Journal of Engineering Research and Education
Volume 12, 2020 [1-12]

EdgeTouch Caching Strategy for Video on Demand Data Contents in Named Data Networking

Muhammad Ali Naeem[1*], Shahrudin Awang Nor[1] and Suhaidi Hassan[1]

[1]*InterNetWorks Research Laboratory School of Computing, Universiti Utara Malaysia.*

ABSTRACT

Named Data Networks (NDNs) agree to cache contents inside the network in order to offer efficient delivery of requested data. The recent works on in-network cache intent to focus on minimizing the homogeneous content to be cached for the betterment of the cache hit ratio, which could not show the way to save considerable bandwidth. On the other hand, the caching of the same content at more locations projected low diversity. Moreover, it will increase the frequent caching operations, i.e., cache placement and replacement, causing supplementary power consumption, which should be kept away from energy-limit in data delivery environments, e.g., wireless networks. In this paper, a content disseminated caching strategy alongside the data routing path called Edge Touch Caching Strategy (ETCS) is proposed. The main goal of ETCS is to reduce the caching of homogeneous contents at numerous locations during data dissemination along the data delivery path to increase content diversity. This research also captured the effect of the cache hit ratio through a comparison between the ETCS caching strategy with several state-of-art caching strategies in NDN. The results show that the ETCS strategy can increase the content diversity along with the cache hit ratio even for different cache size it performs better than the existing state of the art strategies.

Keywords: Leave Copy Everywhere, Named Data Networking, Caching.

1. INTRODUCTION

In the current time customers largely use videos over the Internet. In the United States, the national average of daily video consumption on the Internet is about 15 minutes. These 15 minutes amount to 16 Gigabytes (GB) per month per customer which accounts for 66% of all Internet traffic (Zhang, Li, and Lin, 2013). Game consoles, smartphones, tablets, and smart TV are all used to stream video and by 2017 the number of smart devices will be triple the number of people on earth. In particular, web-enabled TV has a four-fold increase. Mobile video will increase 14-fold from 2015 to 2019 (Zhang *et al.,* 2013). A total of 100 hours of videos will be uploaded every minute to YouTube. Netflix, YouTube, and many other services offer an unlimited selection of video files. This consumption of the Internet is depicted in Figure 1 in a more detailed form. Moreover, file sizes keep increasing in the last years with the emergence of high definition technologies such as high definition television (720p or 1080p) or ultra high definition television (4k). Therefore, the 15 minutes of daily video consumption will become hours of consumption. The 16 GB of American video traffic per month will become 600 GB (Carofiglio, Gallo, and Muscariello, 2016). This is a serious problem for the Internet because it is a network built over an outdated communication paradigm. Unlike traditional broadcast networks that send one title to millions across the network at one time, the Internet transmits the same videos many times over. In fact, 10% of content represents 90% of Internet traffic (Abani, Braun, and Gerla, 2017).

*Corresponding Author: *malinaeem7@gmail.com*

Congestion is getting out of control and new mechanisms are needed to fulfil the Internet quality requirements. In this context, temporary storage servers have been used across the Internet to serve the requested contents many times. These temporary storage servers are called caching systems or simply caches.

More recently, Named Data Networking (NDN) has appeared as architecture to replace the Internet architecture (Salsano, Blefari-Melazzi, Detti, Morabito, and Veltri, 2013). With NDN, caches will be available at every node and the network will become a network of caches (Katsaros, Xylomenos, and Polyzos, 2011). Caches have never been deployed on a large scale. Caching contents locally needs the best possible position of caches. Moreover, having caches at every node makes the cache management problem very complicated such as it increases the available space for storing contents but the complexity to manage them increases at the same time. To manage caches, caching strategies are used. Caching strategies decide what, when, and where to store the contents. A number of caching strategies have been proposed such as Leave Copy Everywhere (Zhang *et al.,* 2013), Leave Copy Down (Ren *et al.,* 2014), MAGIC (Ren *et al.,* 2014), ProbCache (Psaras, Chai, and Pavlou, 2012), Cache Less For More (Chai, He, Psaras, and Pavlou, 2013), and some others, for example off-path caching strategies (Amadeo, Campolo, Molinaro, and Ruggeri, 2014) and on-path caching strategies (Ahlgren, Dannewitz, Imbrenda, Kutscher, and Ohlman, 2012; Wählisch, Schmidt, and Vahlenkamp, 2013). However, it is unclear which caching strategy fits best for every possible scenario.

2. RELATED STUDY

2.1 Leave Copy Everywhere (LCE)

Caching is the most central module of all NDN architectures. It plays a great role to implement the main idea of NDN (César Bernardini, Silverston, and Festor, 2015). Caching is used to get better data accessibility utilization of network that directly affects the efficiency of a network. Caching is a temporary storage that embeds in network routers to store a copy of the requested data for a specific period. Any node (router) holding a copy of a requested object can satisfy the primary user and subsequent request for the same content (Carofiglio, Morabito, Muscariello, Solis, and Varvello, 2013). In caching, the user request received by the routers (nodes) directly in response to the request the routers send a locally cached copy to the user through following the back path and a copy of required data is cached along the data delivery path (Wang *et al.,* 2014). It generally uses the router's cache. This caching strategy in NDN caches data contents along the data downloading path during data transmission. This strategy is used to store a copy of the requested data at each router along the delivery path from the source to the user (Eum, Nakauchi, Murata, Shoji, and Nishinaga, 2012). According to LCE, when a user sends a request for his desired data then the source respond to the user by sending a copy of locally cached data as shown by Figure 1. As a result, this strategy increases the availability of contents to fulfil the requirements of subsequent user's requests but in turn, it increases the content redundancy and caching as well as eviction operations. Moreover, it maximizes resource usability and reduces the cache hit rate.

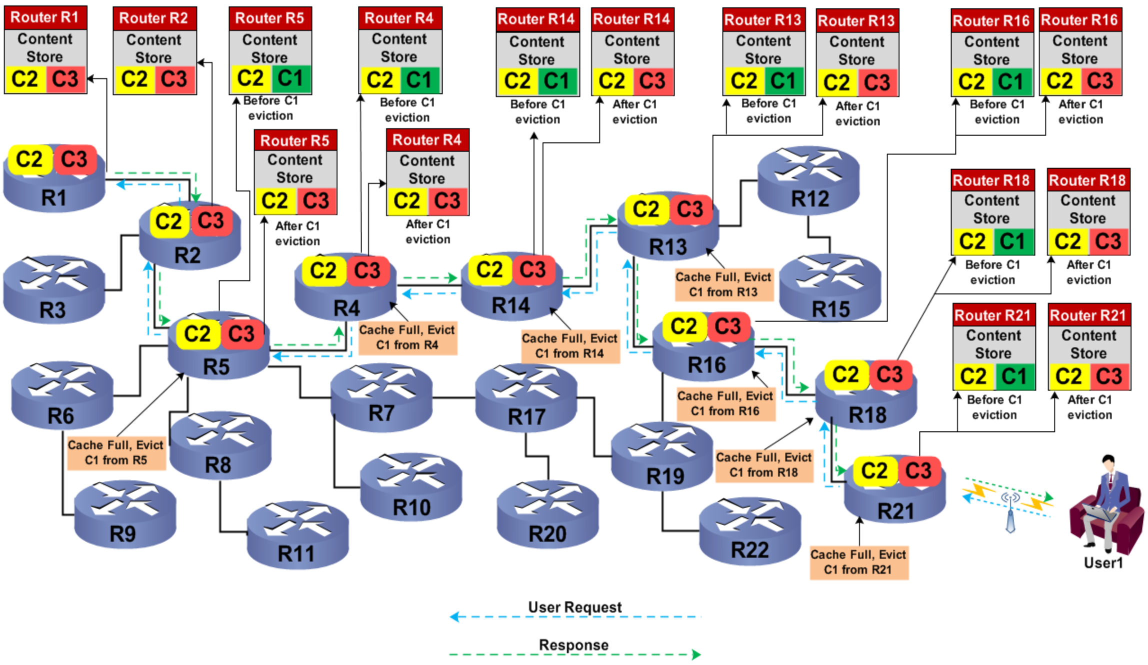


Figure 1. Leave Copy Everywhere Caching Strategy.

2.2. Probabilistic Caching

Probabilistic caching (Psaras *et al.,* 2012) is a renowned content deployment strategy in which the research community takes more interest because of its flexible probabilistic value (Fang, Yu, Huang, Liu, and Liu, 2015). The probabilistic value is defined by the algorithm and it has the predefined probabilistic values to cache the subscriber's requested contents along the data downloading path (Xylomenos *et al.,* 2014). The caching operation for each content is taken individually by all the routers along the routing path without the involvement of cooperation between the nodes. Moreover, the requested content is cached with probability at all routers that have empty cache along the delivery path. If this approach has 1 probability then it will work like LCE (Chai *et al.,* 2013). If it has other than one probability then the transmitted content will store according to probability (e.g., probability based on available caching space in routers). Figure 2 shows the transmission of one content with probability (Psaras, Chai, and Pavlou, 2014). If the strategy has probability (p = 1), then the content will be stored at all routers (nodes) available on the delivery path. For example, in Figure 2, when a request from User1 for content C1 is received at the source, the source sends the content C1 towards the requester (User1) through routers R2, R3, and R4. According to Prob, if the strategy has probability (p = 1), then a copy of the requested content will be cached at all routers during the transmission of C1 from source toward User1. If another request for content C2 is received at the source from User2, the source (router R1) immediately responses by transmitting the content C2 through routers R2, R5, and R6. The router R5 does not have empty space in its cache, therefore, the content C2 has low probability at R2, and thus it will not cache.

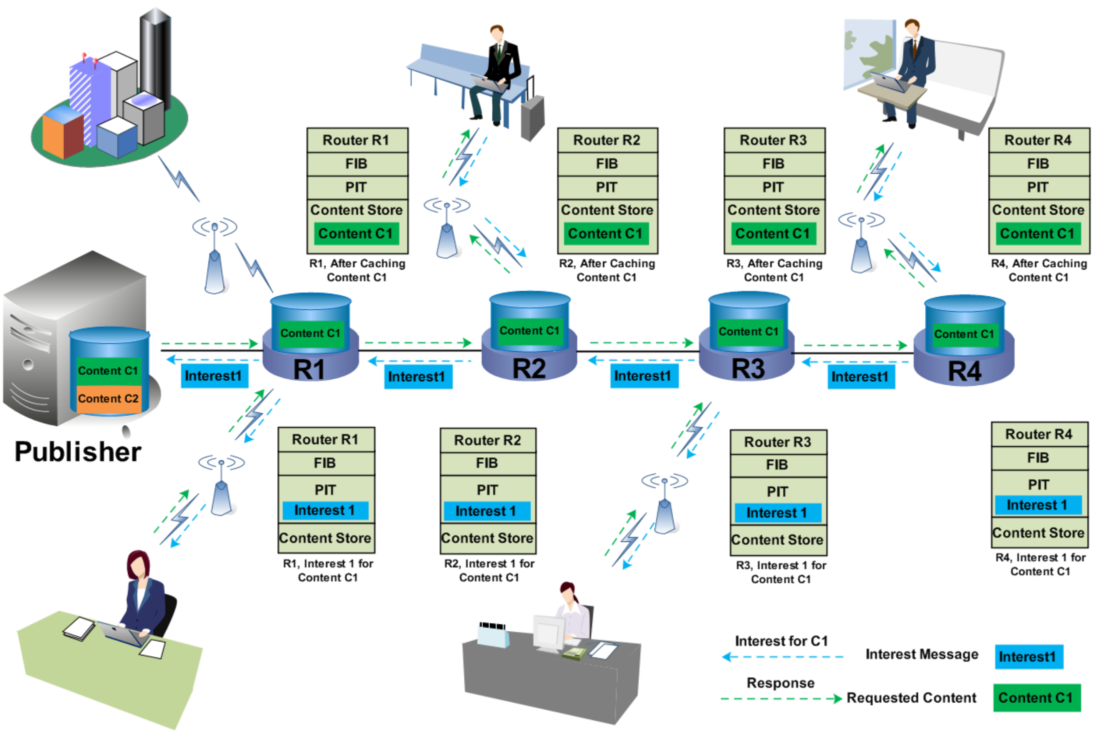


Figure 2. Probabilistic Caching.

3. PROBLEMS DESCRIPTION

NDN reduces the expected flood of global data through cache implementation. Caches are used to store the transmitted contents (Davis *et al.,* 2006). However, the reason is that the amount of transmitting data is much larger than the cache that causes several problems for example congestion, usage of the resources, bandwidth, latency, and server load. Hence, the need for a cache management approach. For this reason, several cache management strategies were proposed. A caching strategy decides when and where to place the transmitted contents (Sourlas, Gkatzikis, Flegkas, and Tassiulas, 2013). The given scenario (Figure 1) presented a portion of the network on which the NDN default caching strategy is implemented. In this scenario when User1 sends two requests to the source routers R1 and R5 for content C1 and C2 then source routers respond by sending C1 and C2 to User1. According to NDN, a copy of the requested contents will be stored at all routers along the data delivery path. However, what will happen with the requested content when the cache of the delivery path overflow. In this case, there is a need to make room for newly requested content by the eviction of stored contents and the eviction will continue until the new content reaches the subscriber. From the given scenario, it can be concluded that some problems still exist which affect the NDN performance. The default NDN caching strategy shows the highest redundancy and engages most of the cache space that results in higher eviction operations when there is a need to cache new contents while the cache is full. It decreases content diversity (Wood and Scott, 2018). When the caches overflow, then the user's requests need to traverse to the main source that increases the distance between the user and the source. Moreover, it also decreases the cache hit rate.

ProbCache provides fair allocation of resources along the delivery path among transmitted contents but it has no content distinction to update the content requests and contents reply packets (computational overhead) arbitrary definition of parameters (Psaras *et al.,* 2012). Probabilistic caching strategies have low overhead but no cooperation among the nodes that caused the high redundancy and increase the utilization of the resources. Random cache strategy presents simplicity and low overhead but it has no content and position distinction and provides unpredictable nature (Din *et al.,* 2017). ProbCache by Seetharam (2017) realized that it has obtained great popularity in NDN to manage the cache efficiently because of its flexible nature of probabilistic. Sometimes, it increases the distance between the subscriber to the publisher because it performs caching operation at the routers wherever it finds the free cache space. Initially, it works as LCE due to free cache space (by Figure 2) and when the cache is overflow. In this case, a number eviction operations need to be executed to make room for newly requested contents, hence the eviction-caching operation will increase and the cache hit ratio will decrease due to its long stretch to retrieve the desired content form the main source. Moreover, the caching at everywhere exploits the storage and increase the content redundancy by the accommodation of similar data at multiple positions (Acs *et al.,* 2017). Consequently, it projected less diversity because of homogeneous content replications while the hop reduction will be higher as each request needs to traverse to the main sources due to less empty cache along the data downloading path.

4. PROPOSED MODEL

4.1 Edge Touch Caching Strategy

Edge Touch Caching Strategy (ETCS) is presented diagrammatically in Figure 3. Based on Figure 3, all incoming contents are cached at the edge routers, but when the caches of these routers become full, the Random replacement policy is followed for content eviction. The random policy is used because its computational complexity is order 1, represented by O(1) (Nowacki, Zhao, and Palesch, 2017), therefore, the searching overhead during the content eviction process will be quite minimal. After that, the evicted contents follow betweenness centrality in each Autonomous System (AS), i.e., they are cached at a node (in each AS) that maximum nodes traverse it during content downloading. In this way, all the requests will go through the node having maximum betweenness centrality and if the contents are found there, the requests are satisfied here rather than forwarded to the server. If there need to evict any content from the betweenness centrality router the Least Replacement Policy (LRU) will be used (Tsai and Lei, 2017). In addition, if the requested contents are not found here, the requests are forwarded to the edge router and hence the edge router replies with the actual object. In this way, the following goals are achieved: the cache hit rate and content diversity are improved, the eviction rate is kept at a minimum, and in turn, the content retrieval latency is reduced and the overall bandwidth is utilized efficiently. When the cache of edge routers becomes full, the Random replacement policy is followed for content C2 eviction and it is cached at the caching routers because the caching routers have maximum connections with other routers. The hit rate will increase by caching the contents at central routers because these caching routers are connected with maximum routers. If there is a need to evict any content from a caching router then the ETCS will use the LRU policy to evict the contents. Besides, the diversity will increase because in each AS only one router caches a copy of the downloaded content, and thus the content redundancy is minimized. Moreover, the following goals are achieved: the cache hit rate and content diversity are improved, the eviction rate is kept at a minimum, and in turn, the content redundancy is reduced, and the overall caching performance is enhanced.

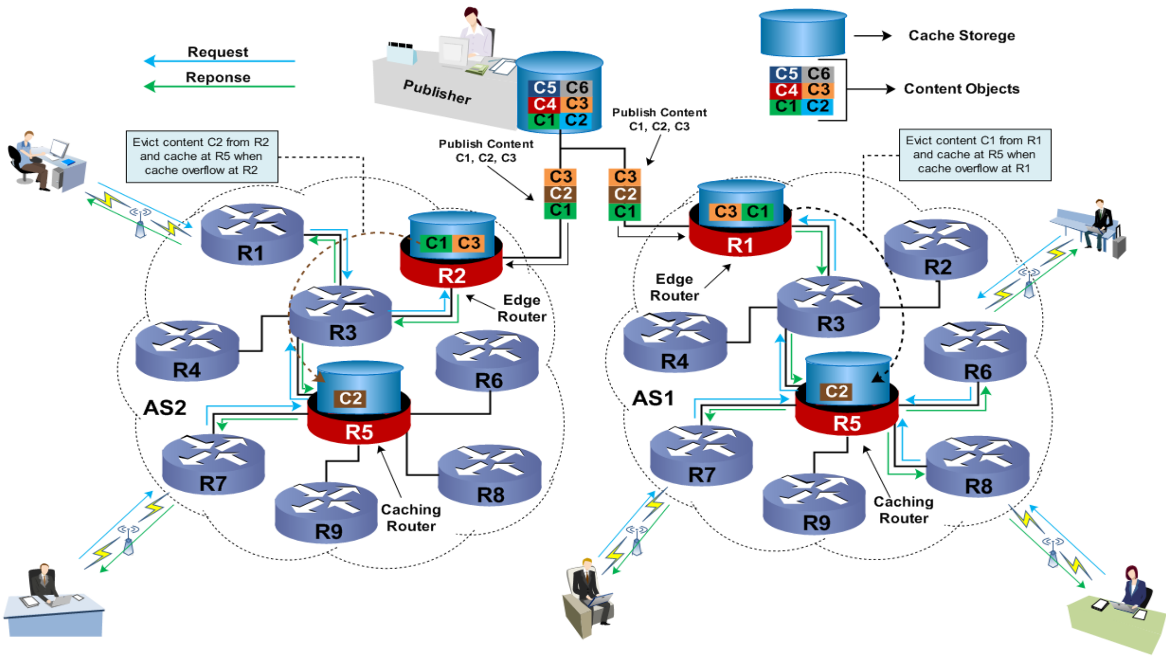


Figure 3. Edge Touch Caching Strategy (ETCS).

5. PERFORMANCE EVALUATION

For the evaluation of proposed ETCS, a simulation was conducted to evaluate the performance by using the SocialCCNSim simulator, in which each node is associated with cache storage. This research compares caching strategies that are not incorporated into the renowned simulator e.g., ccnSim. A custom-built SocialCCNSim simulator is used for the ease of implementation detail. The performance in terms of caches hit ratio, content diversity, hop decrement, and content redundancy of the proposed strategy was compared against the strategies indicated in related work (C. Bernardini). LCE (Tian, Mohri, Otsuka, Shiraishi, and Morii, 2015) and ProbCache (Psaras *et al.,* 2012) were developed in SocialCCNSim for caching procedure. These strategies will be used as a benchmark to compare with the proposed work. For content replacement operations, most of the NDN caching strategies adopted LRU. LRU was agreed as the optimal content replacement policy due to its better performance in terms of overhead and complexity. The present study used Zipf Alpha 1.2 that considered video-on-demand contents because it shows high traffic production due to user interests in media-driven content. According to a recent study by Ren et al. (2014), Zipf distribution is required to select particular traffic and content popularity. Presently, Video on Demand (VoD) is used to examine the performance of ETCS using Abilene topology.

In this study, a proposed Cache new Deployment Strategy is presented. The aim of the proposed strategy is to diminish the challenges faced by NDN cache by enhancing the content heterogeneity and cache hit. To examine the performance of ETCS cache, a social network graph such as Facebook (Sharma, Rathore, and Park, 2017) comprising of the number of users who were focused on the standard procedure for interest (request for content) and served by the source of the data was simulated. In a social network, users usually exhibit their social relationship as insist for contents and network relationship that exist between nodes. For the simulation benefit, a large number of users were 4,039 with 88,234 friends transversely the network known as edges (Leskovec and Mcauley, 2012). For the simulation, several parameters cache sizes (100, 1000) and the number of runs, were selected carefully. The topology and catalogue sizes are depicted in

Table 1. According to the huge demands for media data, the Zipf popularity 1.2 is selected given by (Fricker, Robert, Roberts, & Sbihi, 2012) categorization.

Table 1 Simulation Description Values

Parameter	Value/Description
Simulation time	One day
Chunk Size	10MB
Cache Size	100, 1000
Catalogue Size	10^6
Alpha value	1.2
Topology Replacement Policy	Abilene LRU
Simulator	SocialCCNSim
Traffic Source	SONETOR (Facebook (Cantador, Brusilovsky, and Kuflik, 2011))

5.1 Performance Metrics

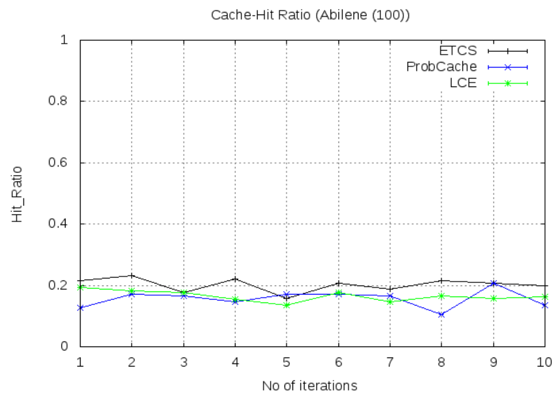
Metrics are used to test the effectiveness of cache management strategies. Adequate caching will increase the performance in terms of content diversity and hit ratio.

5.2 Cache Hit Ratio

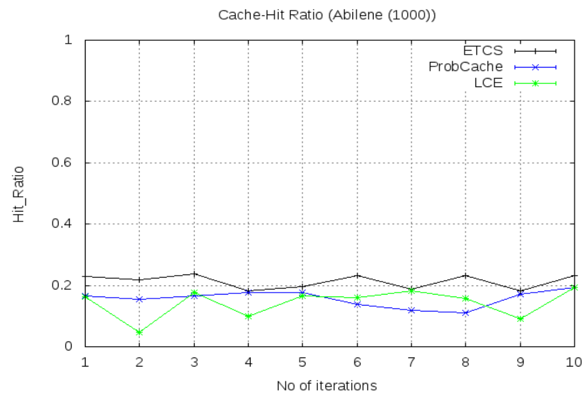
The number of requests is served by the cache used in network routers. Cache delivers substantial advantages once the requested data is served from the adjacent cacheable router (Tourani, Misra, Mick, and Panwar, 2017). Cache hit ratio depends on the effectiveness of the caching system and it is influenced by some factors such as the caching strategy, the number of cacheable contents, cache size and the time required by a content consume in router's cache. An effectual caching strategy reinforces the number of caches hits ratio by the availability of desired contents near the requesters. The result of the cache hit ratio is graphically demonstrated in the graph in Figure 4(a). In this graph, the proposed strategy ETCS performs better in terms of cache hit than the former strategies LCE and ProbCache because of its heterogeneous nature to cache the content along the delivery path. At the second iteration, ETCS achieves the maximum cache hit that shows the global maxima of cache hit gain less hit ratio at fifth iteration because sometimes cache overflow and it takes time to replace the old content to accommodate the new. In the same graph, ProbCache relatively performs fewer than ETCS due to its flexible nature to cache content at all the free cache space. Moreover, only at the ninth iteration, it performs like the proposed strategy but overall, it has a low cache hit ratio. In addition, LCE relatively has less cache hit ratio due to its numerous caching-eviction operations which increases the publisher-subscriber's distance. On the other hand, when the cache size was expanded from 100 to 1000, based on the graphical results in Figure 4, ETCS still performs better than ProbCache and LCE. The overall performance with 100 cache size is illustrated in Figure 5(a) and 5(b). In this graph, ETCS has 6% and 9% better consequence than ProbCache and LCE, respectively. However, when the cache size increased to 1000, it performs about 6% and 8% better than ProbCache and LCE as demonstrated in Figure 5.

5.3 Diversity

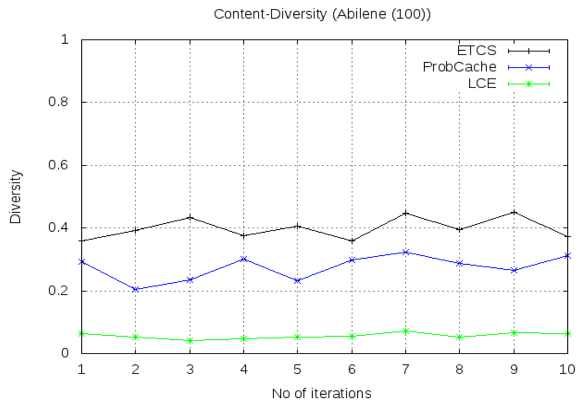
The number of unique contents cached in the given network alongside the data downloading path (Abdullahi & Arif, 2017). The simulation consequences are graphically represented in (c). In which the diversity has different results on comprised strategies and it seems that ETCS projected better performance in terms of content diversity because ETCS increase the amount of miscellaneous content to be cache alongside the delivery path. LCE shows a low diversity ratio as compare to the ETCS and ProbCache due to the design of its cache everything everywhere algorithm. Therefore, the multiple replications of similar contents decrease the cache space to store the diverse content at the selected nodes. ProbCache relatively performs better than LCE but it has low diversity than proposed strategy ETCS. It shows maximum diversity at the seventh iteration but still has less value than ETCS. The overall performance is illustrated individually in Figure 5, through the bar chart. It can be observed that ETCS recorded 12% and 24% better diversity than ProbCache and LCE, respectively. As the cache size expanded from 100 to 1000, ETCS still performs 13% and 28% better than the benchmark strategies as demonstrated in Figure 5.



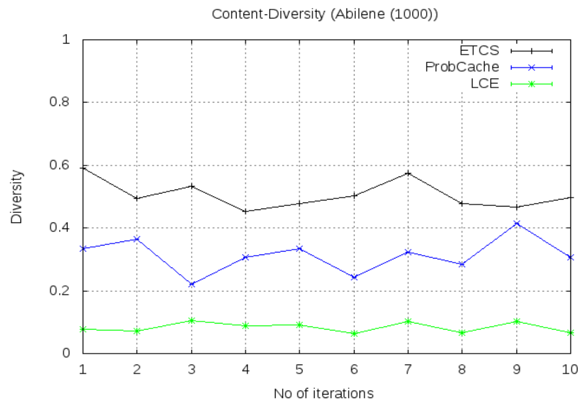
(a) Cache Hit Ratio (Abilene (100 cache size)



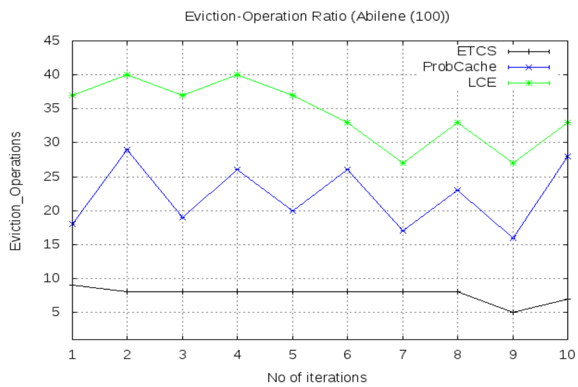
(b) Cache Hit Ratio (Abilene (1000 cache size)



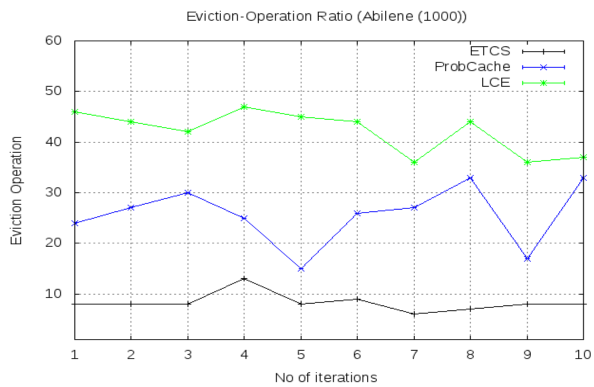
(c) Content Diversity (Abilene (100 cache size)



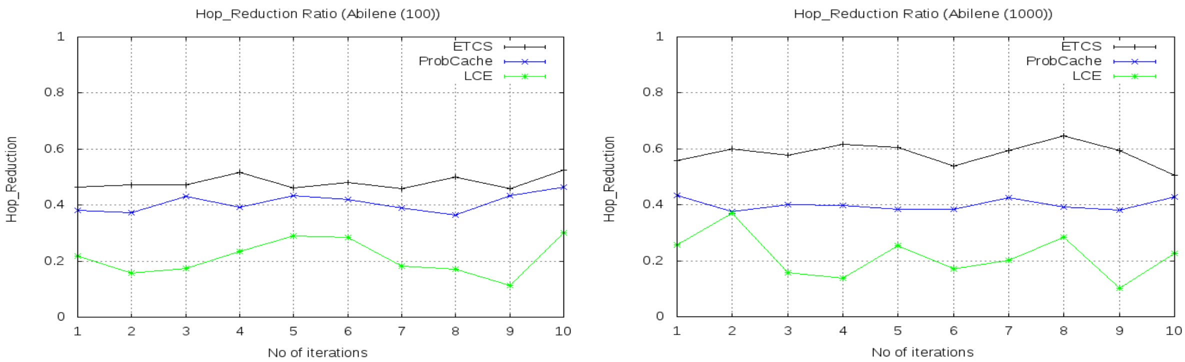
(d) Content Diversity (Abilene (1000 cache size)



(e) Eviction Operations (Abilene (100 cache size)



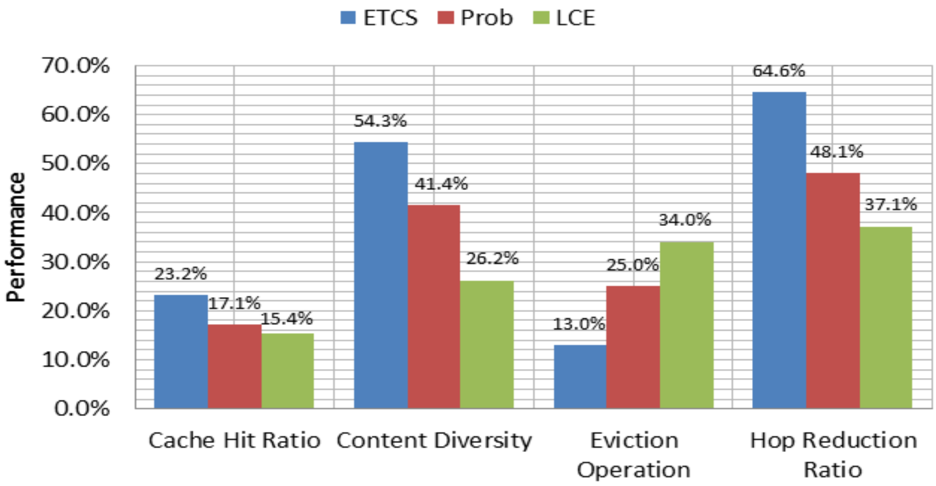
(f) Eviction Operations (Abilene (1000 cache size)



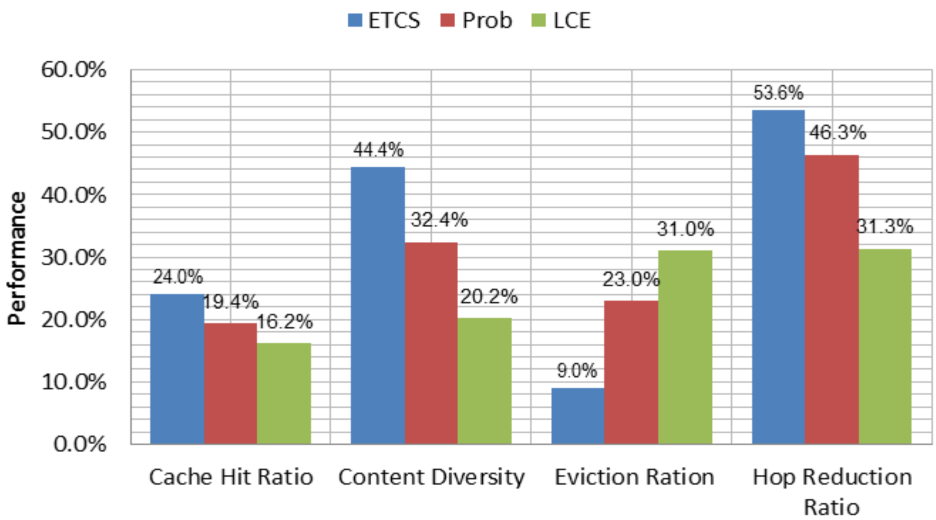
(g) Hop Reduction Ratio (Abilene (100 cache size)

(f) Hop Reduction Ratio (Abilene (100 cache size)

Figure 4. Simulation Results on Caching Strategies.



(a) Performance with cache size =100



(b) Performance with cache size =1000

Figure 5. Overall Performance of Proposed and benchmark caching strategies.

We can conclude that ETCS supports the content heterogeneity but LCE boosted up the homogeneous content replication however, ProbCache depends on the free cache location. It can be said that it somehow promotes the number of diverse contents.

5.4 Eviction Operation

It is a useful matrix to calculate the NDN caching performance. In NDN, when the cache of the routers becomes full and there is a need to make room for new coming content, then the old contents have to be evicted to accommodate the new contents (Arshad, Azam, Rehmani, and Loo, 2017). The consequence presented the correlation among the tested ETCS, ProbCache and LCE, in terms of eviction operations using the cache size 100 and 1000 in Figure 4. The result presents overall content eviction operations where the LCE was observed to have a higher rate of evictions because of its design of caching at all routers traversed. ETCS shows better results which means that it has low traffic congestion than LCE. However, ProbCache performs better as compared to LCE but it still has higher evictions than ETCS due to high congestion of similar content's replication at multiple positions. Consequently, the result showed that ETCS has reduced eviction operations by14% and 22% than ProbCache and LCE, respectively. When the cache size was increased to 1000, a high effect is displayed with a large eviction ratio due to the availability of more cache space for transmitted contents to be cached. Nevertheless, ETCS recorded a better score in terms of eviction ratio than ProCache and LCE as illustrated in Figure 5.

5.5 Hop-Reduction Ratio

It can be defined that the number of hops is traversed by a user request to retrieve its obligatory data content from the source (Abani *et al.,* 2017). If the hop reduction ratio is large, it means that the requested content is cached near the requester, therefore, it will take less time to download the desired content. The consequences of hop reduction using Abilene topology are graphically demonstrated in Figure 4. It is clear that the ETCS can cache transmitted content close to the subscriber because its nature is to store the content at the betweenness the central position within the autonomous system which reduces the stretch from subscriber to the publisher. Therefore, ETCS is effective than the benchmark strategies in terms of hop reduction by providing a short distance for the content retrieval process. The overall performance is illustrated in Figure 4 where ETCS achieved 10.3% and 22.3% improved outcomes than ProbCache and LCE. LCE has a smaller amount of hop reduction because, in this strategy, the new request needs to travel to the main source when the cache is overflow along the data downloading path. ProbCache performs relatively better than LCE, but it still has a low hop reduction ratio due to its caching mechanism to store content at the free cache. In addition, when the cache size is increased (become 1000) then the outcome is better as compared to small cache size illustrated in Figure 4. Consequently, ETCS performs 16.5% and 27.5% better than ProbCache and LCE as demonstrated in Figure 5.

6. CONCLUSION

In this paper, a NDN content caching strategy was developed to diminish the multiple replications of content alongside the data routing path. Due to the limited size of the cache, this proposed caching strategy increases the content diversity through caches the heterogeneous content in the network. We proposed cache deployment and content distributed caching strategy alongside the data routing path, called ETCS (Progressive Content Placement Caching). The goal of ETCSahe is to increase the content diversity by reducing the redundant content and it improves the overall cache hit ratio. The proposed strategy is evaluated through a simulation comparison against some state-of-art caching strategies in a simulated environment. The evaluated performance shows that our strategy achieves better results as well while the cache size is restricted.

REFERENCES

[1] Abani, N., Braun, T., & Gerla, M. Proactive caching with mobility prediction under uncertainty in information-centric networks. Paper presented at the Proceedings of the 4th ACM Conference on Information-Centric Networking, (2017).

[2] Abdullahi, I., & Arif, S. Content Diversity in Information-Centric Network Caching. Advanced Science Letters **23**, 6 (2017) 5361-5364.

[3] Acs, G., Conti, M., Gasti, P., Ghali, C., Tsudik, G., & Wood, C. Privacy-Aware Caching in Information-Centric Networking. IEEE Transactions on Dependable and Secure Computing, (2017).

[4] Ahlgren, B., Dannewitz, C., Imbrenda, C., Kutscher, D., & Ohlman, B. A survey of information-centric networking. IEEE Communications Magazine **50**, 7 (2012).

[5] Amadeo, M., Campolo, C., Molinaro, A., & Ruggeri, G. Content-centric wireless networking: A survey. Computer Networks **72** (2014) 1-13.

[6] Arshad, S., Azam, M. A., Rehmani, M. H., & Loo, J. Information-Centric Networking based Caching and Naming Schemes for Internet of Things: A Survey and Future Research Directions. arXiv preprint arXiv:1710.03473. (2017).

[7] Bernardini, C. SocialCCNSim. [Online]. Available: https://github.com/mesarpe/socialccnsim.

[8] Bernardini, C., Silverston, T., & Festor, O. A comparison of caching strategies for content centric networking. Paper presented at the Global Communications Conference (GLOBECOM), 2015 IEEE, (2015).

[9] Cantador, I., Brusilovsky, P. L., & Kuflik, T. Second workshop on information heterogeneity and fusion in recommender systems (HetRec2011): ACM, (2011).

[10] Carofiglio, G., Gallo, M., & Muscariello, L. Optimal multipath congestion control and request forwarding in information-centric networks: Protocol design and experimentation. Computer Networks **110** (2016) 104-117.

[11] Carofiglio, G., Morabito, G., Muscariello, L., Solis, I., & Varvello, M. From content delivery today to information centric networking. Computer Networks **57**, 16 (2013) 3116-3127.

[12] Chai, W. K., He, D., Psaras, I., & Pavlou, G. Cache "less for more" in information-centric networks (extended version). Computer Communications **36**, 7 (2013) 758-770.

[13] Davis, A. T., Parikh, J., Pichai, S., Ruvinsky, E., Stodolsky, D., Tsimelzon, M., & Weihl, W. E. Java application framework for use in a content delivery network (CDN): Google Patents, (2006).

[14] Din, I. U., Hassan, S., Khan, M. K., Guizani, M., Ghazali, O., & Habbal, A. Caching in Information-Centric Networking: Strategies, Challenges, and Future Research Directions. IEEE Communications Surveys & Tutorials, (2017).

[15] Eum, S., Nakauchi, K., Murata, M., Shoji, Y., & Nishinaga, N. CATT: potential based routing with content caching for ICN. Paper presented at the Proceedings of the second edition of the ICN workshop on Information-centric networking, (2012).

[16] Fang, C., Yu, F. R., Huang, T., Liu, J., & Liu, Y. An energy-efficient distributed in-network caching scheme for green content-centric networks. Computer Networks **78** (2015) 119-129.

[17] Fricker, C., Robert, P., Roberts, J., & Sbihi, N. Impact of traffic mix on caching performance in a content-centric network. Paper presented at the Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on. (2012).

[18] Katsaros, K., Xylomenos, G., & Polyzos, G. C. MultiCache: An overlay architecture for information-centric networking. Computer Networks **55**, 4 (2011) 936-947.

[19] Leskovec, J., & Mcauley, J. J. Learning to discover social circles in ego networks. Paper presented at the Advances in neural information processing systems, (2012).

[20] Nowacki, A. S., Zhao, W., & Palesch, Y. Y. A surrogate-primary replacement algorithm for response-adaptive randomization in stroke clinical trials. Statistical methods in medical research **26**, 3 (2017). 1078-1092.

[21] Psaras, I., Chai, W. K., & Pavlou, G. Probabilistic in-network caching for information-centric networks. Paper presented at the Proceedings of the second edition of the ICN workshop on Information-centric networking, (2012).
[22] Psaras, I., Chai, W. K., & Pavlou, G. In-network cache management and resource allocation for information-centric networks. IEEE Transactions on Parallel and Distributed Systems **25**, 11 (2014) 2920-2931.
[23] Ren, J., Qi, W., Westphal, C., Wang, J., Lu, K., Liu, S., & Wang, S. Magic: A distributed max-gain in-network caching strategy in information-centric networks. Paper presented at the Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on. (2014).
[24] Salsano, S., Blefari-Melazzi, N., Detti, A., Morabito, G., & Veltri, L. Information centric networking over SDN and OpenFlow: Architectural aspects and experiments on the OFELIA testbed. Computer Networks **57**, 16 (2013) 3207-3221.
[25] Seetharam, A. On Caching and Routing in Information-Centric Networks. IEEE Communications Magazine, (2017).
[26] Sharma, P. K., Rathore, S., & Park, J. H. Multilevel learning based modeling for link prediction and users' consumption preference in Online Social Networks. Future Generation Computer Systems, (2017).
[27] Sourlas, V., Gkatzikis, L., Flegkas, P., & Tassiulas, L.. Distributed cache management in information-centric networks. IEEE Transactions on Network and Service Management **10**, 3 (2013) 286-299.
[28] Tian, H., Mohri, M., Otsuka, Y., Shiraishi, Y., & Morii, M. Lce in-network caching on vehicular networks for content distribution in urban environments. Paper presented at the Ubiquitous and Future Networks (ICUFN), 2015 Seventh International Conference on. (2015).
[29] Tourani, R., Misra, S., Mick, T., & Panwar, G. Security, privacy, and access control in information-centric networking: A survey. IEEE Communications Surveys & Tutorials, (2017).
[30] Tsai, H.-B., & Lei, C.-L. A page replacement algorithm based on frequency derived from reference history. Paper presented at the Proceedings of the Symposium on Applied Computing, (2017).
[31] Wählisch, M., Schmidt, T. C., & Vahlenkamp, M. Backscatter from the data plane–threats to stability and security in information-centric network infrastructure. Computer Networks **57**, 16 (2013) 3192-3206.
[32] Wang, W., Sun, Y., Guo, Y., Kaafar, D., Jin, J., Li, J., & Li, Z. CRCache: Exploiting the correlation between content popularity and network topology information for ICN caching. Paper presented at the Communications (ICC), 2014 IEEE International Conference on. (2014).
[33] Wood, C. A., & Scott, G. C. Adjusting entries in a forwarding information base in a content centric network: Google Patents, (2018).
[34] Xylomenos, G., Ververidis, C. N., Siris, V. A., Fotiou, N., Tsilopoulos, C., Vasilakos, X., . . . Polyzos, G. C. A survey of information-centric networking research. IEEE Communications Surveys & Tutorials **16**, 2 (2014). 1024-1049.
[35] Zhang, G., Li, Y., & Lin, T. Caching in information centric networking: A survey. Computer Networks **57**, 16 (2013) 3128-3141.