JURNAL ONLINE INFORMATIKA
JOIN (Jurnal Online Informatika)
Volume 7 No. 2 | December 2022: 193-202
DOI: 10.15575/join.v7i2.875
p-ISSN: 2528-1682
e-ISSN: 2527-9165

Performance Analysis of Cache Replacement Algorithm using Virtual Named Data Network Nodes

Leanna Vidya Yovita [1], **Tody Ariefianto Wibowo** [2], **Ade Aditya Ramadha** [3], **Gregorius Pradana S.** [4], **Sevierda Raniprima** [5]
[1,2,3,4,5]School of Electrical Engineering, Telkom University, Indonesia

Article Info

Article history:

Received April 05, 2022
Revised April 21, 2022
Accepted August 30, 2022
Published December 26, 2022

Keywords:

Caching
NDN
Performance
Replacement
Virtual node

ABSTRACT

As a future internet candidate, named Data Network (NDN) provides more efficient communication than TCP/IP network. Unlike TCP/IP, consumer requests in NDN are sent based on content, not the address. The previous study evaluated the NDN performance using a simulator. In this research, we modeled the system using virtual NDN nodes, making the model more relevant to the real NDN. As an essential component in every NDN router, the content store (CS) has a function to keep the data. We use First In First Out (FIFO) and Least Recetly Used (LRU) in our nodes as cache replacement algorithms. The in-depth exploration is done using various scenarios. The result shows that the cache hit ratio (CHR) increases if the size of the CS, the number of interests, and the number of consumers increases. CHR decreases as the number of producers and the number of prefixes increase. As CHR increases, round trip time (RTT) decreases. LRU provides better performance for all cases: higher CHR of 5-15% and lower RTT of 1-10% than FIFO.

Corresponding Author:

Leanna Vidya Yovita
School of Electrical Engineering,
Telkom University,
Jl. Telekomunikasi no. 1, Bojong Soang, Bandung, Indonesia
Email: leanna@telkomuniversity.ac.id

1. INTRODUCTION

Named Data Network (NDN) is a network architecture that emphasizes content rather than server location (address). In NDN, the request is addressed to the name of the content, no matter where the content is. This means whichever node owns the requested content will deliver the intended content to the consumer. This concept is very much different from the TCP/IP network. On a TCP/IP network, each consumer will request data addressed to a specific server address, causing only that server can respond. Therefore, the TCP/IP network is also called a host-based system.

On a TCP/IP network, the same and repeated data requests from multiple consumers will increase the network load. For example, in figure 1, consumer AA wants specific data. Consumer AA has to send the request to a certain server. The request is sent using an interest packet from consumer AA through router A, router C, router E, router F, and finally reaches the server. The server then sends the requested data to the consumer AA. If consumer BB wants the same data as consumer AA, consumer BB will send requests through router B, router A, router C, router E, router F, and finally, the request message reaches the server. The server then sent the requested data to the consumer BB. Even though the consumer requested the same data on an IP network, the request packet is addressed to the same server, so other nodes cannot respond. The high demand of consumers for the same data will burden the network and server.

The NDN architecture causes content placed on the node closest to the consumer so that data communication will be more efficient [1]. Each NDN router node has a content store (CS) to keep data that has been requested by consumers before. The first request will usually go to the content's original owner node (producer) because a copy is not yet on the network. When there is a request for content from a consumer, the content will be sent to the consumer reversely through the request packet's path. The content will be stored in the NDN nodes on the path, so the nodes that do not initially have content will keep a copy.

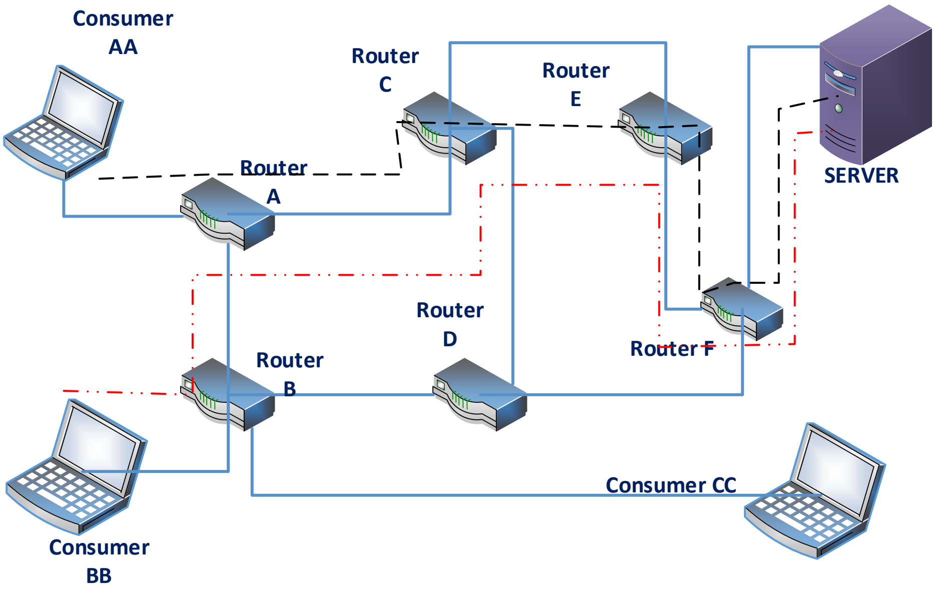


Figure 1. the illustration of a network with repeated requests for the same data

Content stores (CS) that keep copies of content have limited capacity. Therefore, several strategies were carried out on the CS: content placement, cache content selection, and cache policy [2] [3] [4]. The content placement algorithm determines where content will be placed on the network [3]. Cache content selection algorithm will choose which content should be stored and deleted if the content store is full [5]. Cache policy can be related to many things, such as determining the cooperation strategy between router nodes in determining content storage [6], defining special divisions on the content store for multiclass content [7] [8], and so forth. Thus, when the CS is full, obsolete content must be removed to be replaced with fresher content [9]. Some cache replacement algorithms used are FIFO and LRU [9]. In FIFO, the oldest content will be removed from the content store whenever new content is decided to be saved [10]. LRU is an algorithm that eliminates the most aged content not requested by the consumer to be replaced with new content.

There are two types of packets on NDN: interest packets and data packets. Interest packets are packets sent by consumers to request content, while data packets contain the information requested by consumers.

An NDN router has three crucial components: content store (CS), Pending Interest Table (PIT), and Forwarding Information Based (FIB). The consumer will send interest packets to request content in the network. CS is used to keep the data that consumers have requested. PIT stores information related to the data requested by the consumer that has not been replied to with a data packet. Using PIT, the router can find out what content is requested through the router node and has not received a reply. FIB stores information about the next-hop node to send interest packets until it arrives at the node with the consumer desired data.

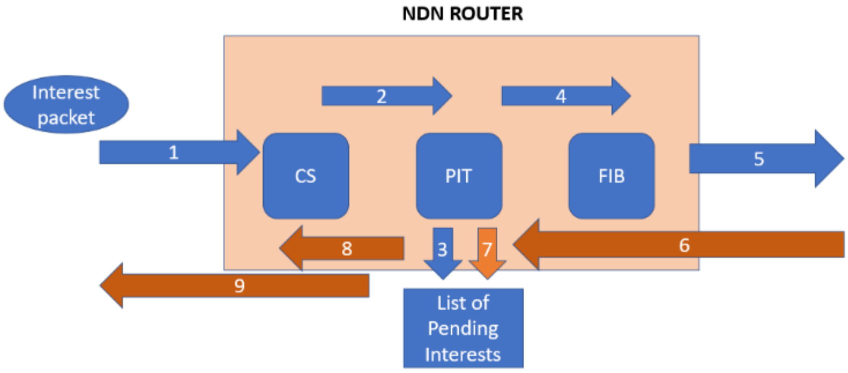


Figure 2. Components of NDN Router and the processing of packets

Figure 1 illustrates when an interest packet comes to an NDN router. Based on Figure 1, if the interest packet enters the NDN router, the router will first check the CS to find out whether the requested data is in the CS or not (arrow 1). If the data doesn't exist, the request is forwarded to PIT (arrow 2). The router updates the pending-content lists in the PIT (arrow 3). The pending-content is content that the router cannot immediately provide. After updating the PIT, the router will check FIB to find the next-hop node to forward the interest packet (arrow 4). The request is forwarded again to other nodes based on information from FIB (arrow 5). When the data response comes to the NDN router (arrow 6), the PIT will be updated to ensure that the request is no longer pending (arrow 7). The data will be stored in CS (arrow 8) and forwarded to the requesting consumer (arrow 9).

We must use the CS efficiently because it is a limited resource. The cache content selection strategy selects which content to keep and which to deleted. These mechanisms are done parallelly. For example, choose to keep recently requested content and delete the oldest one. Another example mechanism is storing data that has just entered the router node and deleting the last order data in the content store.

Cache replacement is an important caching strategy in deciding which content should be removed from the content store [14]. Based on a previous study, caching performance can be improved by using an efficient caching replacement method [15]. The parameters used by the replacement rule can be classified as recency, the number of content requests, message size, cost to reach the object (such as the number of hops between user and server), and access delay [15].

Common methods of cache replacement on NDN are FIFO and LRU. These two algorithms have different considerations in removing content from the content store. FIFO deleted the oldest content when the content store was full. The frequency of requests and content recency is not considered [16]. In LRU, deleted content is content that has not been accessed for the longest time [16]. These two cache replacement algorithms are widely used in NDN cache systems. Research from Dehghan et al. uses FIFO and LRU cache replacement algorithms for NDN systems with different content storage rules for two operators. [17]. Shailendra et al. used a cache replacement algorithm and tested its performance on the Sprint topology using the NDNsim simulator [10].

Previous studies have compared the performance of cache replacement algorithms using simulators [10] [18] [19] [7]. The simulator provides convenience in testing involving multiple nodes. However, many simplifications are made in the simulator, such as not considering the data processing and coordination processes as in actual NDN nodes. In previous studies, generally limited testing was carried out for changes in the size of the content store or changes in topology. Therefore, further research and testing are needed with a more realistic model and more possible cases.

Most of the research conducted is modeling the network and testing the performance of the cache using a simulator. The simulator commonly used is NDNsim, as in the study conducted by Shailendra et al. and Situmorang et al. [10] [11], using MatLab as in the research undertaken by Alamsyah et al. [12] and Yovita et al. [7]. Other experiments were also carried out using a CCNx-based prototype by Carogfiglio et al. [13].

In this research, NDN nodes were modelled using virtual nodes running on Mini-NDN based on the Indonesia Digital Network (IDN) topology, making it more realistic. Each node has its own mechanism and resources regarding caching. Five scenarios were carried out that explored various sizes of content stores, the number of interests, the number of consumers, the number of producers, and the number of content prefixes. All tests use two cache replacement algorithms, namely FIFO and LRU.

The remainder of the paper is organized as follows. Section 2 describes the method. Section 3 presents the result and discussion. Finally, Section 4 concludes the paper.

2. METHOD

To create an NDN system consisting of virtual nodes, we use Mini-NDN [20]. The topology is modeled according to the Indonesia Digital Network (IDN) network. The location of the nodes is adjusted to the IDN network to connect essential points in Indonesia. The government, together with stakeholders, are building the digitization of Indonesia through the Indonesia Digital Network (IDN), which focuses on developing and providing connectivity that connects all points to remote areas. The digital telecommunication infrastructure is deployed, such as fiber optic and terrestrial cables on land, submarine cables at sea, and satellites in the air.

In our research, each NDN modeled node has NDN Forwarding Daemon (NFD). So that all node behaves like actual NDN node. In this study, an environment modeled closer to the real implementation of NDN is modeled. Each node is configured separately as a consumer, router, or producer. Producer nodes are nodes that produce content. Consumer nodes are nodes that can generate requests for content. Router nodes are intermediate nodes on the network and can forward request data and store them according to the NDN mechanism.

There are five scenarios carried out in this study: scenarios A, B, C, D, and E. Scenario A evaluates the various CS size. Scenario B analyzes the effect of changes in the frequency of interest. Scenario C is conducted to see the effect of the number of consumers. We analyze the effect of the number of producers in scenario D, and scenario E is conducted to see the effect of the various number of the prefix.

For the evaluation using various values in one parameter, a fixed value is used for the other parameters, according to Table 1. Furthermore, we can analyze the impact of changes in the parameter's value being tested. We also built the system with FIFO and LRU cache replacement algorithms for every scenario and evaluated 10,000 total requests from consumers.

The determination of nodes that act as producers and consumers is done based on distance considerations. The Semarang and Medan nodes are set as the consumer nodes for the two-consumers scenario. For the scenario with five consumers, we set the consumer group consisting of Semarang, Medan, Pontianak, Makassar, and Denpasar. For ten-consumer scenario, we set the consumer group consisting of Semarang, Medan, Pontianak, Makassar, Denpasar, Balikpapan, Palembang, Aceh, Surabaya and Batam.

For the one-node producer scenario, Jakarta is set. We set Jakarta and Palembang as the producer nodes for the two-producer scenario. For the five-producer scenario, we set Jakarta, Palembang, Balikpapan, Batam, and Surabaya as a group of producers. And for ten-producer scenario, we set producer group consist of Jakarta, Palembang, Balikpapan, Batam, Surabaya, Padang, Ambon, Yogyakarta, Bandung, and Aceh.

The performance parameters analyzed were cache hit ratio (CHR) and round trip time (RTT). The cache hit ratio (CHR) parameter is the ratio between hit requests to the total number of requests from consumers. The term ‘hit request’ is a consumer request fulfilled by the NDN router. This means the request is served directly by the NDN router node. The greater the CHR value, the better the performance. RTT shows how long the average time takes from the start of the consumer requesting data until the data is received. The larger the RTT value, the worse the performance.

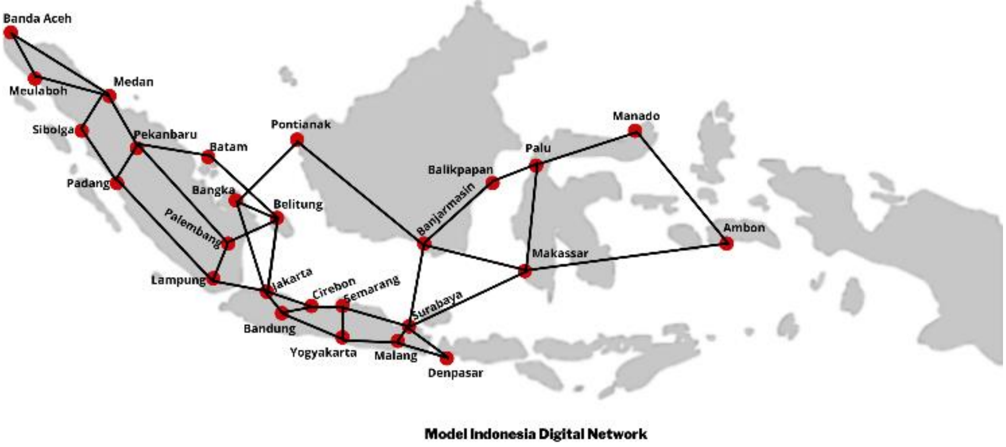


Figure 3. Indonesia Digital Network (IDN) Topology

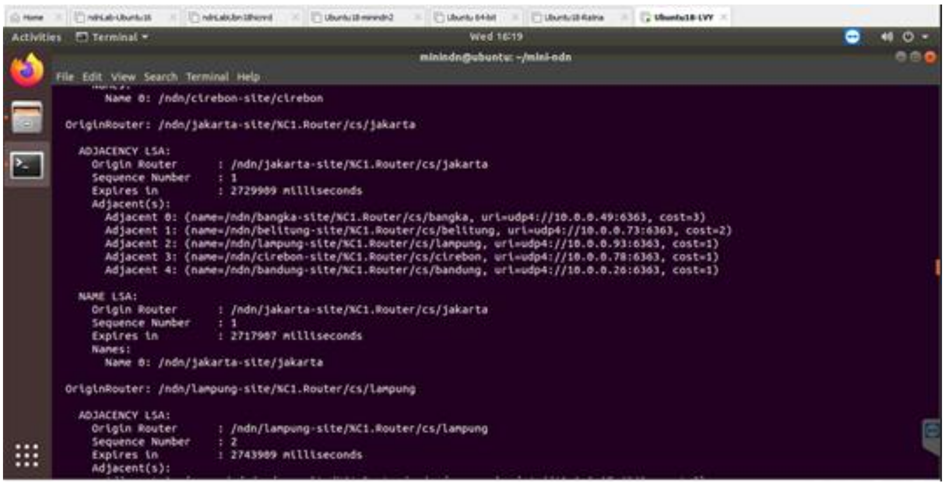
Table 1. Testing parameters

Scenarios **Parameters**	**A**	**B**	**C**	**D**	**E**
CS Size (packets)	10, 20, 30, 40	30	30	30	30
Frequency of interests (int/s)	10	10,30, 60	10	60	60
Number of Consumers	2	2	2,5,10	5	5
Number of Producers	1	1	1	1, 2, 5, 10	1
Number of contents for each producer	100	100	100	100	10, 50, 100, 200, 300

3. RESULTS AND DISCUSSION

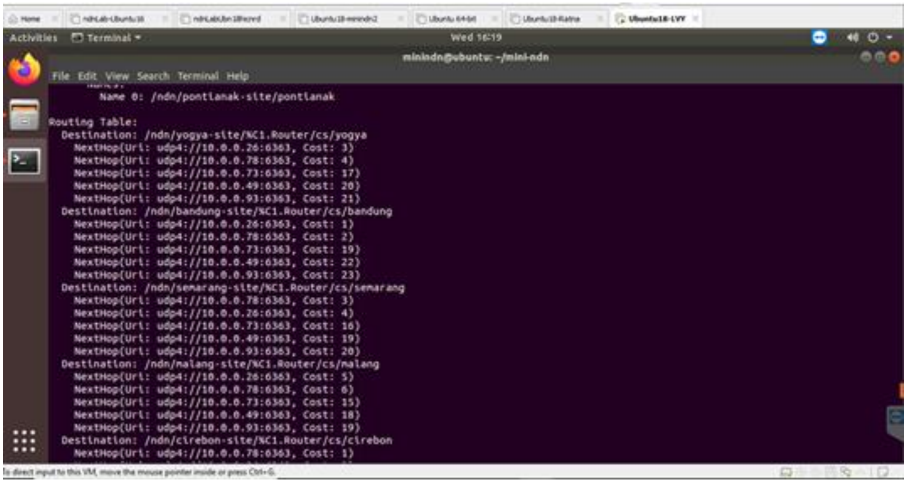
Discussion of performance evaluation based on test results is described in this section based on the type of scenario and discussed for a cache hit ratio (CHR) and Round Trip Time (RTT) parameters. Because we have to make all nodes virtual NDN with connectivity as planned topology, we have to enable it on mini-NDN. As in a real network, every node must be recognized one to another. Furthermore, it is necessary to run a routing algorithm. The routing algorithm in our NDN system uses NLSR (Named-Data Link State Routing Protocol). Figure 4 shows that the NLSR routing protocol is active. The Link-state Advertisement (LSA) message carries information about the existing nodes on the network. Every NDN router knows each other because of this mechanism.

After routing convergence occurs, a routing table exists at each node to support sending requests and data, as shown in Figure 5. The prefix is a key that defines the content on NDN. The NDN nodes recognize content according to its prefix. Without giving a name for the content, the consumer cannot ask about the content in the system. Figure 6 shows that the prefixes already exist in our system model.



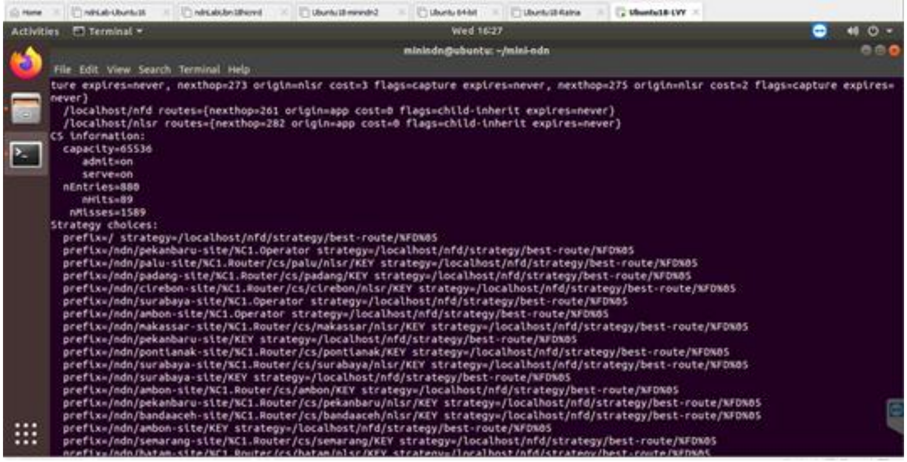
```
      Name 0: /ndn/cirebon-site/cirebon

 OriginRouter: /ndn/jakarta-site/%C1.Router/cs/jakarta

  ADJACENCY LSA:
    Origin Router       : /ndn/jakarta-site/%C1.Router/cs/jakarta
    Sequence Number     : 1
    Expires in          : 2729989 milliseconds
    Adjacent(s):
      Adjacent 0: (name=/ndn/bangka-site/%C1.Router/cs/bangka, uri=udp4://10.0.0.49:6363, cost=3)
      Adjacent 1: (name=/ndn/belitung-site/%C1.Router/cs/belitung, uri=udp4://10.0.0.73:6363, cost=2)
      Adjacent 2: (name=/ndn/lampung-site/%C1.Router/cs/lampung, uri=udp4://10.0.0.93:6363, cost=1)
      Adjacent 3: (name=/ndn/cirebon-site/%C1.Router/cs/cirebon, uri=udp4://10.0.0.78:6363, cost=1)
      Adjacent 4: (name=/ndn/bandung-site/%C1.Router/cs/bandung, uri=udp4://10.0.0.26:6363, cost=1)

  NAME LSA:
    Origin Router       : /ndn/jakarta-site/%C1.Router/cs/jakarta
    Sequence Number     : 1
    Expires in          : 2717987 milliseconds
    Names:
      Name 0: /ndn/jakarta-site/jakarta

 OriginRouter: /ndn/lampung-site/%C1.Router/cs/lampung

  ADJACENCY LSA:
    Origin Router       : /ndn/lampung-site/%C1.Router/cs/lampung
    Sequence Number     : 2
    Expires in          : 2743989 milliseconds
    Adjacent(s):
```

Figure 4. The NLSR routing protocol is active on the node (sample node: Jakarta).



```
      Name 0: /ndn/pontianak-site/pontianak

Routing Table:
  Destination: /ndn/yogya-site/%C1.Router/cs/yogya
    NextHop(Uri: udp4://10.0.0.26:6363, Cost: 3)
    NextHop(Uri: udp4://10.0.0.78:6363, Cost: 4)
    NextHop(Uri: udp4://10.0.0.73:6363, Cost: 17)
    NextHop(Uri: udp4://10.0.0.49:6363, Cost: 20)
    NextHop(Uri: udp4://10.0.0.93:6363, Cost: 21)
  Destination: /ndn/bandung-site/%C1.Router/cs/bandung
    NextHop(Uri: udp4://10.0.0.26:6363, Cost: 1)
    NextHop(Uri: udp4://10.0.0.78:6363, Cost: 2)
    NextHop(Uri: udp4://10.0.0.73:6363, Cost: 19)
    NextHop(Uri: udp4://10.0.0.49:6363, Cost: 22)
    NextHop(Uri: udp4://10.0.0.93:6363, Cost: 23)
  Destination: /ndn/semarang-site/%C1.Router/cs/semarang
    NextHop(Uri: udp4://10.0.0.78:6363, Cost: 3)
    NextHop(Uri: udp4://10.0.0.26:6363, Cost: 4)
    NextHop(Uri: udp4://10.0.0.73:6363, Cost: 16)
    NextHop(Uri: udp4://10.0.0.49:6363, Cost: 19)
    NextHop(Uri: udp4://10.0.0.93:6363, Cost: 20)
  Destination: /ndn/malang-site/%C1.Router/cs/malang
    NextHop(Uri: udp4://10.0.0.26:6363, Cost: 5)
    NextHop(Uri: udp4://10.0.0.78:6363, Cost: 6)
    NextHop(Uri: udp4://10.0.0.73:6363, Cost: 15)
    NextHop(Uri: udp4://10.0.0.49:6363, Cost: 18)
    NextHop(Uri: udp4://10.0.0.93:6363, Cost: 19)
  Destination: /ndn/cirebon-site/%C1.Router/cs/cirebon
    NextHop(Uri: udp4://10.0.0.78:6363, Cost: 1)
```

Figure 5. The routing table is active (sample on the Jakarta node)



```
ture expires=never, nexthop=273 origin=nlsr cost=3 flags=capture expires=never, nexthop=275 origin=nlsr cost=2 flags=capture expires=
never}
  /localhost/nfd routes={nexthop=261 origin=app cost=0 flags=child-inherit expires=never}
  /localhost/nlsr routes={nexthop=282 origin=app cost=0 flags=child-inherit expires=never}
CS information:
  capacity=65536
     admit=on
     serve=on
  nEntries=880
     nHits=89
   nMisses=1589
Strategy choices:
  prefix=/ strategy=/localhost/nfd/strategy/best-route/%FD%05
  prefix=/ndn/pekanbaru-site/%C1.Operator strategy=/localhost/nfd/strategy/best-route/%FD%05
  prefix=/ndn/palu-site/%C1.Router/cs/palu/nlsr/KEY strategy=/localhost/nfd/strategy/best-route/%FD%05
  prefix=/ndn/padang-site/%C1.Router/cs/padang/KEY strategy=/localhost/nfd/strategy/best-route/%FD%05
  prefix=/ndn/cirebon-site/%C1.Router/cs/cirebon/nlsr/KEY strategy=/localhost/nfd/strategy/best-route/%FD%05
  prefix=/ndn/surabaya-site/%C1.Operator strategy=/localhost/nfd/strategy/best-route/%FD%05
  prefix=/ndn/ambon-site/%C1.Operator strategy=/localhost/nfd/strategy/best-route/%FD%05
  prefix=/ndn/makassar-site/%C1.Router/cs/makassar/nlsr/KEY strategy=/localhost/nfd/strategy/best-route/%FD%05
  prefix=/ndn/pekanbaru-site/KEY strategy=/localhost/nfd/strategy/best-route/%FD%05
  prefix=/ndn/pontianak-site/%C1.Router/cs/pontianak/KEY strategy=/localhost/nfd/strategy/best-route/%FD%05
  prefix=/ndn/surabaya-site/%C1.Router/cs/surabaya/nlsr/KEY strategy=/localhost/nfd/strategy/best-route/%FD%05
  prefix=/ndn/surabaya-site/KEY strategy=/localhost/nfd/strategy/best-route/%FD%05
  prefix=/ndn/ambon-site/%C1.Router/cs/ambon/KEY strategy=/localhost/nfd/strategy/best-route/%FD%05
  prefix=/ndn/pekanbaru-site/%C1.Router/cs/pekanbaru/nlsr/KEY strategy=/localhost/nfd/strategy/best-route/%FD%05
  prefix=/ndn/bandaaceh-site/%C1.Router/cs/bandaaceh/nlsr/KEY strategy=/localhost/nfd/strategy/best-route/%FD%05
  prefix=/ndn/ambon-site/KEY strategy=/localhost/nfd/strategy/best-route/%FD%05
  prefix=/ndn/semarang-site/%C1.Router/cs/semarang/KEY strategy=/localhost/nfd/strategy/best-route/%FD%05
  prefix=/ndn/batam-site/%C1.Router/cs/batam/nlsr/KEY strategy=/localhost/nfd/strategy/best-route/%FD%05
```

Figure 6. The prefix is set for the producer node

3.1 A scenario of Various Content Store Size

The evaluation of the various size of the Content Store (CS) shows that the larger the CS size, the higher the cache hit ratio. Tests using the FIFO and LRU caching algorithms also show differences in performance. LRU provides greater CHR 5-12% than FIFO. The bigger the CS size, the more significant the difference in performance. Changes in the size of the CS also affect the RTT. FIFO and LRU provide a difference in value in the RTT of 2-10%. The larger the size of the CS, the greater the difference in the RTT. LRU provides a smaller RTT value compared to FIFO.

3.2 The scenario of various number of interests

The number of interests describes the number of request from consumers. Changes in the number of interests per second affect the cache hit ratio value. LRU provides 11-15% higher CHR than FIFO. A large number of interests packet, in this case, 60 interests per second, causes consumers can request more variety of content. At a small number of requests, CS with size 30 can still accommodate well. LRU can accommodate a varying number of interests. With CS size limitations, FIFO is not as good as LRU in managing content in CS. Changes in the number of interests also affect RTT, where LRU gives a 3-6% lower RTT value than FIFO.

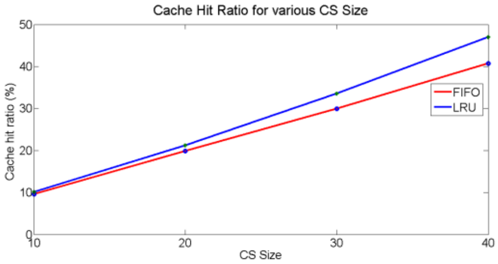


Figure 7. CHR for various content store size

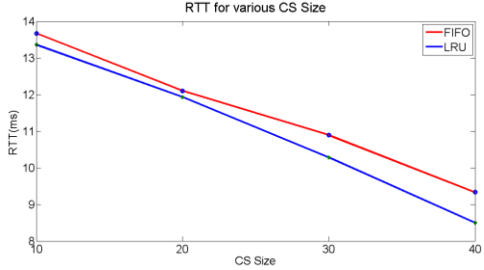


Figure 8. RTT for various content store size

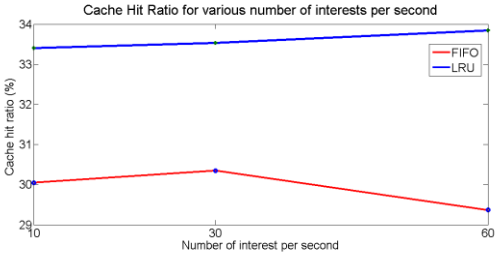


Figure 9. CHR for various interest rate

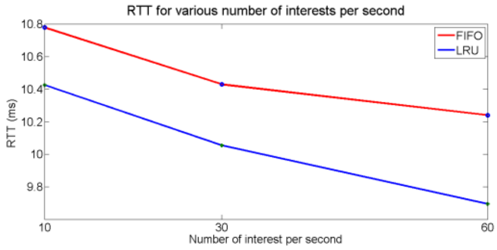


Figure 10. RTT for various interest rate

3.3 Scenarios of various number of consumers

The increasing number of consumers will also increase the CHR. LRU provides 6-12 % bigger CHR than FIFO. The bigger the consumers number, the higher the total number of content requests in the network. The demand for the same content also increases. The router can serve the repeatedly requested content and result in a higher CHR value.

RTT LRU 1-3% smaller than FIFO. This value does not change significantly, only about 1% for the double number of consumers. An increase in the number of consumers does not significantly affect system performance from the RTT side if the number of content populations on the network does not change.

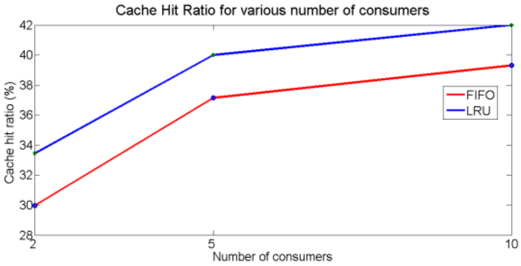


Figure 11. CHR for various number of consumers

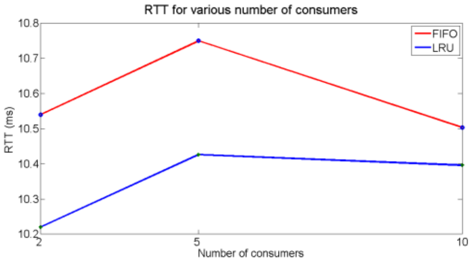


Figure 12. RTT for various number of consumers

3.4 Scenario of various number of producers

The increase of producers number means the increasing number of content populations in the system. Each producer has 100 pieces of content. Consumer demand for content corresponds to a normal distribution spread across all producers. The more consumers, the more dispersed and varied content demand, so CS needs to store more content. With the number of producers up to 5, CS size 30 still provides a larger CHR. However, when the producer goes up to 10, which means there are 1000 contents population in the network, the CHR decreases. RTT increased by about 3 ms for a 2-fold increase in consumers.

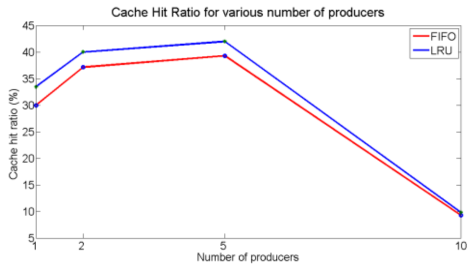


Figure 13. CHR for various number of producers

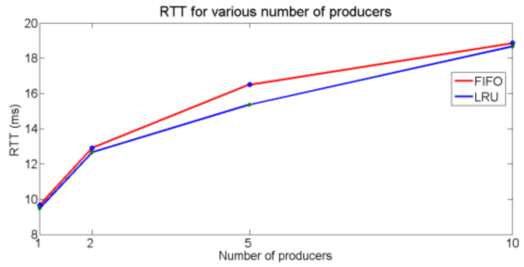


Figure 14. RTT for various number of producers

3.5 Scenario of various number of prefixes for each producer

This scenario uses the various number of prefixes for each producer. More prefixes mean more content in the system. We set the fixed number of consumers and the number of requests. The scenario in subsection 4.4 (scenario of the various producers) represents that content owners are more dispersed when producers increase. The scenario in subsection 4.5 modeled that only one producer owns all content.

The testing result shows that the higher the number of prefixes, the lower the CHR. Due to the fixed CS size, more varied content requests cause CS to keep more content, so there will be a miss-content if it is unavailable at the router. The request is then forwarded to the producer node. LRU still provides greater performance than FIFO, around a maximum of 10%. A decrease in CHR is also followed by an increase in RTT.

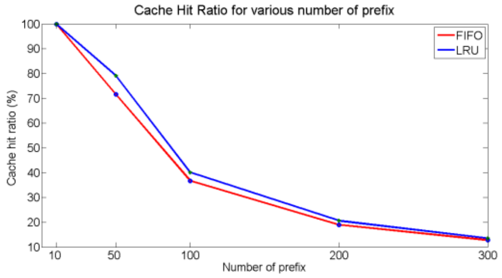


Figure 15. CHR for various number of prefixes

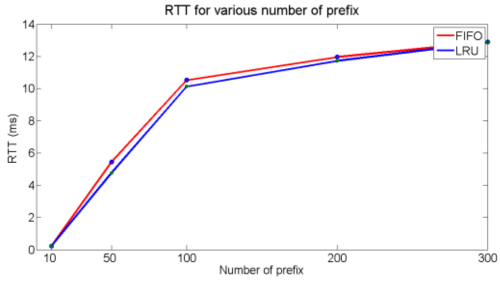


Figure 16. RTT for various number of prefixes

4. CONCLUSION

We have modeled the more realistic NDN system in this research and evaluated the system's performance using two cache replacement algorithms with various possible scenarios. The CS size affects the CHR. The bigger the CS size, the bigger the CHR. The larger the size of the CS, the less the RTT. LRU is superior to FIFO in all cases. The larger the size of the CS, the greater the difference in CHR between FIFO and LRU, as well as the RTT. The more the number of interests, the greater the CHR. However, in the case of the highest number of interests, CHR decreases because the size of CS 30 cannot accommodate too many variations in consumer demand. Using the concept of storing content as requested with CS size limitation, FIFO is not as good as LRU in managing content stored in CS. The bigger the CHR, the smaller the RTT. If the number of content population is not changing, the growth of consumers affects the increase of CHR, while the RTT does not significantly change. In the case of changes in the number of producers, the more producers, the more dispersed and varied demand from consumers, so CS needs to store more content. The higher the number of prefixes, the lower the CHR. Due to the fixed CS size, more varied requests for various content causes CS to be unable to accommodate user requests, so there will be miss content. Based on our tests, LRU provides better performance than FIFO in CHR and RTT parameter. For the further research, we will develop an NDN system using single board computer for each NDN node and bring the NDN ready to be implemented.

ACKNOWLEDGEMENTS

This research was supported by the Telkom University.

5. REFERENCES

[1] A. Afanasyev, J. Burke, L. Wang, and B. Zhang, “A Brief Introduction to Named Data Networking,” 2018.
[2] I. Networking, M. Zhang, H. Luo, and H. Zhang, “A Survey of Caching Mechanisms in Information

centric Networking," *IEEE Commun. Surv. TUTORIALS*, vol. 17, no. 3, pp. 1473–1499, 2015.

[3] H. Jin, D. Xu, C. Zhao, and D. Liang, "Information-centric mobile caching network frameworks and caching optimization: a survey," *Eurasip J. Wirel. Commun. Netw.*, vol. 2017, no. 1, pp. 1–32, 2017, doi: 10.1186/s13638-017-0806-6.

[4] L. V. Yovita and N. R. Syambas, "Caching on named data network: A survey and future research," *Int. J. Electr. Comput. Eng.*, vol. 8, no. 6, pp. 4456–4466, 2018, doi: 10.11591/ijece.v8i6.pp.4456-4466.

[5] L. V. Yovita, N. R. Syambas, I. Joseph, M. Edward, and N. Kamiyama, "Performance Analysis of Cache Based on Popularity and Class in Named Data Network," in *Future Internet*, 2020, vol. 12, no. 227, doi: https://doi.org/10.3390/fi12120227.

[6] J. H. Mun and H. Lim, "Cache sharing using bloom filters in named data networking," *J. Netw. Comput. Appl.*, vol. 90, pp. 74–82, 2017, doi: 10.1016/j.jnca.2017.04.011.

[7] L. V. Yovita, N. R. Syambas, I. Joseph, and M. Edward, "Weighted-CAPIC Caching Algorithm for Priority Traffic in Named Data Network," *Futur. Internet*, vol. 14, no. 3, pp. 1–15, 2022, doi: 10.3390/fi14030084.

[8] V. Sourlas, "Partition-based Caching in Information-Centric Networks," in *Seventh IEEE International Workshop on Network Science for Communication Networks (NetSciCom 2015)*, 2015, pp. 396–401.

[9] S. Podlipnig and L. Böszörmenyi, "A survey of Web cache replacement strategies," *ACM Comput. Surv.*, vol. 35, no. 4, pp. 374–398, 2003, doi: 10.1145/954339.954341.

[10] S. Shailendra, S. Sengottuvelan, H. K. Rath, B. Panigrahi, and A. Simha, "Performance evaluation of caching policies in NDN-an ICN architecture," in *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, 2017, pp. 1117–1121, doi: 10.1109/TENCON.2016.7848182.

[11] H. Situmorang, N. R. Syambas, and T. Juhana, "The effect of scaling the size of Topology and Content Stored on the Named Data Networking," in *Proceeding of 2016 10th International Conference on Telecommunication Systems Services and Applications, TSSA 2016: Special Issue in Radar Technology*, 2017, pp. 16–21, doi: 10.1109/TSSA.2016.7871110.

[12] F. C. Alamsyah, L. V. Yovita, and R. M. Negara, "The Effect of Content Population and Frequency Interest for Named Data Networking with Modified-Optimal Replacement Algorithm," 2021.

[13] G. Carofiglio, V. Gehlen, and D. Perino, "Experimental evaluation of memory management in content-centric networking," in *IEEE International Conference on Communications*, 2011, pp. 1–6, doi: 10.1109/icc.2011.5962739.

[14] X. Hu, J. Gong, G. Cheng, and C. Fan, "Enhancing in-network caching by coupling cache placement, replacement and location," in *IEEE International Conference on Communications*, 2015, vol. 2015-Septe, pp. 5672–5678, doi: 10.1109/ICC.2015.7249226.

[15] E. Hattab and S. Qawasmeh, "A Survey of Replacement Policies for Mobile Web Caching," in *Proceedings - 2015 International Conference on Developments in eSystems Engineering, DeSE 2015*, 2016, pp. 41–46, doi: 10.1109/DeSE.2015.13.

[16] D. Meint and S. Liebald, "From FIFO to Predictive Cache Replacement," 2019, doi: 10.2313/NET-2019-06-1.

[17] M. Dehghan, L. Massoulie, D. Towsley, D. S. Menasche, and Y. C. Tay, "A Utility Optimization Approach to Network Cache Design," in *IEEE/ACM Transactions on Networking*, 2016, vol. 27, no. 3, pp. 1013–1027, doi: 10.1109/TNET.2019.2913677.

[18] C. M. I. N. Park and R. A. Rehman, "Packet Flooding Mitigation in CCN-based Wireless Multimedia Sensor Networks for Smart Cities Packet Flooding Mitigation in CCN-Based Wireless Multimedia Sensor Networks for Smart Cities," *IEEE Access*, vol. 5, no. June, pp. 11054–11062, 2017, doi: 10.1109/ACCESS.2017.2715407.

[19] L. V. Yovita and N. R. Syambas, "Content Storage Effect on the Named Data Network Traffic Load," 2017.

[20] Mini-NDN_Team, "Mini-NDN," 2022. https://minindn.memphis.edu/ (accessed Mar. 15, 2022).