Journal of Engineering Research and Education

In Press, Accepted Manuscript – Note to users

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

Muhammad Ali Naeem[1], Shahrudin Awang Nor[1], Suhaidi Hassan[1]
[1]InterNetWorks Research Laboratory School of Computing Universiti Utara Malaysia
(malinaeem7@gmail.com, shah@uum.edu.my, suhaidi@uum.edu.my)

ABSTRACT. Named Data networks (NDNs) agree to cache contents inside the network, 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 keep away from energy-limit in data delivery environments, *e.g.*, wireless networks. In this paper, we propose a content disseminated caching strategy alongside the data routing path, called Edge Touch Caching Strategy. The main goal of ETCS is to reduce the caching of homogeneous contents at the numerous locations during data dissemination along the data delivery path to increase the content diversity. We also capture the effect of cache hit ratio through compare 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 perform 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 in the Internet is about 15 minutes. These 15 minutes amount to 16 Gigabytes per month per customer. This accounts for 66% of all Internet traffic (Zhang, Li, & Lin, 2013). Game consoles, Smartphones, tablets, and Smart TV are all used to streaming 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 between up 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 Internet is depicted in Figure 1.4 in a more detailed form. Moreover, file sizes have grown 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 video daily consumption will become hours of consumption. The 16GB of American video traffic per month will become 600GB (Carofiglio, Gallo, & 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 which 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 the Internet traffic (Abani, Braun, & Gerla, 2017). Congestion is getting out of control and new mechanisms are needed to fulfill the Internet quality requirements. In this context, temporary storage servers have been used all 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, & Veltri, 2013). With NDN, caches will be available at every node and the network will become a network of caches (Katsaros, Xylomenos, & Polyzos, 2011). Caches have never been deployed in such 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, i.e., it increases the available space for storing contents but the complexity to manage them increases at the same time. To manage the 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, & Pavlou, 2012), Cache Less For More (Chai, He, Psaras, & Pavlou, 2013), and some others, for example off-path caching strategies (Amadeo, Campolo, Molinaro, & Ruggeri, 2014) and on-path caching strategies (Ahlgren, Dannewitz, Imbrenda, Kutscher, & Ohlman, 2012; Wählisch, Schmidt, & Vahlenkamp, 2013). However, it is unclear which caching strategy fits best for every possible scenario.

2. RELATED STUDY

2.1 Leave Copy Everywhere (LCE)

Note: Accepted manuscripts are articles that have been peer-reviewed and accepted for publication by the Editorial Board. These articles have not yet been copyedited and/or formatted in the journal house style.

Caching is a most central module of all NDN architectures. It plays a great role to implement the main idea of NDN (César Bernardini, Silverston, & 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 requested data for a specific time span. Any node (router) holding a copy of requested object can satisfy the user primary and subsequent request for the same content (Carofiglio, Morabito, Muscariello, Solis, & 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 follow the back path and a copy of required data is cached along the data delivery path (Wang et al., 2014). It generally uses 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 requested data at each router along the delivery path from the source to the user (Eum, Nakauchi, Murata, Shoji, & 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 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 the resource usability and reduces the cache hit rate.

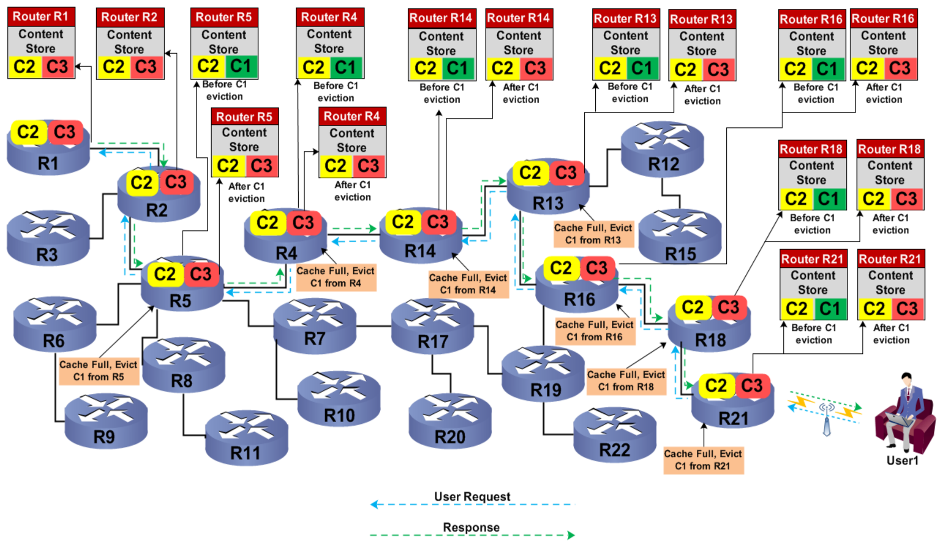
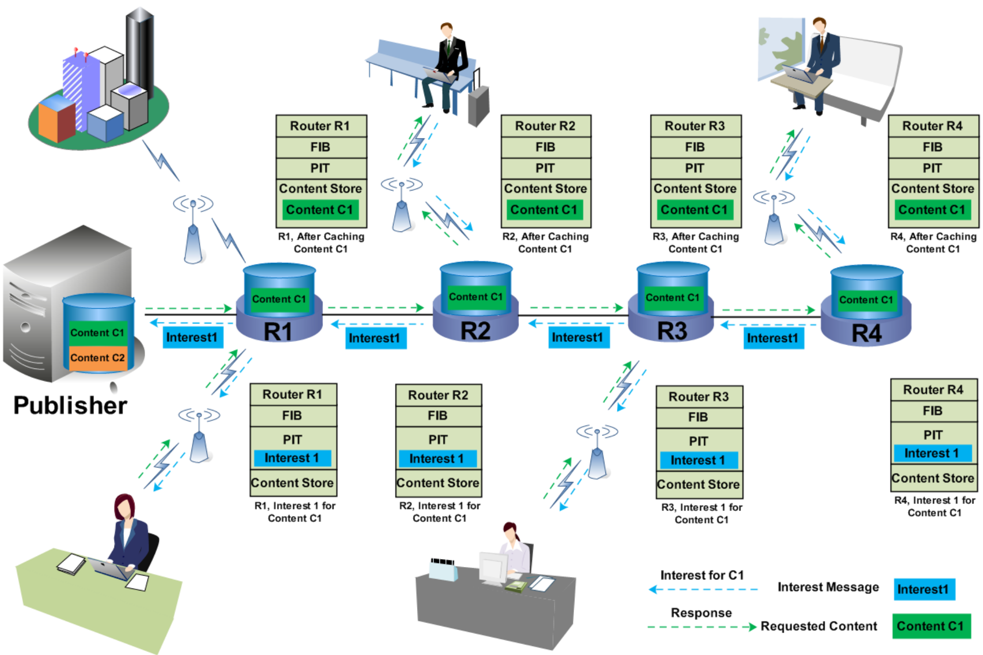


Figure 1. Leave Copy Everywhere Caching Strategy



Note: Accepted manuscripts are articles that have been peer-reviewed and accepted for publication by the Editorial Board. These articles have not yet been copyedited and/or formatted in the journal house style.

Figure 2. Probabilistic Caching

2.2. Probabilistic Caching

Probabilistic caching (Psaras et al., 2012) is a renowned content deployment strategy for that the research community takes more interest because of its flexible probabilistic value (Fang, Yu, Huang, Liu, & Liu, 2015). The probabilistic value is defined by the algorithm and it has the pre-defined 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, & Pavlou, 2014). If the strategy has probability (p = 1) then the content will store at all routers (nodes) that are available on the delivery path, for example, in the given figure, 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 the User. 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.

3. PROBLEMS DESCRIPTION

NDN reduces the expected flood of the global data through cache implementation. Cache is used to store the transmitted contents (Davis et al., 2006). But the reason is that the amount of transmitting data is much larger than the cache. That caused several problems for example congestion, usage of the resources, bandwidth, latency, server load. So it demands for a cache management approach. For this reason a number of cache management strategy were proposed. A caching strategy decides when and where to place the transmitted contents (Sourlas, Gkatzikis, Flegkas, & Tassiulas, 2013). The given scenario (Figure 1) is presented a portion of the network on which 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 sources routers respond by sending C1 and C2 to the user1 according to NDN a copy of requested contents will store at all the routers along the data delivery path. But what will happen with the requested content when the cache of the delivery path will 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 reach to the subscriber. We can conclude by the given scenario, some problems still exist which affect the NDN performance. The default NDN caching strategy shows the highest redundancy and engages most of cache space that results higher eviction operations when there is a need to cache new contents while the cache is full. It decreases the content diversity (Wood & Scott, 2018). When the caches overflow then the users requests need to traverse to the main source that increase the distance between the user and the source. It also decrease the cache hit rate.

ProbCache provides fair allocation of resources along the delivery path among transmitted contents but it has no any 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 any cooperation among the nodes that caused the high redundancy and increase the utilization of the resources. Random cache strategy presents the simplicity and low overhead but it has no content and position distinction and provides unpredictable nature (Din et al., 2017). ProbCache by (Seetharam, 2017) is realized that it has obtained great popularity in NDN to manage the cache in efficient manner because of its flexible nature of probabilistic. Sometime it increase the distance between the subscriber to the publisher because it performs caching operation at the routers wherever it find 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 needs to be execute to make room for newly requested contents hence the eviction-caching operation will increases 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 the each request needs to traverse to the main sources due to less empty cache along the data downloading path.

4. PROPOSED MODEL

EdgeTouch Caching Strategy

ETCS is presented diagrammatically in Figure 3. In the given figure, 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. Random policy is used because its computational complexity is order 1, represented by O(1) (Nowacki, Zhao, & Palesch, 2017), therefore, the searching overhead during content eviction process will be quite minimum. 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 used (Tsai & 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. Through this way the following goals are achieved: the cache hit rate and content diversity are improved, the eviction rate is kept at minimum, and in turn the content retrieval latency is reduced and the overall bandwidth is utilized in an efficient manner. 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. In addition, 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 minimum, and in turn the content redundancy is reduced and the overall caching performance is enhanced.

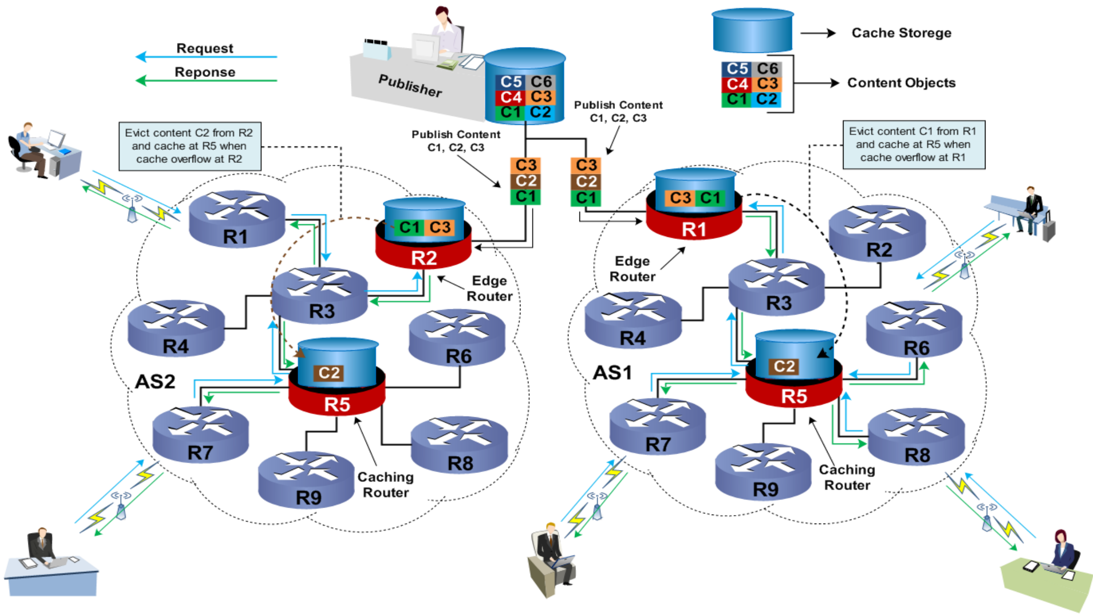


Figure 3. Edge Touch Caching Strategy (ETCS)

5. PERFORMANCE EVALUATION

For the evaluation of proposed ETCS, we conducted a simulation platform to evaluate the performance by using the socialccnsim simulator, in which each node is associated with cache storage. In this paper we are comparing caching strategies which are not incorporated into the renowned simulator e.g., ccnSim, we are using a custom built socialccnsim simulator for the ease of implementation detail. The performance in term of cache hit ratio, content diversity, hop decrement and content redundancy of the proposed strategy will compare against the strategies indicated in related work (C. Bernardini). LCE (Tian, Mohri, Otsuka, Shiraishi, & Morii, 2015) and ProbCache (Psaras et al., 2012) were developed in SocialCCNSim for caching procedure. These strategies will use as benchmark to compare with the proposed work. For content replacement operations, most of the NDN caching strategies was adopted LRU. LRU agreed as the optimal content replacement policy due to its better

Note: Accepted manuscripts are articles that have been peer-reviewed and accepted for publication by the Editorial Board. These articles have not yet been copyedited and/or formatted in the journal house style.

performance in term of overhead and complexity. The present study used Zipf Alpha as 1.2 that that considered for video on demand contents because it shows high traffic production due to user interests in media driven contents. According to the recent studies (Ren et al., 2014), Zipf distribution is required to select a particular traffic and content popularity. In present case 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 proposed strategy is to diminish the challenges faced by NDN cache through enhance the content heterogeneity and cache hit. To examine the performance of ETCS cache, a Social network graph as Facebook (Sharma, Rathore, & Park, 2017) included of a number of users was focused to the standard procedure for interest (request for content) and serving by source of the data was simulated. In 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 & Mcauley, 2012). For the simulation, several parameters cache sizes (100, 1000) and the number of runs, were selected carefully. The topology and Catalog Size depicted on 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
Catalog Size	10^6
Alpha value	1.2
Topology	Abilene
Replacement Policy	LRU
Simulator	SocialCCNSim
Traffic Source	SONETOR (Facebook (Cantador, Brusilovsky, & Kuflik, 2011))

Performance Metrics

Metrics are used to test the effectiveness of cache management strategies. Adequate caching will effect to perform better in term of content diversity and hit ratio.

5.1 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 adjacent cacheable router (Tourani, Misra, Mick, & 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 amount 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 cache hit ratio by the availability of desired contents near the requesters. The result on cache hit ratio is graphically demonstrated in graph (a). In this graph we can see clearly 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 second iteration ETCS achieves the maximum cache hit that shows the global maxima of cache hit it 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 ninth iteration it performs like the proposed strategy but overall it has low cache hit ratio. In addition, LCE relatively have less cache hit ratio due to its numerous caching-eviction operations which increases the publisher-subscriber's distance. On the other hand, when we expand the cache size from 100 to 1000 the graphical results in Figure 4, cache hit ratio graph, ETCS still performs better than ProbCache and LCE. The overall performance with 100 cache size is illustrated in Figure 5 (e.g, (a) and (b)). In this graph we can conclude ETCS have 6% and 9% better consequence than ProbCache and LCE respectively. However, when we increase However, when we increase the cache size as 1000 it performs about 6% and 8% better than ProbCache and LCE as demonstrated in Figure 5.

5.2 Diversity

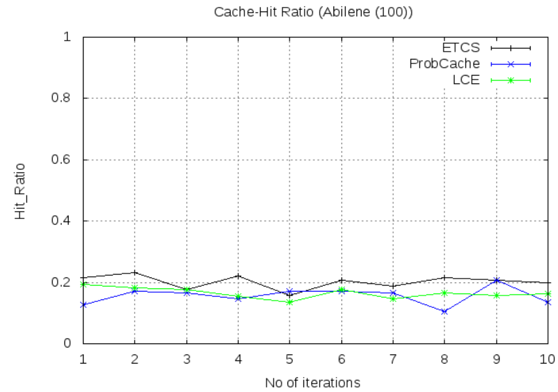
The number of unique contents cached in the given network alongside the data downloading path (Abdullahi & Arif, 2017). The simulation consequences are graphically represented in (c). In which the diversity

Note: Accepted manuscripts are articles that have been peer-reviewed and accepted for publication by the Editorial Board. These articles have not yet been copyedited and/or formatted in the journal house style.

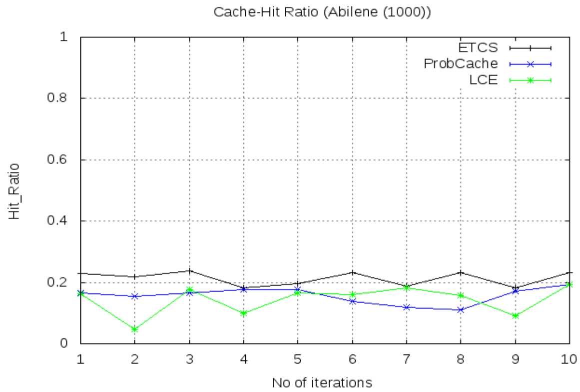
Journal of Engineering Research and Education

In Press, Accepted Manuscript – Note to users

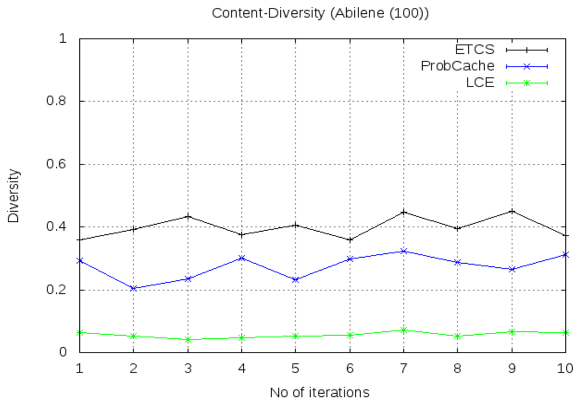
have different results on comprised strategies and it seem that ETCS projected better performance in terms of content diversity because ETCS increase the amount miscellaneous content to be cache alongside the deliver path. LCE shows low diversity ratio as compare to the ETCS and ProbCache due to design 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 seventh iteration but still have less value than ETCS. The overall performance is illustrated individually in Figure 5, through column graph. It can observe that ETCS recorded 12% and 24% better diversity than ProbCache and LCE respectively. As the cache size expand from 100 to 1000 however ETCS still performs 13% and 28% better than the benchmark strategies as demonstrated in Figure 5.



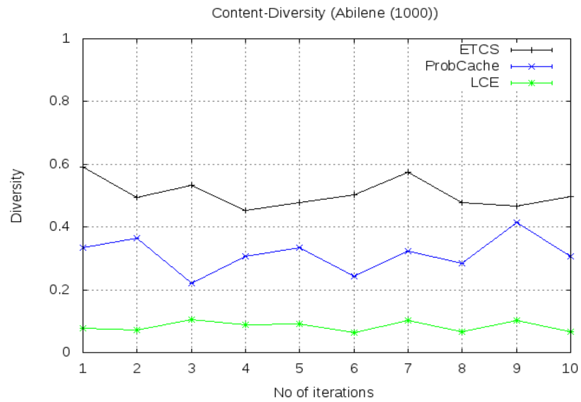
(a) Cache Hit Ratio (Abilene (100 cache size)



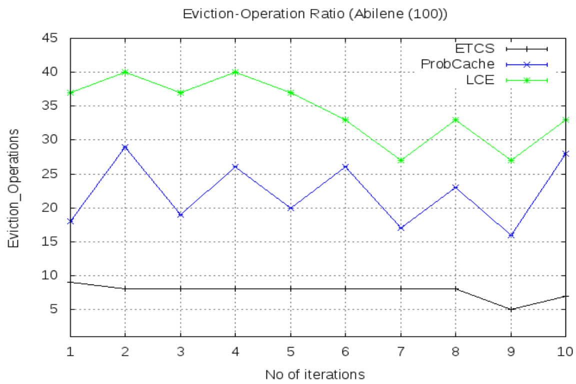
(b) Cache Hit Ratio (Abilene (1000 cache size)



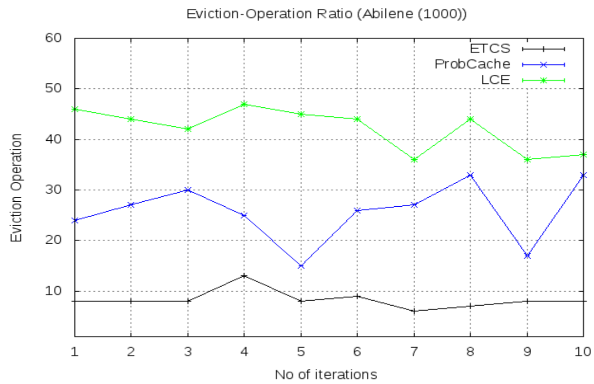
(c) Content Diversity (Abilene (100 cache size)



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



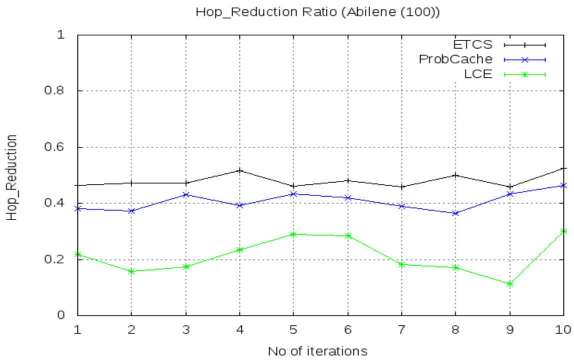
(e) Eviction Operations (Abilene (100 cache size)



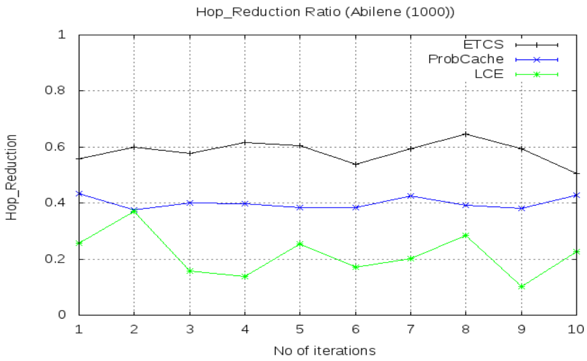
(f) Eviction Operations (Abilene (1000 cache size)

Note: Accepted manuscripts are articles that have been peer-reviewed and accepted for publication by the Editorial Board. These articles have not yet been copyedited and/or formatted in the journal house style.

Journal of Engineering Research and Education
In Press, Accepted Manuscript – Note to users

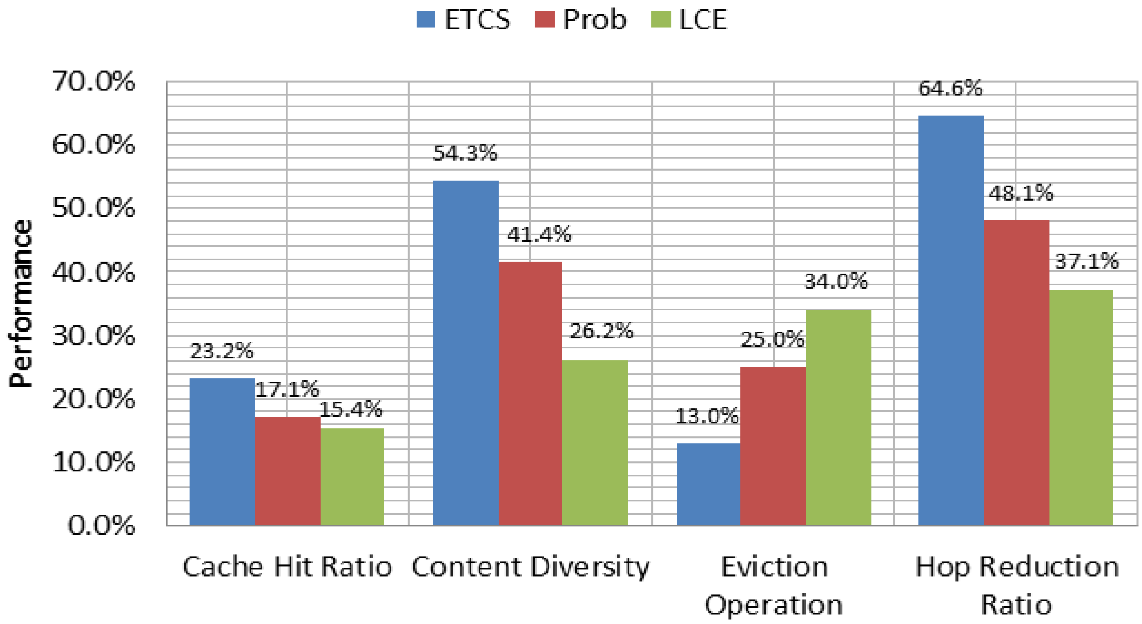


(g) Hop Reduction Ratio (Abilene (100 cache size)



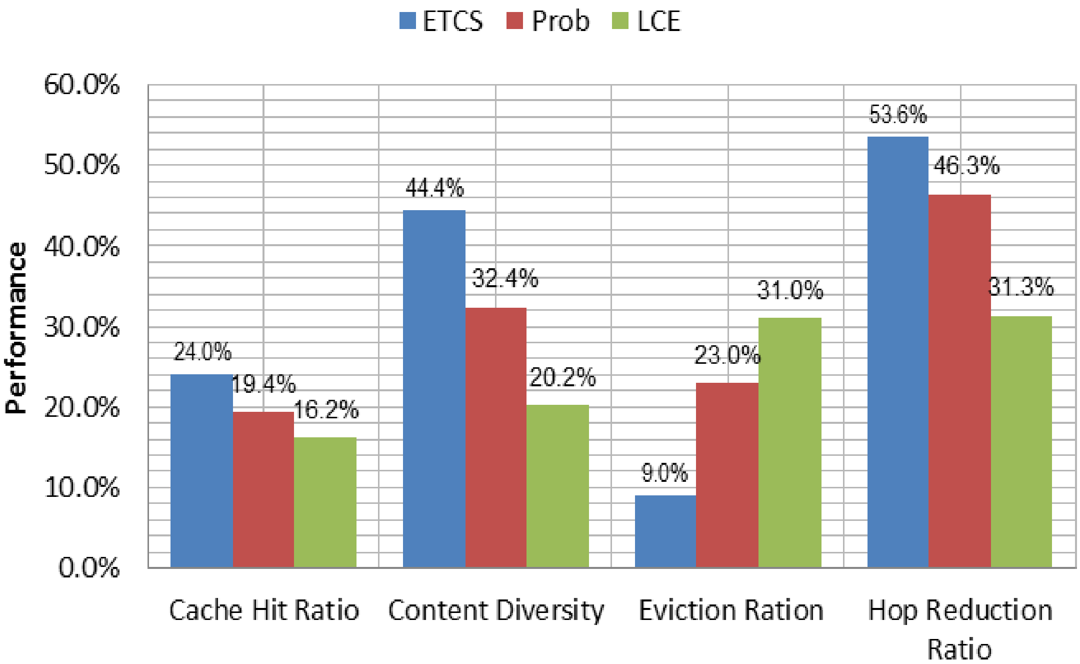
(f) Hop Reduction Ratio (Abilene (100 cache size)

Figure 4. Simulation Results on Caching Strategies



(a) Performance with cache size =100

Note: Accepted manuscripts are articles that have been peer-reviewed and accepted for publication by the Editorial Board. These articles have not yet been copyedited and/or formatted in the journal house style.



(b) Performance with cache size =1000

Figure 5. Overall Performance of Proposed and benchmark caching strategies

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

5.3 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 evicting to accommodate the new contents (Arshad, Azam, Rehmani, & Loo, 2017). The consequence is 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 the overall content eviction operations the LCE was observed to have 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 compare to LCE but it still have higher evictions than ETCS due to high congestion of similar content's replication at multiple positions. Consequently, the result Showed that the ETCS has reduce eviction operations about 14% and 22% than ProbCache and LCE. When we increase the cache size as 1000, a high effect is displayed with large eviction ratio due to the availability of more cache space for transmitted contents to be cached. Nevertheless, the ETCS recorded better score in terms of eviction ratio than ProCache and LCE as illustrated in Figure 5.

5.4 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 on hop reduction using Abilene topology is graphically demonstrated in Figure 4. It is clear that the ETCS have the ability to 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 reduce the stretch from subscriber to the publisher. Therefore ETCS is effective than the benchmark strategies in terms of hop reduction by providing the short distance for content retrieval process. The overall performance is illustrated in Figure 4. By the given figure we can see that ETCS achieve 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 have low hop reduction ratio due to its caching mechanism to store a content at the free cache. In addition, when the cache size is increased (become 1000) then the outcome is better as compare to small cache size which is 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, we developed an NDN content caching strategy to diminish the multiple replications of content alongside the data routing path. Due to the limited size of cache this proposed caching strategy increases the content diversity through caches the heterogeneous content in 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. We evaluate our strategy by make a simulation comparison against some state-of-art caching strategies in simulation environment. The evaluated performance shows that our strategy achieves better results as well while the cache size is restricted.

REFERENCES

Abani, N., Braun, T., & Gerla, M. (2017). *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.

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

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

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

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

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

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

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

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

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

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

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

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

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

Eum, S., Nakauchi, K., Murata, M., Shoji, Y., & Nishinaga, N. (2012). *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.

Note: Accepted manuscripts are articles that have been peer-reviewed and accepted for publication by the Editorial Board. These articles have not yet been copyedited and/or formatted in the journal house style.

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

Fricker, C., Robert, P., Roberts, J., & Sbihi, N. (2012). *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.

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

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

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

Psaras, I., Chai, W. K., & Pavlou, G. (2012). *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.

Psaras, I., Chai, W. K., & Pavlou, G. (2014). In-network cache management and resource allocation for information-centric networks. *IEEE Transactions on Parallel and Distributed Systems, 25*(11), 2920-2931.

Ren, J., Qi, W., Westphal, C., Wang, J., Lu, K., Liu, S., & Wang, S. (2014). *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.

Salsano, S., Blefari-Melazzi, N., Detti, A., Morabito, G., & Veltri, L. (2013). Information centric networking over SDN and OpenFlow: Architectural aspects and experiments on the OFELIA testbed. *Computer Networks, 57*(16), 3207-3221.

Seetharam, A. (2017). On Caching and Routing in Information-Centric Networks. *IEEE Communications Magazine*.

Sharma, P. K., Rathore, S., & Park, J. H. (2017). Multilevel learning based modeling for link prediction and users' consumption preference in Online Social Networks. *Future Generation Computer Systems*.

Sourlas, V., Gkatzikis, L., Flegkas, P., & Tassiulas, L. (2013). Distributed cache management in information-centric networks. *IEEE Transactions on Network and Service Management, 10*(3), 286-299.

Tian, H., Mohri, M., Otsuka, Y., Shiraishi, Y., & Morii, M. (2015). *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.

Tourani, R., Misra, S., Mick, T., & Panwar, G. (2017). Security, privacy, and access control in information-centric networking: A survey. *IEEE Communications Surveys & Tutorials*.

Tsai, H.-B., & Lei, C.-L. (2017). *A page replacement algorithm based on frequency derived from reference history.* Paper presented at the Proceedings of the Symposium on Applied Computing.

Wählisch, M., Schmidt, T. C., & Vahlenkamp, M. (2013). Backscatter from the data plane–threats to stability and security in information-centric network infrastructure. *Computer Networks, 57*(16), 3192-3206.

Wang, W., Sun, Y., Guo, Y., Kaafar, D., Jin, J., Li, J., & Li, Z. (2014). *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.

Wood, C. A., & Scott, G. C. (2018). Adjusting entries in a forwarding information base in a content centric network: Google Patents.

Xylomenos, G., Ververidis, C. N., Siris, V. A., Fotiou, N., Tsilopoulos, C., Vasilakos, X., . . . Polyzos, G. C. (2014). A survey of information-centric networking research. *IEEE Communications Surveys & Tutorials, 16*(2), 1024-1049.

Zhang, G., Li, Y., & Lin, T. (2013). Caching in information centric networking: A survey. *Computer Networks, 57*(16), 3128-3141.