International Journal of Future Generation Communication and Networking
Vol. 9, No. 6 (2016), pp. 99-112
http://dx.doi.org/10.14257/ijfgcn.2016.9.6.10

Evaluation of the Caching Placement Algorithms in Named Data Network for Video on Demand Service

Kire Jakimoski[1], Slavcho Chungurski[2] and Sime Arsenovski[3]

[1,2,3]Faculty of Informatics, FON University, Skopje, Republic of Macedonia
[1]kire.jakimoski@fon.edu.mk; [2]slavcho.chungurski@fon.edu.mk;
[3]sime.arsenovski@fon.edu.mk

Abstract

In the existing literature for Information-Centric Networking (ICN) architectures improvement metrics are well researched and studied including the cache hit rate, origin server load reduction, and the reduction in the overall network footprint [1]. Metrics could be optimized using various aspects of caching including the content placement to make a decision which routers should cache the content on a request patch [2-5].

Impact of different ICN caching placement strategies on the main cloud TV performance metrics is deeply researched in this work, and results are presented. Seven cache placement algorithms are evaluated including four cache sizes of 1 GB, 10 GB, 100 GB, and 1 TB.

Keywords*: cache hit ratio, cache placement algorithms, cache size, delay time, Information-Centric Networking, Video on Demand, queuing delay*

1. Introduction

In-network caching is very important in ICN (Information-Centric Networking) architectures [6], and there are a lot of researches in this subject including two main areas, content placement and content replacement [7-8]. Strategy of content replacement [9] decides where an object should be cached on the request path across the routers. Per example, it could be the sequence of routers from the content origin to the requesting client. In the simulation scenario of this work seven particular content placement strategies are researched:

- LCE (Leaving a copy everywhere);
- LCD (Leaving a copy only on the immediate downstream router when there is a hit on an upstream router);
- Rand (Leaving a copy on a randomly selected router along the requested path);
- Prob (constant probability of caching on each router);
- Pprob (Probability of caching on a router is function of the distance from the origin server and the shared storage capacity of the path);
- Centrality (it uses centrality-based measures where the router with the largest value keeps a copy);
- Cross (hybrid approach where the probability that a router caches an object is function of the router and the popularity of the content).

2. Description of the Algorithms

LCE (Leave Copy Everywhere) caching placement algorithm is typical mode of operation that is currently in use in most of the multi-level caches [10]. If some hit happens at a level *l* cache or the origin server, a copy of the requested object is cached in all intermediate caches (levels *l*-1, …, 1) on the path from the location of the hit down to the requesting client [11].

LCD (Leave Copy Down) caching placement algorithm enables new copy of the requested object to be cached only at the (*l*-1)-level cache. That is the cache is located directly below the location of the hit on the path to the requesting client [12]. If LCD algorithm is compared to LCE, it could be stated that it is more "conservative" than LCE because it involves multiple requests to bring an object to a leaf cache, with each request advancing a new copy of the object one hop closer to the client.

Random caching placement algorithm selects one of nodes on the trajectory randomly [3]. In this scheme a content file is published on a randomly selected node and its advertisement packet is flooded through the network in order potential field to be constructed. A randomly selected user then generates a request for the content file, and the query is routed to the node where the content file is published. So, the request is forwarded randomly until it moves into the potential field.

Prob caching placement algorithm is randomized version of LCE algorithm. On the path from the location of the hit down to the requesting client each intermediate cache is appropriate to store a copy of the requested object. Local copy with probability *p* is kept from the intermediate cache, and as a result of this, replacement algorithm is invoked. Copy is not kept from the intermediate cache with probability *1-p*. The above mentioned algorithms are presented in Figure 1.

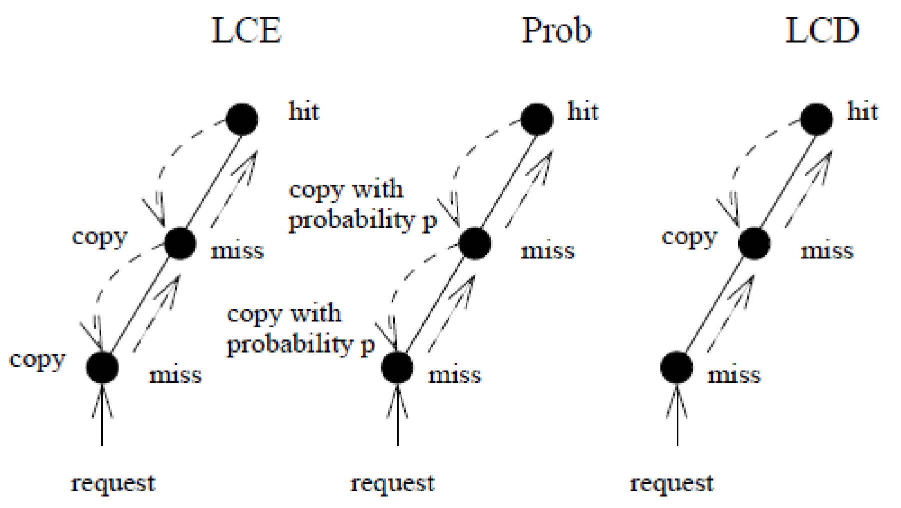

Figure 1. Operation of the LCD, LCE, and Prob Cache Placement Algorithms

Pprob caching algorithm is probabilistic scheme for distributed content caching along a patch of caches. This algorithm approximates the caching capability of some path and caches contents probabilistically. As a result of this it leaves caching space for other flows sharing the same path, and fairly multiplex contents of various flows among caches of a shared path [6].

Centrality-based caching algorithm improves the caching gain and eliminates the ambiguity in the performance of the simplistic random caching strategy. Number of times a specific node lies on the content delivery path is measured between all pairs of nodes in the network topology. So, if some node lies along a high number of content delivery paths, it is more probable to get a cache hit [13]. The cache replacement rate is reduced by caching only at those more important nodes while still caching content where a cache hit is most likely to occur.

Cross caching placement algorithm is caching scheme that uses a cross-layer design in order to cache contents in a few selected routers based on the correlation of content popularity and the network topology. According to this scheme, it exploits available information at both network and application layers and intends to improve the cache hit rate and reduce the overall network traffic [14].

3. Simulation Methods and Results

In this work Video on Demand through cloud-TV is used for evaluating the seven caching placement algorithms mentioned in the previous section. QoE (Quality of Experience) quantifies either objectively by metrics measurements the played video quality, or its subjective perception by the end user. The most important QoE metrics are buffering ratio, startup latency, and average bit rate. These metrics have impact on the user engagement [15]. Buffering ratio is actually the fraction of the session time spent in buffering and it is equal to:

$$Buffering\ ratio = \frac{BufferingTime}{TotalViewTime}$$

Average bit rate could be presented with the following equation:

$$Average\,bitrate = \frac{TotalBytesDownloaded}{TotalPlayTime}$$

In this work the emphasis is put on the analysis of the delay or join time between the request and video start, and it is equal to:

$$Delay = StartTimeOfPlay - \mathrm{Re}questTime$$

In this equation *RequestTime* variable refers to the time when the user requests the video from the cloud-TV server. *StartTimeOfPlay* variable refers to the time when the requested video plays after the user sends video request message to the cloud-TV server.

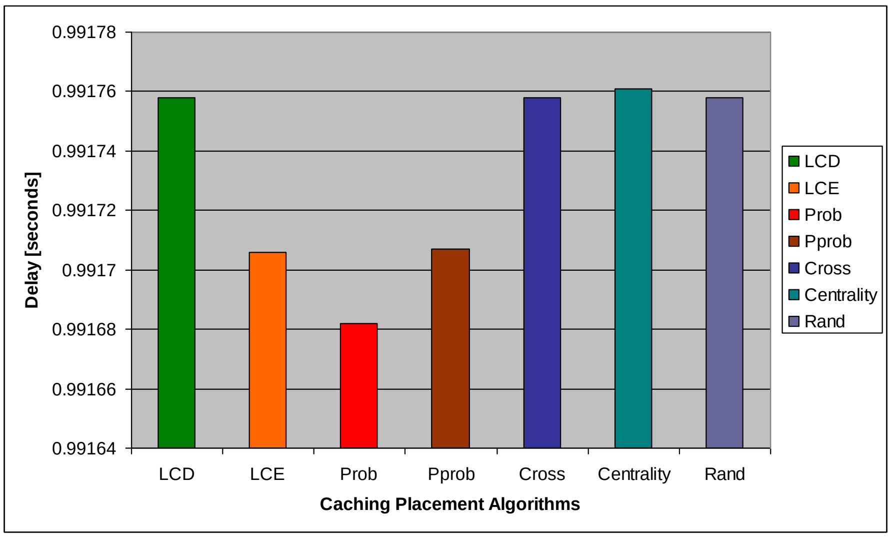

Figure 2. Delay Time of the Caching Placement Algorithms

Figure 2 presents the delay time of all seven cache placement algorithm mentioned before. According the presented delay results in Fig. 2 we can conclude that Prob algorithm gives the best results for delay if we compare it with other six algorithms. It has the lowest value of 0.991682 seconds among all seven presented caching placement algorithms.

Cache Hit Ratio of the Prob algorithm. Average cache hit ratio across all routers could be expressed by the following equation:

$$HitRate = \frac{\sum_{r=1}^{R} \frac{Hit_r}{Hit_r + Misses_r}}{R}$$

In the above equation Hit_r presents the number of requests that router r was capable to serve from its cache, and $Misses_r$ presents the number of requests that the router forwarded upstream because of the cache miss.

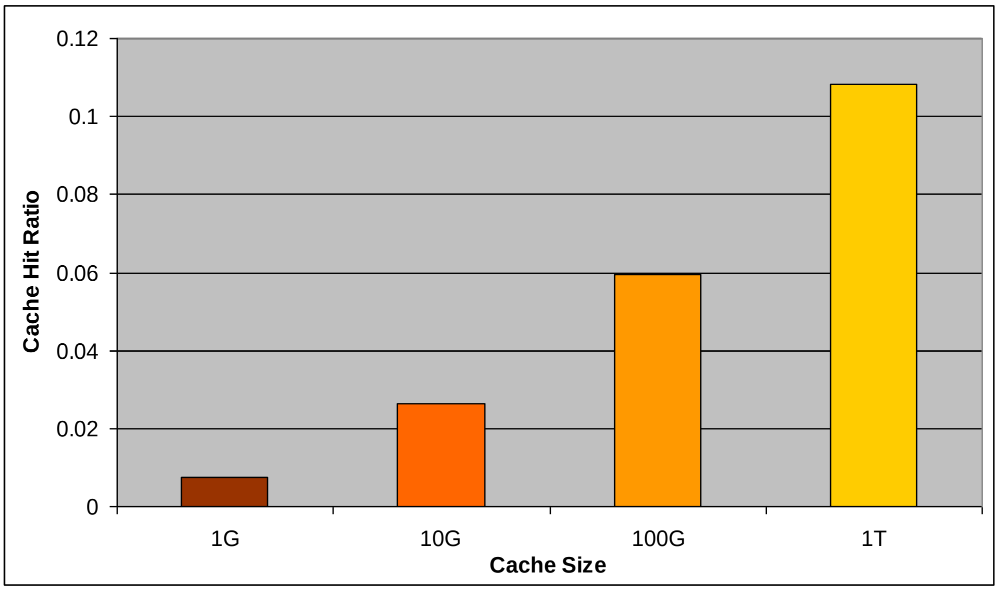

Figure 3. Cache Hit Ratio of the Prob Algorithm

Figure 3 presents obtained results for the cache hit ratio of the Prob algorithm for different cache sizes starting from 1 GB to 1 TB. It is clearly shown that when the cache size is increased from 1 GB to 1 TB, the cache hit ratio also increases.

In the next diagrams we will present the cache hit ratio of all seven caching placement algorithms for cache sizes of 1G, 10G, 100G, and 1T. In this way we will compare the cache hit ratio results of the Prob algorithm to other algorithms for different cache sizes.

Figure 4 presents the results of the cache hit ratio of all algorithms when the cache size is 1G. As it is shown in the diagram Prob algorithm has value of 0.007. The same value has also LCE, Rand, Pprob and Cross algorithm. LCD algorithm has low value comparing to previous algorithms and it is around 0.003. And the lowest cache hit ratio in this case when the cache size is 1G has Centrality algorithm with value less than 0.002.

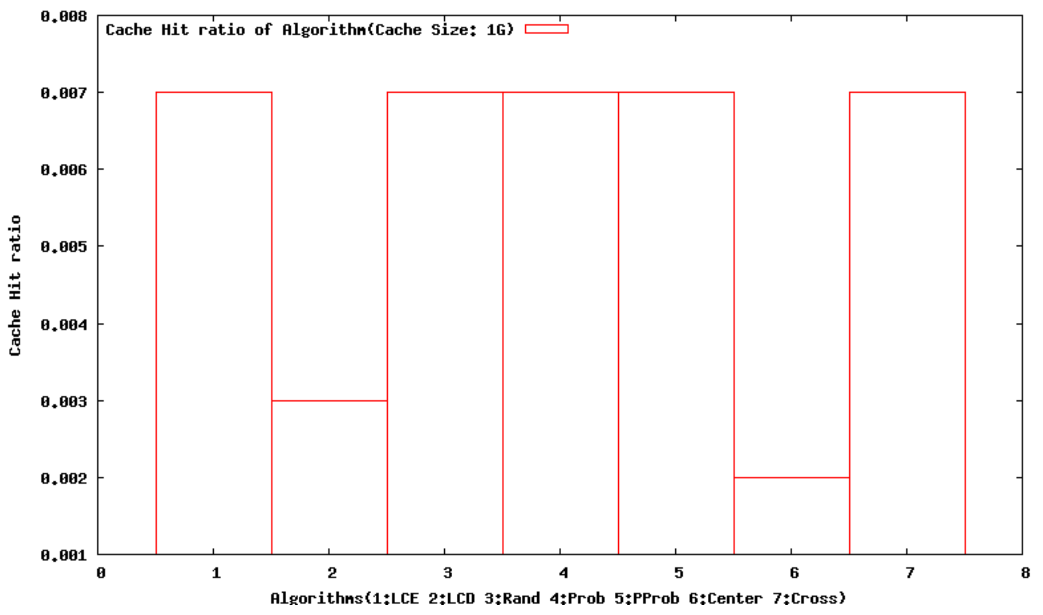

Figure 4. Cache Hit Ratio of the Caching Placement Algorithms for 1 GB of Cache Size

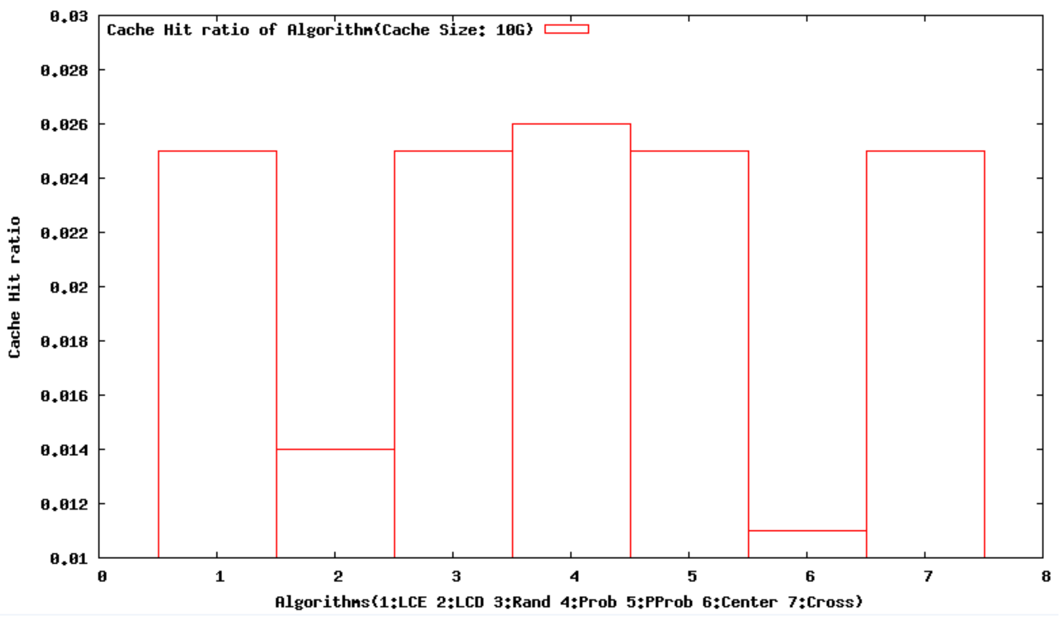

Figure 5. Cache Hit Ratio of the Caching Placement Algorithms for 10 GB of Cache Size

In Fig. 5 the cache size is increased to 10G. We can already notice here that the cache hit ratio of the Prob algorithm give the best performances. It has the highest value of cache hit ratio of 0.026. LCE, Rand, Pprob and Cross algorithm give similar results for cache hit ratio like in the previous case when the cache size was 1G. Centrality algorithm again gives the lowest value for cache hit ratio.

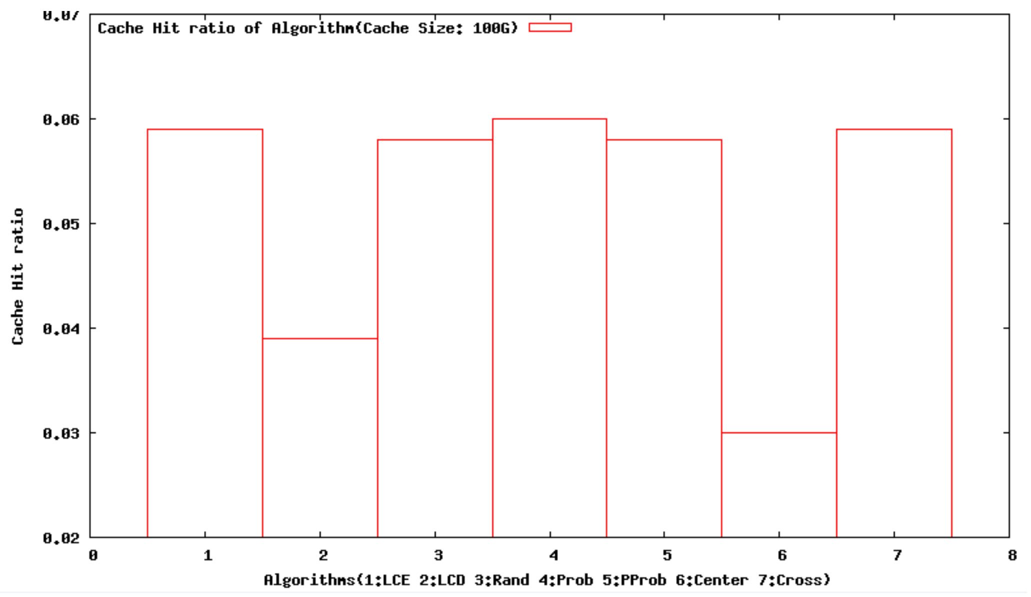

Figure 6. Cache Hit Ratio of the Caching Placement Algorithms for 100 GB of Cache Size

Figure 6 presents the results of the cache hit ratio when the cache size is 100 GB. We can conclude here that Prob algorithm again has the largest value of cache hit ratio of all seven algorithms, similarly to Figure 4. This time it has value of around 0.06 cache hit ratio. The lowest cache hit ratio gives Centrality algorithm.

Figure 7 presents the cache hit ratio of all seven caching placement algorithms when the cache size is 1 TB. Prob algorithm has the highest value for the cache hit ratio comparing to other six algorithms, while the LCE and cross algorithms have cache hit ratio that is little less than the Prob algorithm. Centrality algorithm again gives the lowest results for the cache hit ratio.

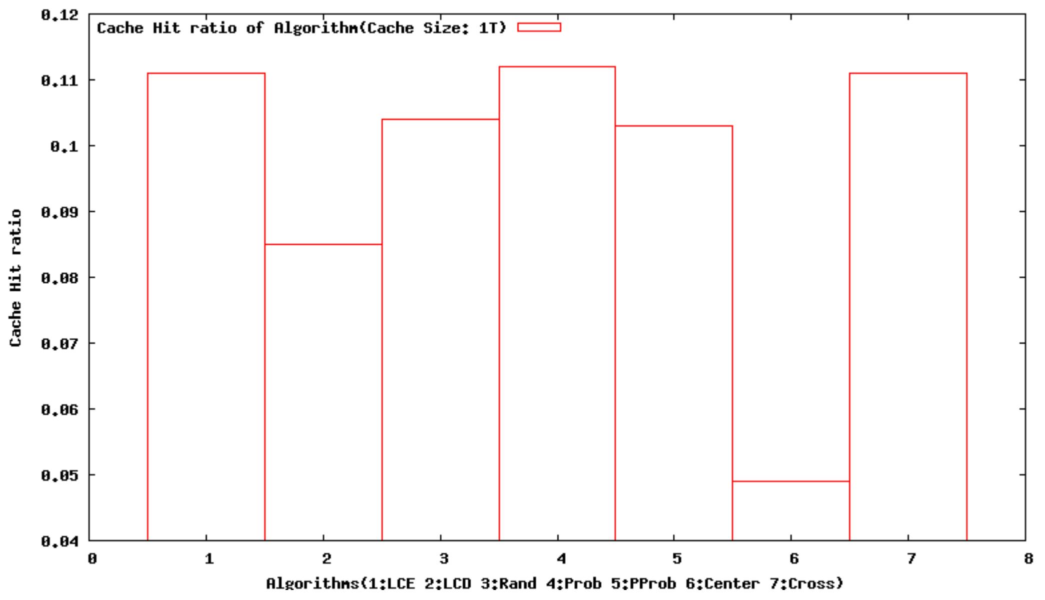

Figure 7. Cache Hit Ratio of the Caching Placement Algorithms for 1 TB of Cache Size

Generally in all diagrams from Figure 4 to Figure 7, the cache hit ratio increases in all cases for all seven caching placement algorithms as the cache size increases. We can also conclude that the Prob algorithm gives the best results for cache hit ratio and Centrality algorithm the worst.

Total Reduction in the Network Footprint. Total reduction in the network footprint is in fact the total number of byte-hops that are saved via in-network caching. It can be expressed with the following equation:

$$TraffiReduction = \frac{\sum_{q=1}^{V} (HopsNoCache_q - HopsICN_q) \times Bytes_q}{\sum_{q=1}^{V} HopsNoCache_q \times Bytes_q}$$

In the above equation *HopsNoCache*$_q$ refers to the number of hops the request would take to get to the nearest origin server. *HopsICN*$_q$ presents the number of hops that the request would take to reach the closest video content provider either an ICN cache or the origin server.

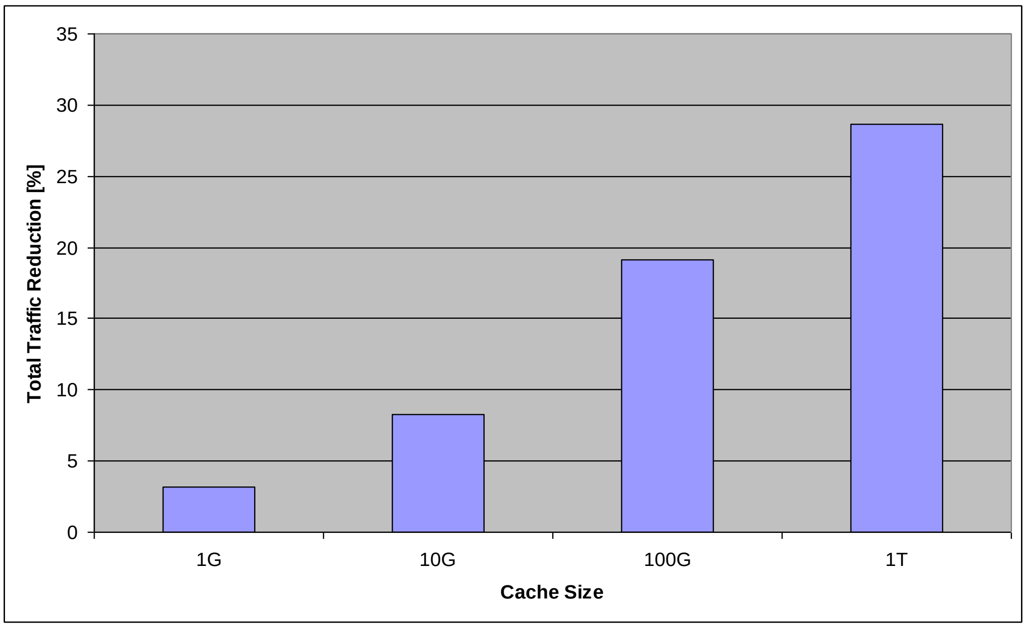

Figure 8. Total Reduction in the Network Footprint of the Prob Cache Placement Algorithm

Figure 8 depicts the results of the total reduction in the network footprint of the Prob cache placement algorithms. Results are given for different sizes of cache size from 1G to 1T. For 1G cache size total traffic reduction is 3.20123, for 10G it is 8.23421, for 100G – 19.1223, and for 1T it is 28.6098. So, as the cache size increases, we can notice that total reduction in the network footprint also increases from 3.20123 for 1G to 28.6098 for 1T.

Figure 9 presents results for the reduction in the network footprint for all algorithms when the cache size is 1G. We can notice here that the Prob algorithm has the highest value of reduction in the network footprint together with the Rand algorithm. Pprob, LCE, and Cross algorithms have also high reduction in the network footprint. The lowest reduction in the network footprint gives Centrality algorithm.

Figure 10 shows the results for the reduction in the network footprint for all algorithms when the cache size is 10G. Prob algorithm again gives the highest value of reduction in the network footprint together with the Rand algorithm. Centrality algorithm again gives the lowest value for the reduction in the network footprint. Comparing to Figure 9, all seven algorithms have higher values for the reduction in the network footprint, because the cache size is increased from 1G to 10G.

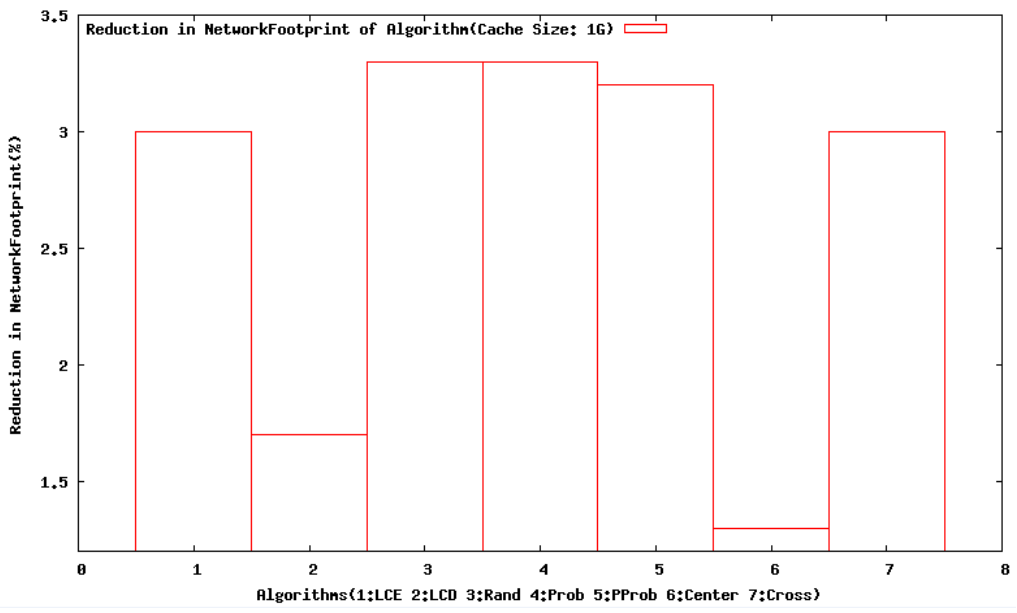

Figure 9. Total Reduction in the Network Footprint of the Cache Placement Algorithms for cache Size of 1 GB

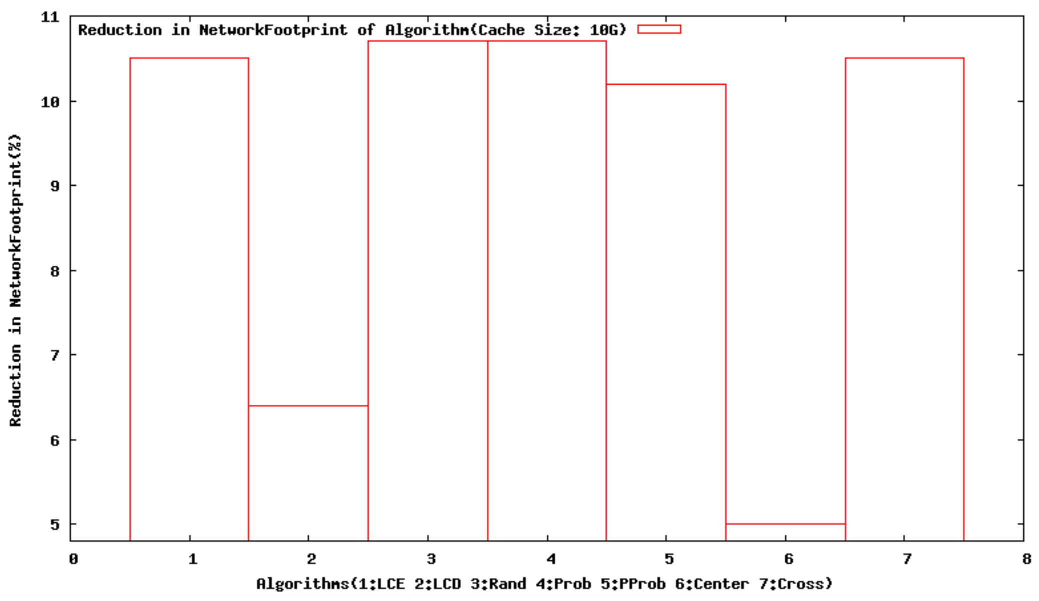

Figure 10. Total Reduction in the Network Footprint of the Cache Placement Algorithms for Cache Size of 10 GB

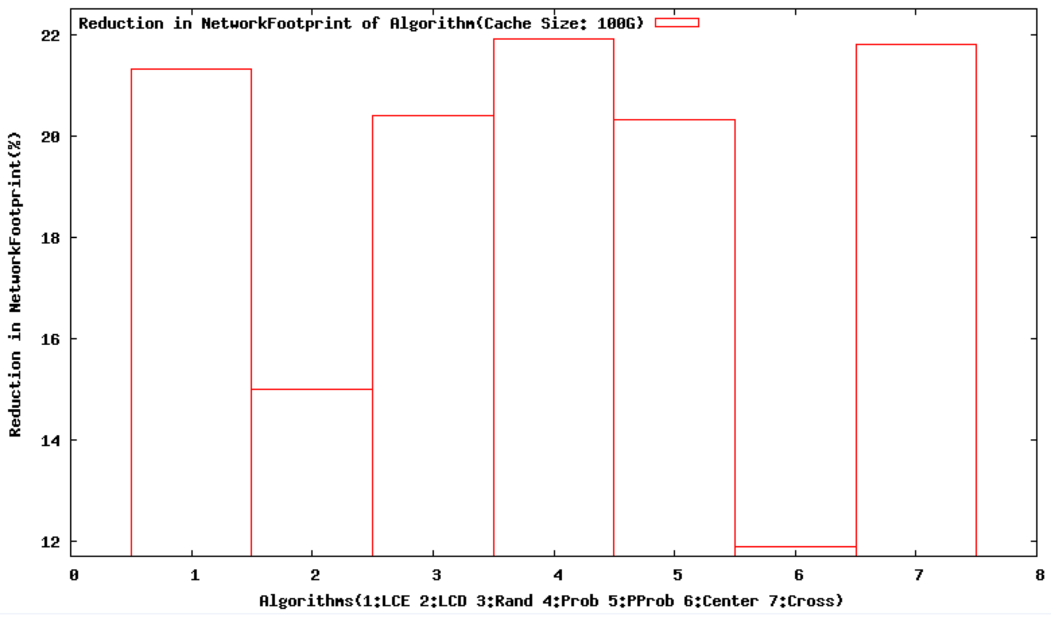

Figure 11. Total Reduction in the Network Footprint of the Cache Placement Algorithms for Cache Size of 100 GB

Figure 11 depicts total reduction in the network footprint for all algorithms when the cache size is 100G. Prob algorithm again gives the highest reduction in the network footprint and the Centrality algorithm gives the lowest reduction.

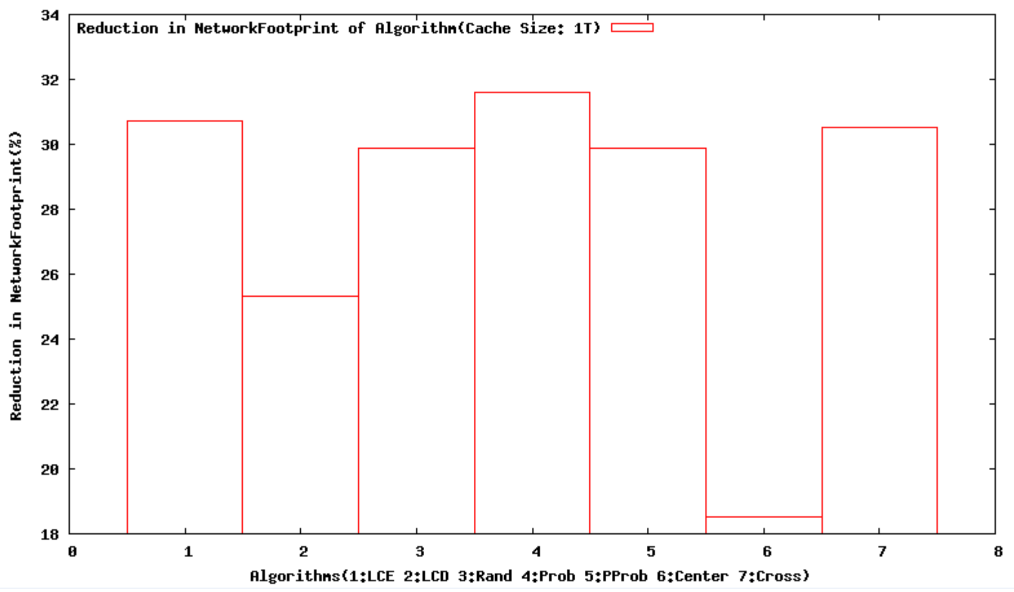

Figure 12. Total Reduction in the Network Footprint of the Cache Placement Algorithms for Cache Size of 1 TB

Figure 12 confirms the previous attitude of the algorithms. In this case again Prob algorithm gives the highest value for reduction in the network footprint, while the Centrality algorithm gives the lowest result.

Hence, we can generally conclude that from Figure 9 to Figure 12 as cache size increases, total reduction in the network footprint also increases for all seven presented algorithms. Prob algorithm gives the highest reduction in the network footprint, while the Centrality algorithm gives the lowest reduction.

Reduction in the Server Load. Reduction in load on the origin servers is very important performance metrics and it could be defined using the following equation:

$$Rsl = 1 - \frac{NumCached\,\mathrm{Req}}{V}$$

In the above equation *NumCachedReq* presents the number of requests that were served from *any* in-network cache. On the other hand, the variable *V* presents all video requests.

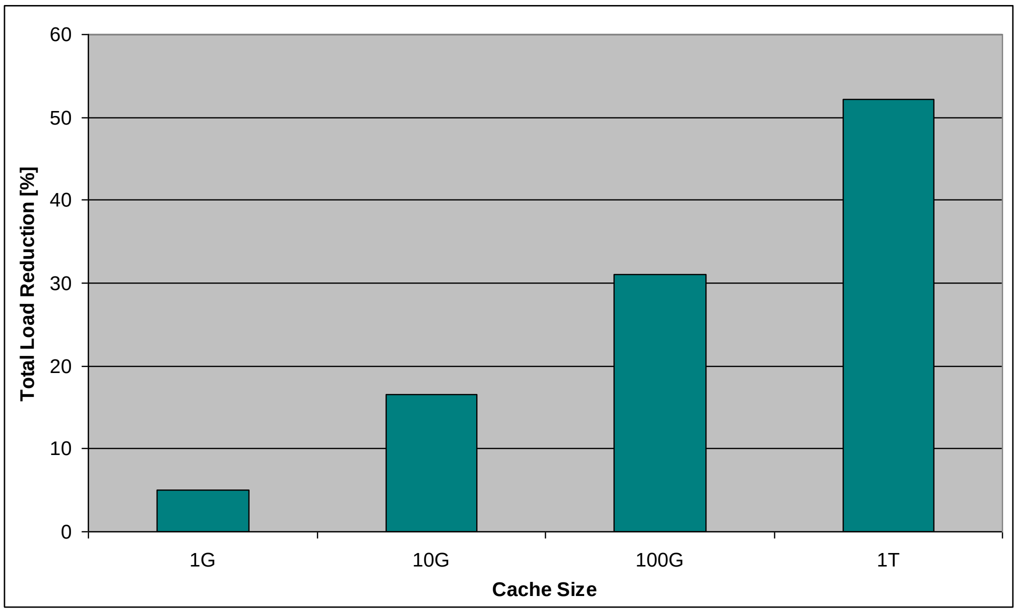

Figure 13. Reduction of Server Load for the Prob Cache Placement Algorithm

Figure 13 presents the results of the reduction in the server load of the Prob cache placement algorithm for different cache sizes. It is clearly shown that as the cache size increases, the reduction in the server load is also increased. For 1G cache size the reduction in the server load is 4.9908%, for 10G cache size it is 16.5067%, for 100G cache size it is 31.1048, and for 1T cache size it is 52.2098.

In the following diagrams we will present the values of the reduction in the server load for all seven cache placement algorithms for different cache sizes. We are doing this to compare the results of the Prob algorithm to other six cache placement algorithms.

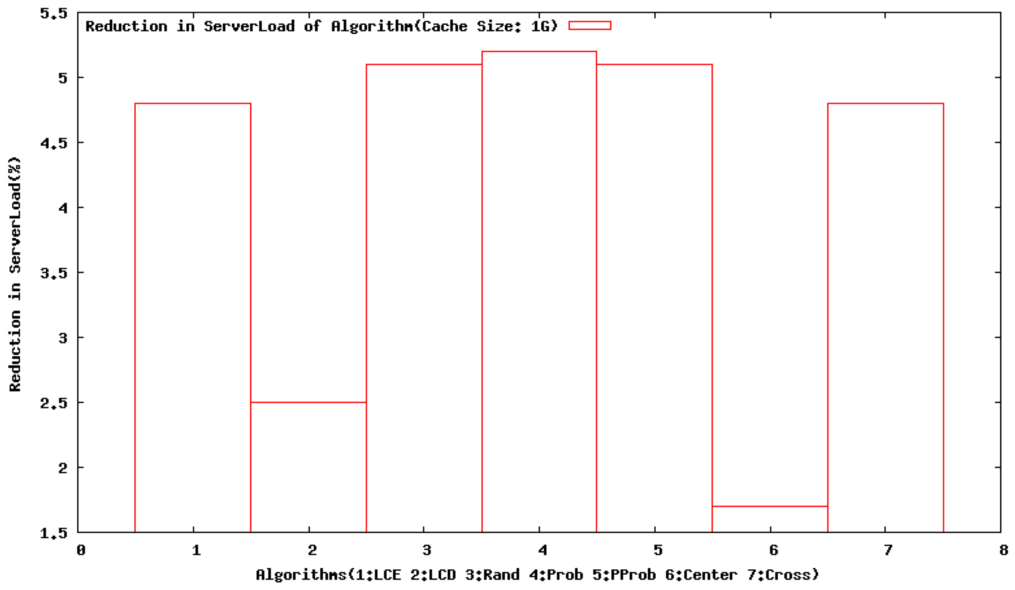

Figure 14. Reduction in the Server Load for all Seven Algorithms when the Cache Size is 1G

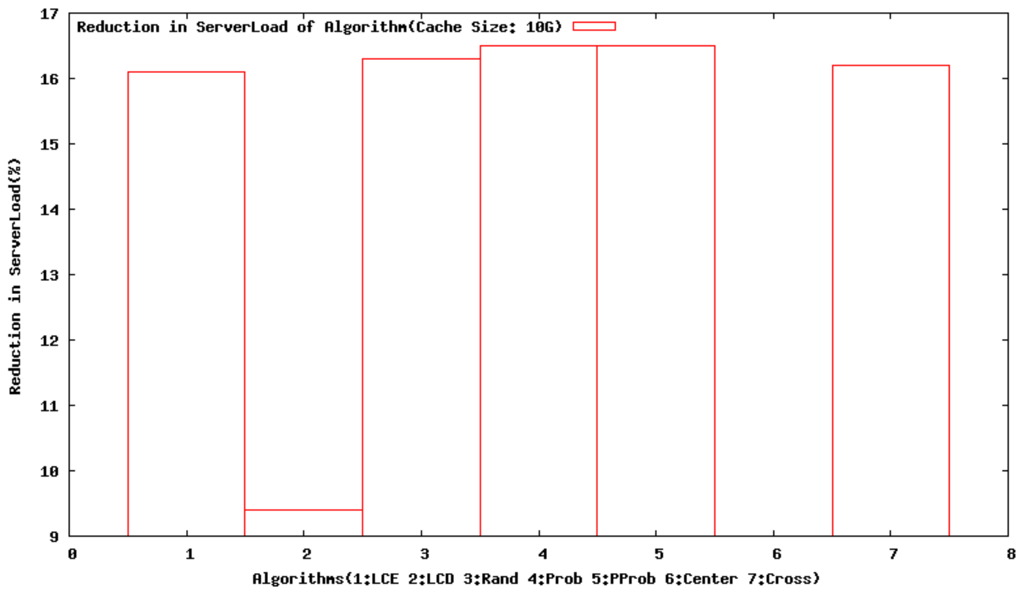

Figure 15. Reduction in the Server Load of all Seven Algorithms for Cache Size of 10G

Figure 14 presents the results of the reduction in the server load expressed in % for all seven algorithms when the cache size is 1G. It is obvious that the reduction in the server load is largest when Prob algorithm is used, and the Centrality algorithm gives the lowest reduction in the server load. So, in this case Prob algorithm gives the most favorable results because higher reduction in the server load gives better overall performances. Anyway, except LCD and Centrality algorithms, results of the other algorithms compared to the Prob algorithm are almost similar.

Figure 15 shows the results of the reduction in the server load for all seven algorithms when the cache size is 10G. Here we can see that the reduction in the server load of the Prob algorithm is again the highest together with the Pprob algorithm - around 16.5%. Centrality algorithm continues to give the lowest result, so in this case its value for reduction in the server load is lower than 9%.

Figure 16 presents the results of the reduction in the server load when the cache size is 100G. It is clearly shown that Prob algorithm gives the highest value of around 34% for the reduction in the server load.

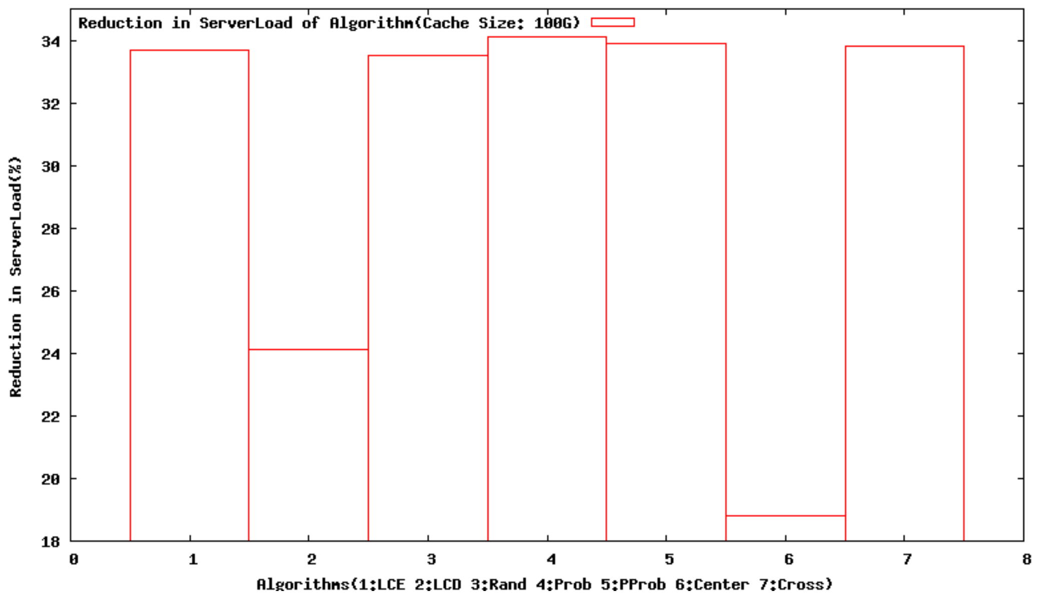

Figure 16. Reduction in the Server Load of all Seven Algorithms for Cache Size of 100G

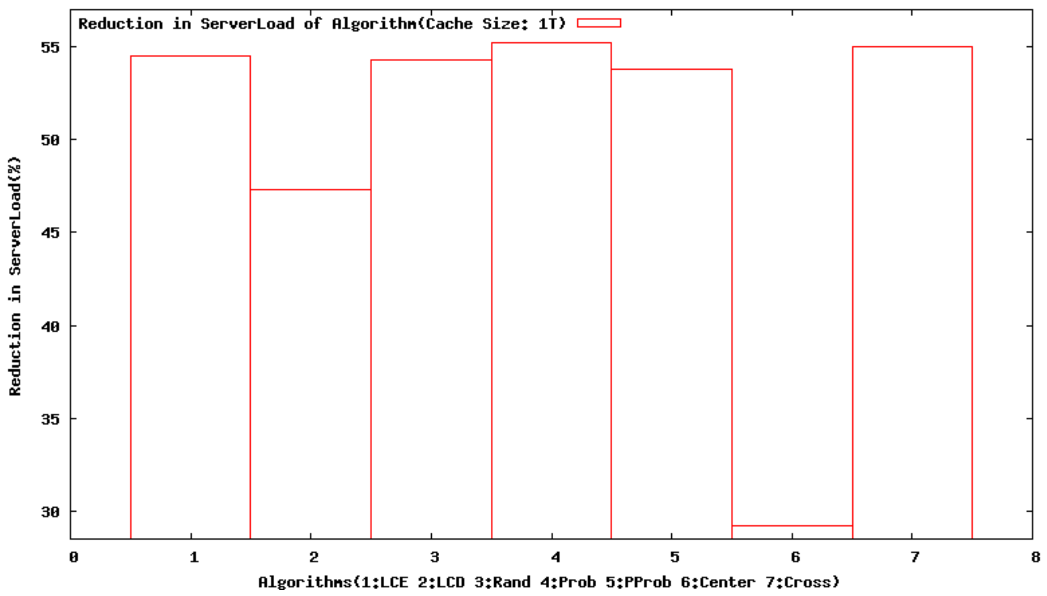

Figure 17. Reduction in the Server Load of all Seven Algorithms for Cache Size of 1T

Figure 17 shows reduction in the server load results for all seven algorithms when the cache size is 1T. From the presented results it is obvious that again Prob algorithm has the greatest value for reduction in the server load of around 55%. Centrality algorithm gives the lowest value from all of the presented algorithms, and it is less than 30%.

From the presented diagrams from Figure 14 to Figure 17, generally, we can conclude that the reduction in the server load is increased when the cache size is increasing for all seven algorithms. We can also confirm that the server load of the Prob algorithm is larger than other algorithms for each of the cache sizes. So, we can conclude that the Prob algorithm gives the best results when we are analyzing the performance metric *reduction in the server load*.

4. Conclusion

In this work we have evaluated seven cache placement algorithms, LCD, Prob, PProb, LCE, Cross, Centrality, and Rand on a cloud-TV network in order to understand better the delay time for each of them. From the obtained results, it can be concluded that the best results for time delay gives the Prob algorithm. The delay time was lowest when Prob algorithm was used in the experiments.

After this part of the work, we have examined the performance metrics of cache hit ratio, total reduction in the network footprint, and reduction in the server load for the previously selected algorithm with the lowest time delay – Prob algorithm. Additionally, the results of this algorithm are compared to the other six algorithms. In this context, we analyzed and gave conclusions particularly for each of the presented diagrams.

Generally, it can be concluded that the Prob algorithm, which was the algorithm with the lowest time delay, gives excellent results in the examined performance metrics. It gives the best results for the cache hit ratio, total reduction in the network footprint, and reduction in the server load for different cache sizes.

References

[1] S. K. Fayazbakhsh, "Less pain, most of the gain: Incrementally deployable icn", ACM SIGCOMM Computer Communication Review, vol. 43, no. 4, **(2013)**.

[2] W. K. Chai, "Cache "less for more" in information-centric networks", NETWORKING 2012, Springer Berlin Heidelberg, **(2012)**, pp. 27-40.

[3] S. Eum, "CATT: potential based routing with content caching for ICN", Proceedings of the second edition of the ICN workshop on Information-centric networking, ACM, **(2012)**.

[4] N. C. Laoutaris and I. Stavrakakis, "The LCD interconnection of LRU caches and its analysis", Performance Evaluation, vol. 63, no. 7, **(2006)**, pp. 609-634.

[5] N. Laoutaris, S. Syntila and I. Stavrakakis, "Meta algorithms for hierarchical web caches", Performance, Computing, and Communications, 2004 IEEE International Conference on IEEE, **(2004)**.

[6] I. Psaras, W. K. Chai and G. Pavlou, "Probabilistic in-network caching for information-centric networks", Proceedings of the second edition of the ICN workshop on Information-centric networking, ACM, **(2012)**.

[7] G. Zhang, Y. Li and T. Lin, "Caching in information centric networking: a survey", Computer Networks, vol. 57, no. 16, **(2013)**, pp. 3128-3141.

[8] Y. Kim and I. Yeom, "Performance analysis of in-network caching for content-centric networking", Computer Networks, vol. 57, no. 13, **(2013)**, pp. 2465-2482.

[9] S. Podlipnig and L. Böszörmenyi, "A survey of web cache replacement strategies", ACM Computing Surveys (CSUR), vol. 35, no. 4, **(2003)**, pp. 374-398.

[10] J. Li, B. Liu and H. Wu, "Energy-efficient in-network caching for content-centric networking", Communications Letters, IEEE, vol. 17, no. 4, **(2013)**, pp. 797-800.

[11] N. Laoutaris, H. Che and I. Stavrakakis, "he LCD interconnection of LRU caches and its analysis", Performance Evaluation, vol. 63, no. 7, **(2006)**, pp. 609-634.

[12] Y. Li, "A chunk caching location and searching scheme in content centric networking", Communications (ICC), 2012 IEEE International Conference on IEEE, **(2012)**.

[13] W. K. Chai, "Cache 'less for more' in information-centric networks", NETWORKING 2012, Springer Berlin Heidelberg, **(2012)**, pp. 27-40.

[14] W. Wang, "CRCache: exploiting the correlation between content popularity and network topology information for ICN caching", Communications (ICC), 2014 IEEE International Conference on IEEE, **(2014)**.

[15] S. S. Krishnan and R. K. Sitaraman, "Video stream quality impacts viewer behavior: inferring causality using quasi-experimental designs", Networking, IEEE/ACM Transactions, vol. 21, no. 6, **(2013)**, pp. 2001-2014.

Authors

Kire Jakimoski, he received his B.Sc. degree in the field of Telecommunications from the Military Academy "Mihailo Apostolski" in Skopje, R. Macedonia in 2002, M.Sc. degree in Electrical Engineering in the field of Telecommunications from the Ss. Cyril and Methodius University in Skopje, R. Macedonia in 2007, and Ph.D. in technical sciences from the Ss. Cyril and Methodius University in Skopje, R. Macedonia in 2013. From 2002 to 2006 he works as an Officer for Telecommunications in the Ministry of Defense in the Republic of Macedonia. From January, 2006 to March, 2012 he works as an adviser for information security in the Directorate for Security of Classified Information in the Republic of Macedonia. From March, 2012 he is with the Faculty of Informatics, FON University in Skopje. Also, he is an author/co-author of above 30 published research papers and one book. He is now an Assistant Professor and Vice Dean at the Faculty of Informatics, FON University in Skopje, Macedonia. His research interests include Wireless and Mobile Networks, Heterogeneous Wireless Networks, Computer Networks, Digital Telecommunications, Information Security.

Sime Arsenovski, he is a Full Professor at the Faculty of Informatics, FON University, Skopje, Republic of Macedonia.

Slavcho Chungurski, he is an Associate Professor and Dean of the Faculty of Informatics, FON University, Skopje, Republic of Macedonia.