3] W. Ali, S.M. Shamsuddin, A.S. Ismail, Intelligent Web proxy caching approaches based on machine learning techniques, Elsevier (2012) pp: 0167-9236.

[4] A. S. M. A. Ibrahim Abdullahi, Ibrahim Badamasi, Cacheskip approach for Information-Centric Network, ARPN Journal of Engineering and Applied Sciences, vol. 11, no. 5, pp. 3413-3418, 2016.

[5] E. H. Miller, A note on reflector arrays (Periodical styleAccepted for publication), IEEE Trans. Antennas Propagat., to be publised.

[6] J. Wang, Fundamentals of erbium-doped fiber amplifiers arrays (Periodical styleSubmitted for publication), IEEE J. Quantum Electron., submitted for publication.

[7] Csar Bernardini, Thomas Silverston, Olivier Festor. A Comparison of Caching Strategies for Content Centric Networking. IEEE Global Communication Conference, Dec 2015, San Diego, United States.IEEE Global Communication Conference. ¡hal-01251968¿ Networking,

[8] K. Cho, M. Lee, K. Park, T. T. Kwon, Y. Choi, S. Pack, "Wave: Popularity-based and collaborative in-network caching for content-oriented networks", Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS), pp. 316-321, Mar. 2012.

[9] Psaras I, Chai WK, Pavlou G. Probabilistic In-Network Caching for Information-Centric Networks. In proceedings of the second edition of the ICN workshop on Information-centric networking. New York, NY, USA, 2012; 5560.

[10] B. Banerjee, C. Tellambura, "Study of mobility in cache-enabled wireless heterogeneous networks", 2017 IEEE Wireless Communications and Networking Conference (WCNC), March 2017.

[11] A. Mahmood ; C. Casetti ; C.F. Chiasserini ; P. Giaccone ; J. Harri, "Mobility-aware edge caching for connected cars", 2016 12th Annual Conference on Wireless On-demand Network Systems and Services (WONS), Cortina d'Ampezzo, Italy, Jan. 2016

[12] F. Zhang et al., "EdgeBuffer: Caching and prefetching content at the edge in the mobilityfirst future Internet architecture", Proc. IEEE 16th Int. Symp. World Wireless Mobile Multimedia Netw. (WoWMoM), pp. 1-9, Jun. 2015.

[13] S. P. Bingulac, On the compatibility of adaptive controllers (Published Conference Proceedings style), in Proc. 4th Annu. Allerton Conf. Circuits and Systems Theory, New York, 1994, pp. 816.

[14] N. Abani, T. Braun, M. Gerla, "Proactive Caching with Mobility Prediction under Uncertainty in Information-centric Networks",ICN '17 Proceedings of the 4th ACM Conference on Information-Centric Networking, Berlin, Germany September 26 - 28, 2017 , PP. 88-97

[15] L. Yao, A. Chen, J. Deng, J. Wang, G. Wu, "A cooperative caching scheme based on mobility prediction in vehicular content centric networks", IEEE Transactions on Vehicular Technology, Volume: 67 , Issue: 6 , June 2018, pp. 5435 - 5444.

[16] C. Jarray, A. Giovanidis, "The effects of mobility on the hit performance of cached d2d networks", 2016 14th International Symposium on Modeling and Optimization in Mobile Ad Hoc and Wireless Networks (WiOpt), pp. 1-8, May 2016.

[17] C. Bernardini, T. Silverston, O. Festor, "A Comparison of Caching Strategies for Content Centric Networking", Proc. IEEE Global Communications Conference (GLOBECOM), pp. 1-6, 2015.

[18] A. Araldo, M. Mangili, F. Martignon, and D. Rossi, Cost-aware caching: Optimizing cache provisioning and object placement in ICN, in Proc. IEEE Global Commun. Conf. (GLOBECOM), Austin, TX, USA, Dec. 2014, pp. 11081113.

[19] Mitchell J, Rizvi S, Ryoo J (2015) A fuzzy-logic approach for evaluating a cloud service provider. In: To The 2015 The 1st International Conference on Software Security and Assurance (ICSSA15), July 27, 2015, Sungkyunkwan University, Korea.

[20] Psaras I, Chai WK, Pavlou G. Probabilistic In-Network Caching for Information-Centric Networks. In proceedings of the second edition of the ICN workshop on Information-centric networking. New York, NY, USA, 2012; pp. 5560.

[21]] M. Ahmed, S. Traverso, P. Giaccone, E. Leonardi, and S. Niccolini,Analyzing the performance of LRU caches under non-stationary traffic patterns, arXiv preprint arXiv:1301.4909, 2013.

[22] D. Rossi and G. Rossini, Caching performance of content centric networks under multi-path routing (and more), Relatorio tecnico, Telecom ParisTech, pp. 16, 2011.

[23] M. Tortelli, D. Rossi and E. Leonardi, ModelGraft: Accurate, Scalable, and Flexible Performance Evaluation of General Cache Networks, Proc.Internat. Teletraffic Congress ITC28, Wrzburg, Germany (2016)

[24] D. Rossi and G. Rossini, Caching performance of content centric networks under multi-path routing (and more), Relatorio tecnico, Telecom ParisTech, pp. 16, 2011.

[25] M. Hefeeda and O. Saleh, Traffic modeling and proportional partial caching for peer-to-peer systems, IEEE/ACM Trans. on Networking 16/6 (2008) 1447-1460

[26] G. Hasslinger, K. Ntougias, F. Hasslinger, "Performance and Precision of Web Caching Simulations Including a Random Generator for Zipf Request Pattern", Proc. 18th MMB Conf. Mnster, vol. 9629, pp. 60-76, 2016.

[27] Q. Zhu, Y. Zhou, "Power aware storage cache management", IEEE Transactions on Computers, vol. 54, no. 5, pp. 587-602, May 2005.

[28] Liao, X. W., Li, Y., Lu, B. "A model for selecting an ERP system based on linguistic information processing". Information Systems, vol 32(7),pp. 10051017, 2007.