Performance Evaluation of the Replacement Policies for Pending Interest Table

Suhaidi Hassan[1], Raaid Alubady[1,2] and Adib Habbal[1]
[1]InterNetWorks Research Laboratory, School of Computing, Universiti Utara Malaysia, Kedah, Malaysia.
[2]Computer Science Department, University of Babylon, Babylon, Iraq.
raaid.alubady@internetworks.my

Abstract*—Information (content) plays an important role in a Named Data Networking (NDN). Hence, an information model is essential in representing information appropriately to supports meaningful information spreading. As a distinction from the current network practice, the NDN shall concentrate on the content itself, rather than the location of the information itself. One important and common feature of NDN is leveraging through its built-in network caches (temporal store) to improve the communication and efficiency of content dissemination. Thus, caching is well thought-out as one of the most crucial features (especially in PIT) of the NDN. Its efficiency is due to it required feature of producing a flexible strategy in deciding what content to store and replace when the PIT overflows. Thus, PIT management in NDN continues being one of the primary concerns of high-speed forwarding. To address this issue, replacement policies, as one of the key factors for determining the effectiveness of a PIT in line with many researcher's haven to propose numerous replacement policies, i.e. LRU, Random and Persistent, which have been projected to attain the improved Interest drop rate, reduce the delay and Interest retransmission as when the PIT is full. However, to the best of our knowledge, there have not been studies that dealt with the performance and evaluation between the mentioned policies under different network topologies. Therefore, in this paper we study the performance of Interest drop rate, delay and Interest retransmission under different network topologies, i.e. Tree, Abilene and Germany when the PIT is full. The significance yearned for this study would be to provide a solid starting point in research directions of new PIT replacement policies for contemporary workload or selectively turning off of fewer used cache ways.

***Index Terms*—Named Data Networking; Pending Interest Table; Replacement Policy.**

I. INTRODUCTION

The Information-Centric Networks (ICNs) paradigm serves as a structural redesigning for Future Internet architecture, placing named data with referral to content rather than host locations (IP addresses). This is thus instantiated at the core of the network design [1]. In order to overcome some mismatch between the current use of the Internet and its original submitted design, several ICN projects and architectures such as Data-Oriented Network Architecture (DONA) [2], Publish Subscribe Internet Technology (PURSUIT)[3], Network of Information (NetInf)/Scalable and Adaptive Internet Solutions (SAIL) [4][5], CONVERGENCE [6] , Content-Centric Networking (CCN)/Name Data Networking (NDN) [7][8][9][10][11], COntent-centric inter-NETwork (CONET) [12][13], and MobilityFirst [14] have been proposed and tested by the networking research community.

NDN, also referred to as CCN, is a novel network architectural approached proposed by [9] for name-centric Internet. NDN differs mainly from the current network practice, by its concentration more on the content itself ("what"), as compared to the "where" for the current information management of the host. In NDN, contents are usually sub-divided into data chunks that are uniquely identified by special naming structures such as hierarchical naming structures, which, in turn, directly guide packet forwarding. This thus, avoids the use of IP addressing [15]. Slight modification in the semantics makes nodes interact with each other through a receiver-driven dissemination model, which shows that contents are generated (and served) only in response to the corresponding requests. Consequently, only two forms of messaging types are traversed between NDN users: Interest and Data packets. These message's processing are handled in NDN nodes via the three main special data structures with each having its unique operation:

- Content Store (CS);
- Forwarding Information Base (FIB);
- Pending Interest Table (PIT).

The PIT is related to a record and track keeper of events. It is used to keep track of Interest packets that have been previously requested and served as forwarded data toward content sources or requesters that are yet to be granted. Upon acquiring all information, a reverse path of direction is followed to serve Data packets as forwarded information to their requesters. It slightly partakes in routing and forwarding operations of the executed Interest packets [16][11]. Furthermore, NDN easily supports multicast communications [9]. Part of the advantage enjoyed in NDN is the ability to aggregate requests for the same contents at each node into one PIT entry (i.e., the one created after the forwarding of the first Interest), with the flexibility of keeping track of the relevant incoming faces. Moreover, the PIT should be large enough to store high volume of information. Thus, the PIT needs to be quick in order to mitigate bottleneck in Interest processing [17].

Replacement policy in PIT, is one of the important factors that determine the effectiveness of a cache. It has become even more important with the advent of the technological trends toward highly associative cache practices. The state-of-the-art processors, therefore, employ various cache policies, indicating that there is no common replacement that stands out as the best

[18]. On the one hand, one of the primary goals of the NDN is to manage cache contents (especially in PIT) in NDN routers to accommodate Interest efficiently. Similarly, it is also a challenging task to decide which content should be evicted from the memory of a router if a new packet arrives, and the cache of the router is full [19]. To overcome the above critical issues, our goal in this study is to explore some common PIT replacement policies in greater perspective.

To do this, there is a need to address some fundamental questions that haven't been fully answered in previous work. It is therefore, paramount to investigate the performance of different PIT in relations to replacement policies for contemporary workload, in different PIT configurations. This will address how some existing policies relate to PIT. Additionally, replacement in [20] policies have different effect on the instruction and entries in PIT. We deal with this specific problem by performing a critical evaluation of some possible PIT replacement policies, i.e. Persistent, Random and LRU. In this study, the performance analysis is achieved using a ndnSIM simulator [20][21] against PIT delay, Interest drop rate and Interest retransmission on three variant topologies, i.e., Abilene, Tree, Germany. The results from our study provide an essential starting point for new research in PIT replacement heuristics for contemporary workload or selective building of less used cache approaches.

II. Common PIT Replacement Policies

The PIT provides the full state of each forwarded Interest packet in the network in order to provide paths for traversing Data packet in forwarding [21]. Each PIT entry contains the following information: (*i*) the name associated with the entry; (*ii*) a list of incoming faces; (*iii*) a list of outgoing faces; (*iv*) time when the entry should expire and (*v*) any other forwarding-strategy information. Table 1 represents the data set for 30 incoming Interest packet for recording or updating the PIT.

The common replacement policies in PIT are Persistent, Random and Least Recently Used [20]. These are justified from the main contributions found through the literatures. In the following subsections, we shall briefly present these policies in relation to its functionality with replacement policies implementations.

A. Persistent Replacement Policy

Persistent is assumed as the default replacement police in NDN, which adds to speed up operations thereby reducing the complexity in implementation. It has been proposed as a solution to coordinate PIT in Interest interactions. A Persistent replacement policy requires that mostly new Interest packet should be selected and not to be rejected from its allocated memory space when there isn't free space found in its PIT. As a result, the memory space may be well under-utilized. Thus, this may result in relatively poor performance. This algorithm presents a major drawback since its reject any incoming Interest even with has less Lifetime or a high popular entry in PIT. An example of the operation of the Persistent policy is maintained as depicted in Figure 1 based on incoming Interest that presented in Table 1.

Table 1
Incoming Interest Packet Data Set

Entity Name	Incoming Face	Lifetime	Enqueuer PIT
www.google.com/	4	196	1
www.onlinecorrect.com/	1	131	2
www.ss.uni/	5	269	3
www.facebook.org/	1	220	4
www.onlinecorrect.com/	2	121	5
www.youtube.com	2	183	6
www.tm.com.my/	1	187	7
www.ss.uni/	4	211	8
www.google.com/	3	120	9
www.tm.com.my/	3	161	10
www.lelong.com.my/	3	250	11
www.tm.com.my/	4	47	12
www.internetworks.my/	2	322	13
www.onlinecorrect.com/	3	94	14
www.powervoip.com/	3	329	15
www.uobabylon.edu/	1	130	16
www.tm.com.my/	5	131	17
www.internetworks.my/	1	265	18
www.ss.uni/	1	208	19
www.google.com/	5	139	20
www.uobabylon.edu/	2	72	21
www.youtube.com	3	62	22
www.youtube.com	1	55	23
www.tm.com.my/	2	79	24
www.onlinecorrect.com/	4	100	25
www.internetworks.my/	4	262	26
www.facebook.org/	2	191	27
www.youtube.com	1	109	28
www.facebook.org/	4	239	29
www.lelong.com.my/	1	188	30

New entry record

www.slideshow.com/	3	2,5	207	flooding

PIT is full - before treatment

E_name	I_face	O_face	LT	FS
www.google.com/	4,3,5	1,2,6	166	flooding
www.onlinecorrect.com	1,2,3,4	1,5	102	flooding
www.ss.uni/	5,4,1	2,3	241	flooding
www.facebook.org/	1,2,4	3,5,6	194	flooding
www.youtube.com	2,3,1,5	4,6	158	flooding
www.tm.com.my/	1,3,4,5,2	6	163	flooding
www.lelong.com.my/	3,1	5,2	220	flooding
www.internetworks.my/	3,1,4	2,6,5	304	flooding
www.powervoip.com/	3	1,5	314	flooding
www.uobabylon.edu/	1,2	4	114	flooding

PIT - after treatment

E_name	I_face	O_face	LT	FS
www.google.com/	4,3,5	1,2,6	166	flooding
www.onlinecorrect.com	1,2,3,4	1,5	102	flooding
www.ss.uni/	5,4,1	2,3	241	flooding
www.facebook.org/	1,2,4	3,5,6	194	flooding
www.youtube.com	2,3,1,5	4,6	158	flooding
www.tm.com.my/	1,3,4,5,2	6	163	flooding
www.lelong.com.my/	3,1	5,2	220	flooding
www.internetworks.my/	3,1,4	2,6,5	304	flooding
www.powervoip.com/	3	1,5	314	flooding
www.uobabylon.edu/	1,2	4	114	flooding

New entry will be rejected

~~www.slideshow.com/~~	~~3~~	~~2,5~~	~~207~~	~~flooding~~

Figure 1: PIT before and after treatment based on Persistent policy

As we can notice from Figure 1; above, the new Interest was rejected, and the PIT before and after this operation remains the same as regards to contents. Although the Persistent policy is a simple to implement and easy to operate since it is rejected the new entries when the PIT size reached its limit. However, it is not consideration about Interest Lifetime, Interest frequency, Interest enqueue PIT time, which can effect on the performance of PIT as well as over the whole network.

B. Random Replacement Policy

Just as the name implies, this policy randomly selects a candidate data and discards it in order to present free space when necessary. This algorithm unlike the previous does not require holding any information about the access in history. The replacement policy chosen in random policy is the simplest. The discarding of Interest are randomly initiated. This algorithm seems pretty easy in implementation due to the advantage of a pseudo-random counter for the whole cache operation. It, however, consumes few resources but performs badly due to its non-usage based dimension. Its performance relies solely on real randomness of the sequence. According to a recent study in, it can also be implemented with Linear Feedback Shift Registers; however, the solution is sometimes poorly efficient. An example in practice on how the Random policy is maintained is presented in Figure 2; this is based on incoming Interest that was presented in Table 1.

New entry record

www.slideshow.com/	3	2,5	207	flooding

PIT is full - before treatment

E_name	I_face	O_face	LT	FS
www.google.com/	4,3,5	1,2,6	166	flooding
www.onlinecorrect.com	1,2,3,4	1,5	102	flooding
www.ss.uni/	5,4,1	2,3	241	flooding
www.facebook.org/	1,2,4	3,5,6	194	flooding
www.youtube.com	2,3,1,5	4,6	158	flooding
~~www.tm.com.my/~~	~~1,3,4,5,2~~	~~6~~	~~163~~	~~flooding~~
~~www.lelong.com.my/~~	~~3,1~~	~~5,2~~	~~220~~	~~flooding~~
www.internetworks.my/	3,1,4	2,6,5	304	flooding
www.powervoip.com/	3	1,5	314	flooding
www.uobabylon.edu/	1,2	4	114	flooding

PIT - after treatment

E_name	I_face	O_face	LT	FS
www.google.com/	4,3,5	1,2,6	166	flooding
www.onlinecorrect.com	1,2,3,4	1,5	102	flooding
www.games.com/	3	3,4	207	flooding
www.facebook.org/	1,2,4	3,5,6	194	flooding
www.youtube.com	2,3,1,5	4,6	158	flooding
www.slideshow.com/	3	2,5	207	flooding
www.slideshow.com/	3	2,5	207	flooding
www.internetworks.my/	3,1,4	2,6,5	304	flooding
www.powervoip.com/	3	1,5	314	flooding
www.uobabylon.edu/	1,2	4	114	flooding

Figure 2: PIT before and after treatment based on Random policy

In Random policy, when PIT reaches its limit, random entry (could be the newly created one) will be removed from the PIT. Therefore, there are two inverse cases in this policy. Best case is by removing the entry which has too long expiration time with minimum frequency (i.e., replacing the entry name: "www.lelong.com.my/", frequency: "2", Lifetime: "220" with a new entry name: www.games.com/, frequency: 1", Lifetime: "207"). While the worst case is by removing entry, which has lowest expiration time with maximum frequency (i.e., replacing the entry name: "www.tm.com.my/", frequency: "5", Lifetime: "163" with the new entry name: www.games.com/, frequency: 1", Lifetime: "207").

Although the Random policy is simple to implement in hardware. However, it less efficient than the other policy because some entries may accumulate large request counts or may have little Interest lifetime as a factor is replacing, and it could be the newly created one to be removed from PIT.

C. Least Recently Used (LRU) Replacement Policy

LRU policy is among the most popular algorithms that are those based on the Least-Recently-Used cache replacement rule. The wide popularity of this policy is attributed to its good performance. LRU algorithm tends to keep both more frequent items used in the PIT as well as quick adaptation to the potential changes in document popularity. This results in efficient performance of the overall replacement policy. In order to understand further the insight into a network caching designing and algorithms, it is important to gain a thorough understanding of the baseline LRU cache replacement policy [22][23]. In the analysis of LRU policy, due to the recently in use, the entry which with the highest access is most likely to be accessed again in the near future, and the entry that has been "least recently used" would be replaced by the PIT controller when the PIT demands a new entry adding. An example of how the LRU policy is maintained is shown in Figure 3 based on incoming Interest that we presented in Table 1. Although the LRU replacement is heuristic and relatively important as it requires a number of memories in bits to record PIT whenever an entry is accessed.

New entry record

www.slideshow.com/	3	2,5	207	flooding

PIT is full - before treatment

E_name	I_face	O_face	LT	FS
www.google.com/	4,3,5	1,2,6	166	flooding
www.onlinecorrect.com	1,2,3,4	1,5	102	flooding
~~www.ss.uni/~~	~~5,4,1~~	~~2,3~~	~~241~~	~~flooding~~
www.facebook.org/	1,2,4	3,5,6	194	flooding
www.youtube.com	2,3,1,5	4,6	158	flooding
www.tm.com.my/	1,3,4,5,2	6	163	flooding
www.lelong.com.my/	3,1	5,2	220	flooding
www.internetworks.my/	3,1,4	2,6,5	304	flooding
www.powervoip.com/	3	1,5	314	flooding
www.uobabylon.edu/	1,2	4	114	flooding

PIT - after treatment

E_name	I_face	O_face	LT	FS
www.google.com/	4,3,5	1,2,6	166	flooding
www.onlinecorrect.com	1,2,3,4	1,5	102	flooding
www.slideshow.com/	3	2,5	207	flooding
www.facebook.org/	1,2,4	3,5,6	194	flooding
www.youtube.com	2,3,1,5	4,6	158	flooding
www.tm.com.my/	1,3,4,5,2	6	163	flooding
www.lelong.com.my/	3,1	5,2	220	flooding
www.internetworks.my/	3,1,4	2,6,5	304	flooding
www.powervoip.com/	3	1,5	314	flooding
www.uobabylon.edu/	1,2	4	114	flooding

Figure 3: PIT before and after treatment based on LRU policy

In LRU, the oldest as it compared to the time of usage will be removed when PIT size reached its limit. Based on this, LRU replaces the entry that hasn't been referenced to in the longest time ((i.e., replacing the entry name: "www.ss.uni/", frequency: "3", Lifetime: "241" with a new entry name: www.games.com/, frequency: 1", Lifetime: "207").

LRU is selected as common one for ICN because of its low complexity, although its ability to identify the popular content is poor [24]. However, without consideration of factors, for example, the number of entries to keep track of increases. LRU becomes more expensive if one wants to ensure that the policy always discards the least recently used entry as well as it's harder to implement, slower, often just approximated and not care about entry lifetime as a factor. Moreover, LRU can also lead to many unnecessary cache replace [25].

III. Simulation Environment

Evaluation of performance methods are a very crucial step in evaluating the final results of any research or project [26]. Accordingly, different ICN architectures were critically analyzed in the literature review using a combined method of theoretical analysis, empirical measurements (testbed) and simulation techniques. These are deemed appropriate and popular methods of evaluating network performance, architectures, services and protocols in networking community. Usually, several researchers adhere to a specific methodology in line with the set out goals of their experiment (e.g., to evaluate quantify resource utilization, economic incentives, scalability and so on). Thus, there are many factors that could affect the experimental results such as: the network condition (e.g., available link capacity); topology selected; link delay, node mobility, background traffic load, loss-rate characteristics, disruption patterns, and other aspects (e.g., the variety of devices used) [27].

A. Selection of Simulation

Simulation has widely been the option of mimicking dynamic scenarios, mainly networks and real systems. It is a computer-based system model or generated using computer programming. Furthermore, simulation is a more flexible tool for studying the performance of various protocols [28]. NS3 is a free and open source network simulator that has been made available for teaching, research and development work under the GNU GPLv2 license [29]. It has the capability of being integrated with external animators and data analysis and visualization tools for better presentations of results [30]. From another perspective, ICN is still in its infancy though research, the community is still in the process of developing an effective and evaluation platform, including simulators, emulators, and testbeds [27]. Some of the most popular tools that are available for communication network researchers are CCNPL-Sim[31], ccnSim [32] and ndnSIM [31][21].

ndnSIM is an open source based-module that can be plugged along the NS-3 simulator to support the core features of CCN. One could use ndnSIM to analyze various CCN applications, scenarios and services as well as incorporate components developed for CCN such as routing protocols, caching and forwarding strategies for testing its efficiency. The code in NS-3 and ndnSIM is widely available for modification and updates to the community and can provide a basis for implementing ICN protocols or applications.

B. Topology and Setting Parameters

According to Pentikousis *et al*, [27] “there is no single topology that can be used to easily evaluate all aspects of the ICN paradigm”. In this research, several network topologies with different network sizes and varying number of nodes were used to test and evaluate Persistent, Random and LRU policies. More specifically, the first scenario is a classic topology, namely Tree topology consists of 7 NDN nodes, 9 consumer and 9 publishers [19]; the second scenario is an Abilene topology consists of 11 NDN nodes, 25 consumer and 10 publishers [33] and third scenarios is Germany topology consists of 50 NDN nodes, 30 consumer and 30 publishers.

Experiment setup has been created in ndnSIM-NS3 running on a machine with Intel Core(TM) i7-3612QM at 2.10 GHz CPU, 12 GBytes of RAM, and Linux Ubuntu 14.04 operating system. The experiment setup and the experiments carried out have been explained in table below:

Table 2
Performance Parameters

Parameter	Value
Simulation environment	ndnSIM-NS3
Simulation topology	Tree, Abilene, Germany
PIT replacement policy	Persistent, Random, LRU
PIT size	1000, 10000
Forward strategy	Flooding
Interest traffic generation	1000 Interest/second
Number of links	Random variable
Link delay	Random variable
Link capacity	Random variable
Interest packet size	40 byte
Interest lifetime	400ms

C. Metrics

The main step in performance evaluation is performance metrics selection. According to [34] “performance metrics can mean different things to different researcher depending on the context in which it is used”. On the other hand, It is the key phase in all performance evaluations [35] because it measures the performance of the proposed scheme. This study focuses on three metrics; Interest drop rate, Interest retransmission and delay time metrics that could be used to measure the performance of the study objective.

- Interest drop rate: it can be explained as it is the percentage of dropped interest packets among all the incoming Interest packet.
- Delay time: it perceived by the consumer, which measures the waiting time to receive a given content after sending its request.
- Retransmission: it perceived by the consumer that is measured the number of Interest retransmissions.

IV. RESULTS AND DISCUSSION

In our simulations, we compared the Interest drop rate, Interest retransmission, and delay time for Persistent, Random and LRU replacement policies over three different network topologies, i.e., Tree, Germany and Abilene with PIT sizes 1000 and10000 for entries. On Tree topology (see Figures 4, 5, and 6) overall performance metrics, we observed the Persistent policy performed poorly for PIT size 1000. Especially in terms of the packet drop rate; this was because the policy rejects the incoming Interest directory when PIT is full. Thus, will be leading to an increase rate in the Interest retransmission. Another justification was because when the PIT overflows, consumers’ Interests will be discarded from the routers. Based on this, consumers will experience an increasing retransmission rate. Contrarily, as the PIT size increase to 10000, there was no difference seen in the Interest drop rate as well as the delay. On the other hand, LRU policy was given a good performance score in terms of Interest retransmission when the Interest drops were lower in PIT size 1000 and 10000.

The second simulation was run on a Germany network topology (see Figures 7, 8, and 10). The results show that the dropping is seemingly very close to all policies of PIT size 1000. While in the case of the PIT size being increased to 10000, the Interest drop rate was reduced. The intermediate nodes were forwarding packets to higher-level routers until it reached the Data content. On the other hand, the delay and retransmission in both Persistent and Random probably are better than LRU; this holds from the fact of a huge number of NDN router in our scenario caused incremental delay time as well as the Interest packet retransmission since each policy needs time in order to record or update its entity over all PITs presents in the routers.

In the scenarios (see Figures 10, 11, and 12), where the PIT size was set at 1000 and 10000, the Random policy recorded less efficient measure than the other policies which were attributed to the large accumulation of request counts that were replaced. Hence, it may increase the Interest drop and Interest retransmission. While LRU achieved the lowest Interest drop rate, of about 35.6% with PIT size 1000 and 35.8% of PIT size 10000 as compared to the Interest drop rate of other policies. Moreover, the figures showed the results for Persistent policy achieving the highest performance rate for PIT size 10000. Due to the increasing capacity of PIT, the Interest drops and retransmission decreased. The detailed results of all policies against different topologies with different PIT size are given in Table 3.

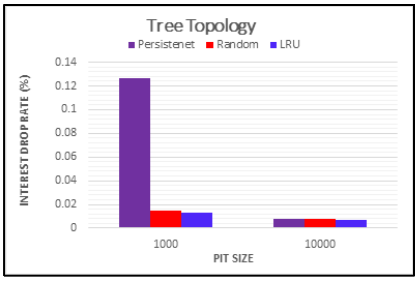

Figure 4: Interest Packet Rate on Tree Topology

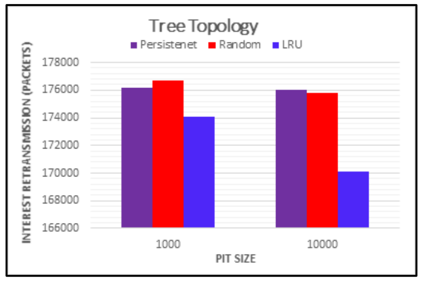

Figure 5: Interest Retransmission on Tree Topology

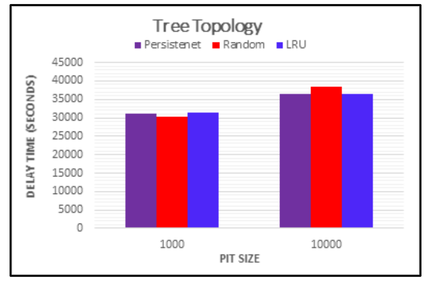

Figure 6: Delay time on Tree Topology

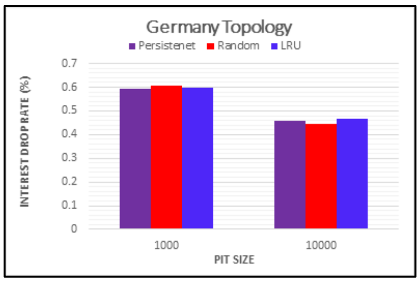

Figure 7: Interest Packet Rate on Abilene Topology

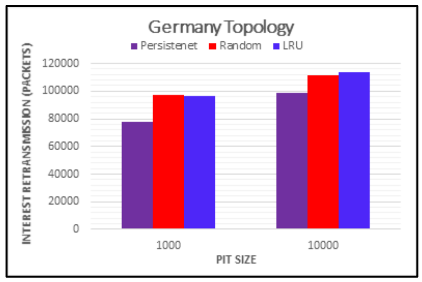

Figure 8: Interest Retransmission on Abilene Topology

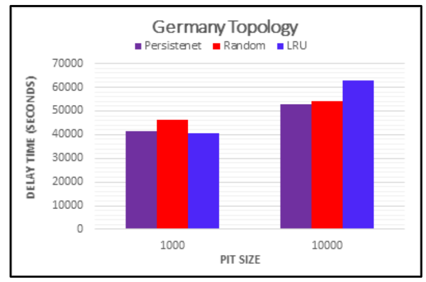

Figure 9: Delay Time on Abilene Topology

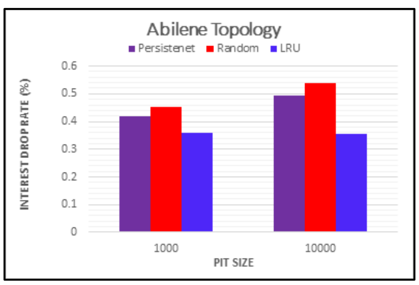

Figure 10: Interest Packet Rate on Abilene Topology

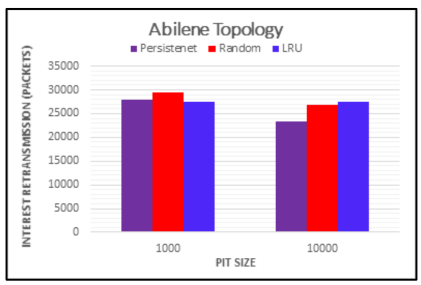

Figure 11: Interest Retransmission on Abilene Topology

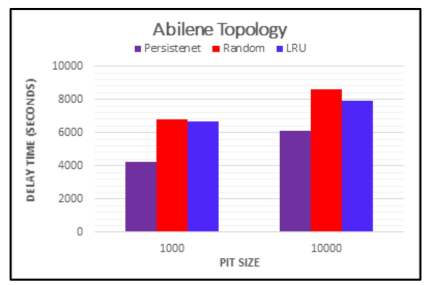

Figure 12: Delay Time on Abilene Topology

Table 3
Interest drop rate, Retransmission and Delay of all PIT replacement Policies on different topologies

Topology	PIT size		Policy	Metrics		
	1000	10000		Interest Drop Rate (%)	Retransmission (packets)	Delay (seconds)
Tree	√		Persistent	0.13786584	176188	31223
		√		0.007446586	176046	36454
	√		Random	0.01552412	176688	30379
		√		0.007910396	175810	38532
	√		LRU	0.013125912	174046	31537
		√		0.007338039	170062	364545
Germany	√		Persistent	0.592130502	77823	41521
		√		0.458530967	98409	52730
	√		Random	0.608732233	97306	46399
		√		0.444195796	111843	53947
	√		LRU	0.59654887	96514	40600
		√		0.4265421	114073	62846
Abilene	√		Persistent	0.492940895	27915	4207
		√		0.418084894	23376	6118
	√		Random	0.539449137	29518	6790
		√		0.45155968	26911	8604
	√		LRU	0.356986568	27430	6651
		√		0.358242554	27841	7948

V. CONCLUSION

Replacement policy in PIT represents one of the most important factors, which determines the effectiveness of a PIT. Moreover, the primary goal of intermediate nodes (NDN routers) is to manage PIT entries in order to accommodate Interest efficiently when a new Interest arrives, and the PIT of a router is full. In this paper, we measured the performance of Persistent, Random and LRU policies in terms of Interest drop rate, delay time and Interest retransmission, which were tested on Abilene, Tree, Germany network topologies using the ndnSIM simulator. Based on our simulation results, we can argue that the configuration of topology and parameters may affect the general performance of policies for contemporary workload. Therefore, the replacement policies have different effect on instruction and entries' PIT. Based on our results, Random and LRU showed the highest delay on all topologies as compared to Persistent. While in terms of Interest drop ratio, Persistent and Random recorded the highest on all topologies as compared to LRU. Finally, Random policy resulted in the highest delay on all topologies as compared to Persistent and LRU with PIT size 10000. Nevertheless, Persistent had less efficiency on PIT size 1000. The results from our study provide an essential starting point for new researches in PIT replacement heuristics for contemporary workload or selective building of less used cache approaches. As a direction for future work, we shall focus on designing new policy that will deal with Interest request and Interest Lifetime in order to remove only the entries that are kept for a long time in the PIT.

ACKNOWLEDGMENT

This research is supported by the Ministry of Higher Education Malaysia research grant scheme FRGS/UUM/13258

REFERENCES

[1] M. Gallo, "Traffic and Resource Management in Content-Centric Networks : Design and Evaluation," TELECOM ParisTech, 2014.

[2] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A Data-Oriented (and Beyond) Network Architecture," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 1--12, Oct. 2007.

[3] N. Fotiou, P. Nikander, D. Trossen, and G. C. Polyzos, "Developing Information Networking Further: From PSIRP to PURSUIT," in *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*, 2010, pp. 1–13.

[4] C. Dannewitz, "NetInf: An Information-Centric Design for the Future Internet," in *Proc. 3rd GI/ITG KuVS Workshop on The Future Internet*, 2009, pp. 1–3.

[5] B. Ahlgren, M. D'Ambrosio, E. Davies, A. Eriksson, J. Frtunikj, and A. El, "The Network of Information: Architecture and Applications," SAIL, 2013.

[6] F. Almeida, M. T. Andrade, N. B. Melazzi, R. Walker, H. Hussmann, and I. S. Venieris, *Enhancing the Internet with CONVERGENCE System*, First. London: Springer Science \& Business Media, 2014.

[7] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking Named Content," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, 2009, vol. 55, no. 1, pp. 1–12.

[8] F. Oehlmann, "Content-Centric Networking," *Netw. Archit. Serv.*, vol. 43, pp. 11–18, 2013.

[9] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. Thornton, D. Smetters, B. Zhang, G. Tsudik, C. Kc, D. Krioukov, D. Massey, C. Papadopoulos, T. Abdelzaher, L. Wang, P. Crowley, and E. Yeh, "Named Data Networking (NDN) Project," 2010.

[10] A. Jaffri, Z. Ahmad, and M. Tahir, "Named Data Networking (NDN), New Approach to Future Internet Architecture Design : A Survey," *Int. J. Informatics Commun. Technol.*, vol. 2, no. 3, pp. 155–164, 2013.

[11] D. Saxena, V. Raychoudhury, S. Neeraj, C. Becker, and J. Cao, "Named Data Networking: A Survey," *Comput. Sci. Rev. - Elsevier*, vol. 19, pp. 15--55, 2016.

[12] G. M. Brito, P. B. Velloso, and I. M. Moraes, *Information Centric Networks: A New Paradigm for the Internet*, 1st ed. Croydon, Surrey CR0 4YY: John Wiley \& Sons, 2013.

[13] R. Alubady, M. Salman, S. Hassan, and A. Habbal, "Review of Name Resolution and Data Routing for Information Centric Networking," in *The 4th International Conference on Internet Applications, Protocols and Services (NETAPPS2015)*, 2015, pp. 48–53.

[14] A. Raychaudhuri, Dipankar and Nagaraja, Kiran and Venkataramani, "MobilityFirst: A Robust and Trustworthy Mobility-Centric Architecture for the Future Internet," *ACM SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 16, no. 3, pp. 2--13, 2012.

[15] Y. Wang, K. He, H. Dai, W. Meng, J. Jiang, B. Liu, and Y. Chen, "Scalable Name Lookup in NDN Using Effective Name Component Encoding," in *2012 IEEE 32nd International Conference on Distributed Computing Systems*, 2012, pp. 688–697.

[16] D. Saxena, V. Raychoudhury, N. Suri, C. Becker, and J. Cao, "Named Data Networking: A survey," *Comput. Sci. Rev.*, vol. 19, pp. 15–55, 2016.

[17] M. Virgilio and P. Torino, "PIT Overload Analysis in Content Centric Networks," in *Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking*, 2013, pp. 67–72.

[18] H. Al-Zoubi, A. Milenkovic, and M. Milenkovic, "Performance Evaluation of Cache Replacement Policies for the SPEC CPU2000 Benchmark Suite," in *PProceedings of the 42nd annual Southeast regional conference*, 2004, pp. 267--272.

[19] I. Ud Din, S. Hassan, and A. Habbal, "Comparison of Caching Strategies on Different Topologies in Information-Centric Networking," in *The 4th International Conference on Internet Applications, Protocols and Services (NETAPPS2015) Comparison*, 2015, vol. 21, no. 11, pp. 38–41.

[20] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM: NDN Simulator for NS-3," *TUniversity of California, Los Angeles, Tech. Rep. NDN-0005*, California, Los Angeles, pp. 1–7, 2012.

[21] S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM 2 . 0 : A New Version of the NDN Simulator for NS-3," *University of California, Los Angeles, Tech. Rep. NDN-0028*, California, Los Angeles, pp. 1–8, 2015.

[22] P. R. Jelenković and X. Kang, "Characterizing the Miss Sequence of the LRU Cache," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 36, no. 2, p. 119, 2008.

[23] M. Dehghan, B. Jiang, A. Dabirmoghaddam, and D. Towsley, "On the Analysis of Caches with Pending Interest Tables," in *Proceedings of the 2nd International Conference on Information-Centric Networking*, 2015, pp. 69–78.

[24] B. Feng, H. Zhou, M. Zhang, and H. Zhang, "Cache-Filter : A Cache Permission Policy for Information-Centric Networking," *KSII Trans. INTERNET Inf. Syst.*, vol. 9, no. 12, pp. 4912–4933, 2015.

[25] H. Dai, Y. Wang, H. Wu, J. Lu, and B. Liu, "Towards line-speed and accurate on-line popularity monitoring on NDN routers," in *IEEE International Workshop on Quality of Service, IWQoS*, 2014, pp. 178–187.

[26] R. Jain, *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*, vol. 491. 1991.

[27] K. Pentikousis, B. Ohlman, D. Corujo, G. Boggia, G. Tyson, E. Davies, P. Mahadevan, S. Spirou, A. Molinaro, D. Gellert, and Others, "ICN Baseline Scenarios and Evaluation Methodology," *ICNRG-Draft-Pentikousis-iICN-Scenarios-04*, pp. 1–64, 2013.

[28] R. Alubady, M. Al-samman, S. Hassan, S. Arif, and A. Habbal, "Internet Protocol MANET vs Named Data MANET : A Critical Evaluation," in *The 4th International Conference on Internet Applications, Protocols and Services (NETAPPS2015) Internet*, 2015, pp. 70–76.

[29] M. R. Kumar and N. R. Babu, "A Simple Analysis on Novel Based Open Source Network Simulation Tools for Mobile Ad Hoc Networks," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 3, no. 9, pp. 856–862, 2013.

[30] V. Dimitrov and V. Koptchev, "PSIRP project – Publish-Subscribe Internet Routing Paradigm. New Ideas for Future Internet," in *Proceedings of the 11th International Conference on Computer Systems and Technologies - CompSysTech'10*, 2010, pp. 167–171.

[31] X. Jiang, J. Bi, and Y. Wang, "MCBS: Matrix Computation Based Simulator of NDN," *J. Comput.*, vol. 9, no. 9, pp. 2007–2012, 2014.

[32] R. Chiocchetti, D. Rossi, and G. Rossini, "ccnSim: An Highly Ccalable CCN Simulator," *IEEE Int. Conf. Commun.*, pp. 2309–2314, 2013.
[33] M. Mangili, F. Martignon, and A. Capone, "A Comparative Study of Content-Centric and Content-Distribution Networks: Performance and Bounds," in *Global Communications Conference (GLOBECOM), 2013 IEEE*, 2013, pp. 1403–1409.
[34] A. Habbal, "TCP SINTOK : Transmission Control Protocol with Delay-Based Loss Detection and Contention Avoidance Mechanisms for Mobile Ad Hoc Networks," Universiti Utara Malalysia (UUM), 2014.
[35] O. M. D. Al-Momani, "Dynamic Redundancy Forward Error Correction Mechanism for the Enhancement of Internet-Based Video Streaming," Universiti Utara Malalysia (UUM), 2010.