2023 International Conference on Computer Science, Information Technology and Engineering (ICCoSITE) | 979-8-3503-2095-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCoSITE57641.2023.10127742

Impact of Data Freshness-aware in Cache Replacement Policy for NDN-based IoT Network

Kania Pradnya Sekardefi
Adaptive Network Laboratory,
School of Electrical Engineering,
Telkom University, Indonesia
kaniapradnyas@student.telkomuniversity.ac.id

Ridha Muldina Negara
Telecommunication Engineering,
School of Electrical Engineering,
Telkom University, Indonesia
ridhanegara@telkomuniversity.ac.id

***Abstract*— The world has entered the development of the digital era, where all information can be obtained easily through internet services. The number of requests that are frequently requested makes internet network traffic only able to accommodate some user requests. A named data network (NDN) is here as a solution to overcome this problem. NDN changed the focus of the internet architecture, which was initially host-centric, to become content-centric. Caching on NDN router nodes can be used as a repository for passing content. Because IoT data always requires fresh data and real-time, one of the features in NDN called freshness can help maintain data freshness in the NDN router cache. This paper explores implementing the freshness method for content replacement decisions in the two cache replacement policies. Cache replacement policies are Least Recently Used (LRU) and First-in, first-out (FIFO). To validate the effectiveness of adding freshness-aware in the caching model, we run the emulation using an NDN emulator, Mini-NDN. The results show that freshness can maintain the freshness of data in IoT data and the performance of NDN caching with LRU policy increases based on parameters of the cache hit ratio and RTT compared to the FIFO policy.**

Keywords— Data Freshness, in-network caching, cache replacement policy, Mini-NDN, Named Data Networking, Internet of Things (IoT).

I. INTRODUCTION

The world has entered the development of the digital era, where all information can be obtained easily through internet services. The large number of requests from users who access needs to IoT has increased data services used by the internet network [1]. Due to the large number of users who retrieve content in requests made repeatedly, internet traffic has become congested. Because many of the exact requests and responses must be made repeatedly, it causes a high increase in bandwidth on the network [2], [4].

The current network architecture focuses on host-centric IP; the work process could be more efficient, so it is experiencing difficulties. To solve this problem, IP-based networks were modified so that a new paradigm emerged called Content-Centric Network (CCN), currently better known as Named Data Network (NDN). Named Data Network (NDN) transforms the previous internet network, which focused on host-centric, into content-centric. So later, when the user requests, the package will be sent directly based on the desired content rather than where the content is. One of the essential features in the Named Data Network is called caching. The advantage of caching is that the NDN router node will store content that passes [3], [5], [6]. The caching feature can speed up the content distribution process on the network. The caching strategy used in this paper is the Cache Replacement Policy [4]. Currently, internet network traffic requires real-time and is temporary, and data freshness is considered essential for implementing IoT data on NDN [7]. In addition, IoT data is usually short-lived and is often updated once in a while [8].

Therefore, one of the features of NDN that can be utilized is freshness. Freshness allows packet data to be stored in the content store (CS) so that the freshness of data stored on the router does not need to be questioned. However, unfortunately, more needs to be discussed about the effect of the Cache Replacement Policy on the freshness of content on the Named Data Network. Like paper [9] which only discusses research on NDN freshness in IoT conducted on LCE, LCD, ProbCache, Btw, and edge-caching cache strategies, and only uses LRU as Cache Replacement. Paper [10] also discusses freshness in IOT at CCN, which does not use a Cache Replacement Policy. Paper [11] conducted a comparative study of a cache hit ratio, traffic, and delay using Cache Replacement Policies such as LRU, LFU, and FIFO but using an NDN-Sim emulator and not implementing a freshness strategy. Paper [12] only discusses freshness as a caching metric on an internet scale.

This paper will explore the influence and analysis of the deployment of the Caching Replacement Policy Least Recently Used (LRU) and First In First Out (FIFO) on the freshness period on the Named Data Network using the mini-ndn emulator, regarding various previous papers. Hopefully, this paper can answer questions about improving performance in serving traffic on the network.

II. RELATED WORK

A. Named Data Network (NDN)

In today's internet development, the hourglass internet architecture on universal networks still uses IP addresses. IP is made for communication networks to send request packets requested by consumers and addressed to specific addresses. NDN proposes the evolution of changes to the IP architecture.

Figure 1 compares hourglass internet architectures that still use IP addresses after being evolved by NDN into pieces of content. More precisely, NDN changes network services initially intended for certain addresses to become specific content. The content mentioned is the package name [13].

Figure 2 shows the NDN architecture; there are two types of packets: interest packets and data packets [14]. These two packets have their respective functions and characteristics. They aim to identify a piece of data that will be transmitted in one data packet. The flow is that the consumer lists the desired name to the packet interest, and then the interest will be sent to the network. The router will use the name given by the consumer to forward the interest to the producer, who has the desired data. After the packet interest arrives at the node with the requested data, the node will respond by sending a packet containing the name, content, and signature to lock these two things [15].

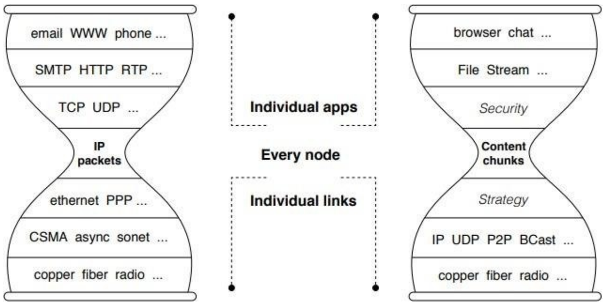

Fig 1. Comparison of Internet Architectures Using IP and Using Content on NDN [13]

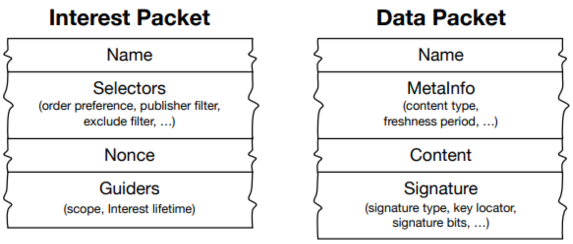

Fig 2. NDN Packets [13]

B. Caching Pada Named Data Network

Many new paradigms have emerged regarding internet development, but the Named Data Network (NDN) is one of the most prominent information centers in network instances [8]. Communication on the network is determined by two packets, namely data packets and interest packets [13]. Caching on NDN is reactive and only done on the base package. The caching commonly used on NDN is Cache Everything Everywhere (CEE) which means that all content will be copied and placed on all nodes on the network [8].

NDN caching has two strategies: cache placement and cache replacement [4]. This paper discusses a caching strategy called Cache Replacement Policy. Cache replacement determines the rules on the Content Store relating to which content will be stored or deleted. There are 2 Cache Replacement Policies used in this study, namely:

a. First In First Out (FIFO) → The default policy cache on the NDN. FIFO performs evictions on each inserted packet.

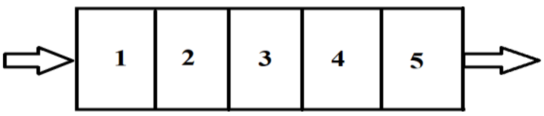

Fig 3. The illustration of incoming packets

From the illustration in figure 3, if a new package arrives, for example, a package named "2", then the package named "5" will automatically be removed from the caching system.

b. Least Recently Used (LRU) → The LRU work system is almost the same as FIFO. However, if it is like the scheme used in the FIFO explanation, the package named "2" will move to place number "1" because package two is requested again by the consumer.

C. Freshness Period

In the Named Data Network (NDN), there are called Interest Packets and Data Packets. Freshness Period is one of the NDN Packet Format Specifications in the Data Packet. Freshness Period is optional and can be set on the NDN-Traffic Generator Server. Freshness regulates how long a node must keep "fresh" content so that it must mark the content as "non-fresh" content.

Freshness is specified by the source and indicates the age of the data in seconds. When the content time runs out, the data packet will be considered stale, and the source will generate a new data packet [7]. Content freshness is timed in CS to determine when data is considered expired or invalid so it can be deleted. The user can determine the freshness period and choose which interest they want to use [16].

Freshness can also indirectly become a cache replacement policy on NDN because it deletes fresh data no longer [8]. Therefore paper [12] proposes a new algorithm called Least Fresh First (LFF) as a cache replacement policy in NDN.

D. NDN For The Internet of Things

The rapid development of IoT networks is currently a big challenge for hardware ecosystems with heterogeneous resources and traffic patterns. As the range of network architectures connects large ecosystems, there is a massive demand for functionality and frameworks and service configuration capabilities, management, security, reliability, resiliency, and scalability. NDN can directly manage all these functions at the network layer level [14], [15].

E. Emulator Mini-NDN

Mini-NDN is a network emulator that can perform tests, experiments, and research on the NDN platform. Mini-NDN was initially based on Mini-CCNx, which was a fork of mininet. Mini-NDN uses several libraries such as NDN, NFD, NLSR, and other tools released by NDN researchers to emulate NDN networks on a system [17]. Mini-NDN is an open-source project that everyone can use. Members of the NSF-sponsored NDN project team developed the first release of Mini-NDN. Because Mini-NDN is a software developed by Mininet, Mini-NDN has some similarities with Mininet. The relationship between Mini-NDN and mininet can be seen in figure 4.

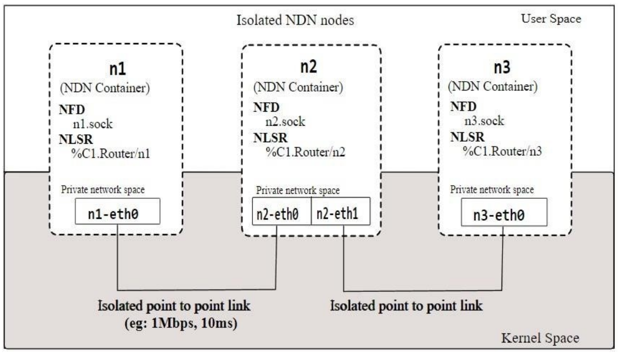

Fig 4. The relationship between Mini-NDN and Mininet [17], [18]

III. METHODOLOGY

This research was conducted to prove the effectiveness of using freshness on the Cache Replacement Policy on the Named Data Network. The tools that can be used to carry out a freshness test scenario on Mini-NDN are the NDN-Traffic Generator. NDN Traffic Generator is one of the tools designed to generate traffic interest packets and packet data on NDN networks. The client and server will receive a traffic configuration file that can be used to determine the required NDN traffic pattern. If the NDN Traffic Generator installation process is already done, many configuration files will be provided. These files include sample how-to files for configuring the parameters provided on the NDN.

In addition, the NDN Traffic Generator makes it easy to process data with lots of traffic simultaneously. The amount of freshness can be set on the NDN-Traffic-Generator server, and on the NDN-Traffic-Generator Client section, the user is free to choose whether he wants fresh data.

Figure 5 shows the topology used in the simulation process. The Banten node will store all the content that the user requests. So, if other city nodes do not have the content the user wants, the request will proceed to the source node.

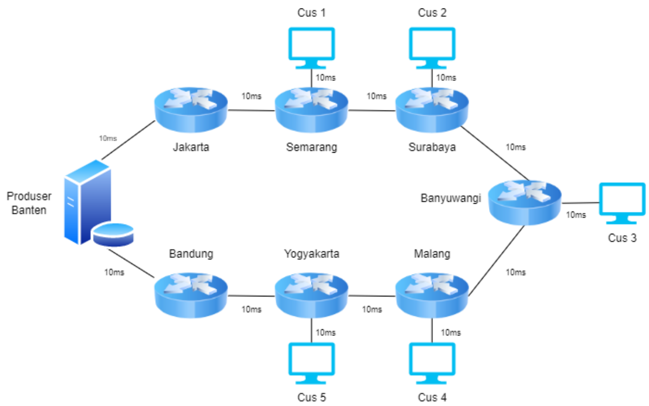

Fig 5. The Topology

Table I. Hardware Specifications

Hardware	Specification
Processor	AMD Ryzen 5 3500U with Radeo Vega Mobile Gfx (8 CPUs), 2.1GHz
RAM	8 GB
Harddisk	40 GB

Table II. Scenario

Parameters	Values
Topology	Fig. 5
Distribution Content	Uniform
Must Be Fresh	0, 1
Freshness Period	200 ms, 2.000 ms, 20.000 ms
Cache Replacement Policy	LRU, FIFO
Content Store Size	Default Mini-NDN
Delay	10ms
Number of Content	17
Content Requested	5 per user
Number of Running	5

Table I is the required hardware specifications, and Table II is the scenario used in this paper's simulation. The producer will provide 17 pieces of content, and each user will request five pieces of data. Figure 6 is the distribution of content requested by each user. Where between adjacent users, there will be slices of the same content.

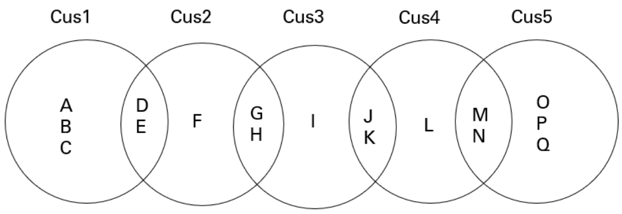

Fig 6. Content Requests for Each User

Requests submitted by each user will be carried out simultaneously. The first test is when freshness is active, and the freshness is set from 200 ms to 20.000 ms. Then the second scenario is an experiment when freshness is not active.

IV. RESULT

The results of the simulations that have been carried out with several different scenarios are as follows:

A. Round Trip Time (RTT)

Figure 7 and Figure 8 are the RTT results from the freshness simulation on Cache Replacement Policy in Mini-NDN. MBF=1 means freshness active. The customer wants fresh content, and MBF=0 means freshness not active so the customer does not want fresh content. In the experiment using the topology in figure 5, seven nodes will distribute content from producers to 5 customers.

From the graphs of figures 7 and 8, it can be seen that when freshness active, the RTT value when the freshness setting is 200 ms reaches its highest point and begins to decrease when the freshness is 20.000 ms. Freshness 200 ms has a minuscule period compared to freshness 2.000 ms and 20.000 ms. So, when a customer requests fresh content, the CS node will check whether each content is still fresh. Because the freshness time is short, the content is already unfresh. This condition means the request must be forwarded to the producer node, which makes the RTT even bigger. It is different when the freshness time is set to be bigger, like 2.000 ms and 20.000 ms. Because the data is still fresh, the request does not need to be forwarded to the producer node. Customers can directly retrieve content from the nearest node. This condition makes the RTT for 2.000 ms and 20.000 ms freshness much smaller than for 200 ms freshness.

When freshness is not active, when freshness is 200 ms, 2.000 ms, and 20.000 ms, there is no significant difference in the RTT. That is because the customer does not ask for fresh data. This condition means that customers can retrieve content from the nearest node. Because the freshness period is not active, the RTT will be the same no matter how much freshness is set at the producer node.

The shape of the graph resembles a triangle for customer 1 to customer five because customer 3 is the farthest from the producer. Therefore, the highest RTT for each freshness is at customer 3 when the freshness level is negligible.

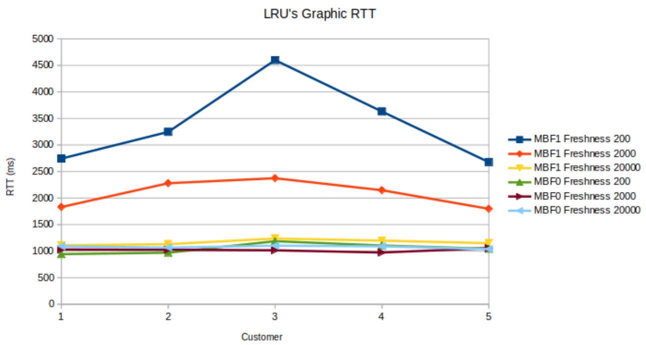

Fig 7. RTT results on the influence of freshness when active and not on LRU

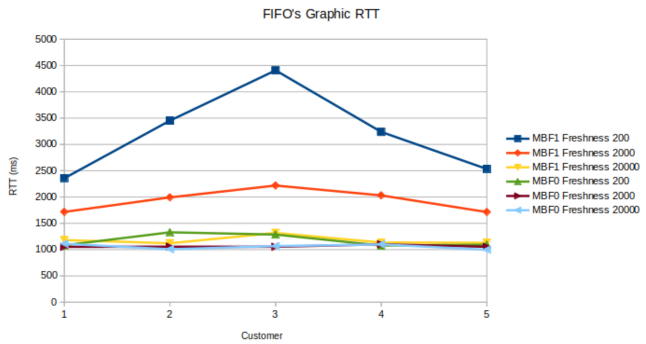

Fig 8. RTT results on the influence of freshness when active and not on FIFO

B. Cache Hit Ratio

Figure 9 and figure 10 are the results of simulated images when freshness is active. When freshness is active in the 200 ms timeframe, the Jakarta and Bandung nodes will receive the lowest CHR because these two nodes do not have customers asking for requests directly, unlike the other nodes. Of the two cache replacement scenarios tested, the amount of CHR at the Jakarta and Bandung nodes was only between 0-2%. Other nodes have a much higher CHR than the Jakarta and Bandung nodes because these nodes have each customer close together. However, the Banyuwangi node is not included because the distance is very far from the producer, so when the freshness level is trimmed, nMiss has more data than the data successfully sent.

Then when the freshness level is increased to 2.000 ms, the CHR results will be better than when the freshness is 200 ms because the period is much longer. The Jakarta and Bandung nodes are the same because they do not have direct customer requests. However, in Figure 9 and Figure 10, when the freshness is at 200 ms, the CHR value is 2-7%, and at 2.000 ms, freshness increases to 12-17%. Why can this happen? Because requests for fresh content do not need to be made multiple times to producers, they can retrieve the data directly from the node that stores it. So less nMiss will be generated so that the CHR results will be better. It also affects when the freshness value is raised again to 20.000 ms. Because the freshness time is longer than before, the CHR value is almost the same when freshness is turned off.

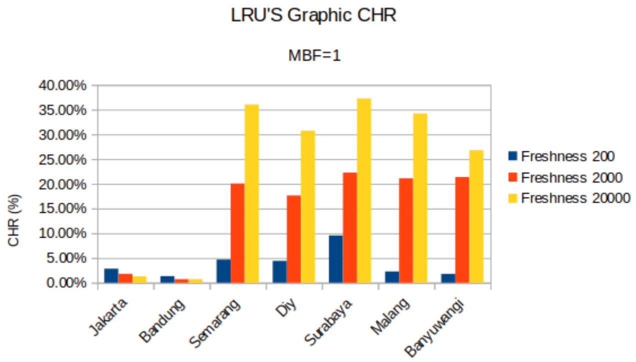

Fig 9. CHR results on the influence of freshness when MBF=1 on LRU

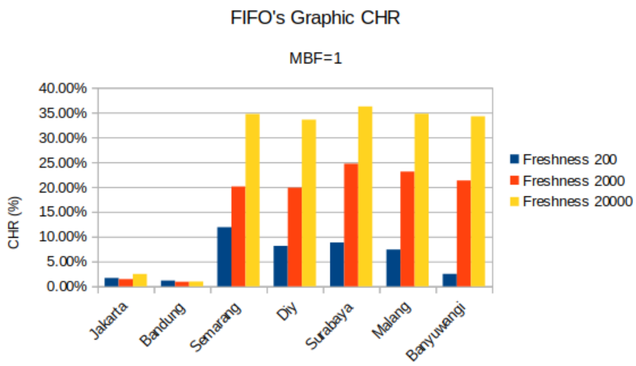

Fig 10. CHR results on the influence of freshness when MBF=1 on FIFO

Figure 11 and figure 12 show the CHR results when freshness is turned off in both LRU and FIFO. The graphics are much different from the previous ones. If we look again at the Semarang, Yogyakarta, Surabaya, Malang, and Banyuwangi nodes, there is no increase in CHR at each significant change of freshness. This condition can happen because if freshness is turned off, much data will be successfully cached and sent so that the nMiss is much less.

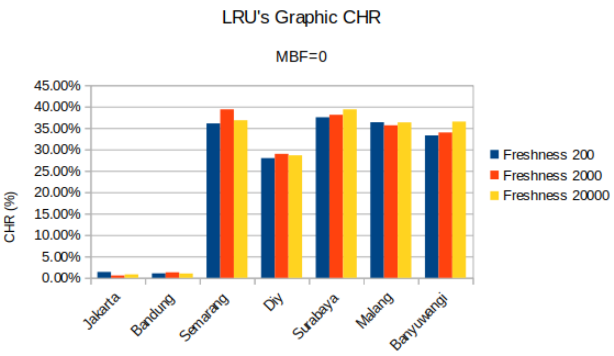

Fig 11. CHR results on the influence of freshness when MBF=0 on LRU

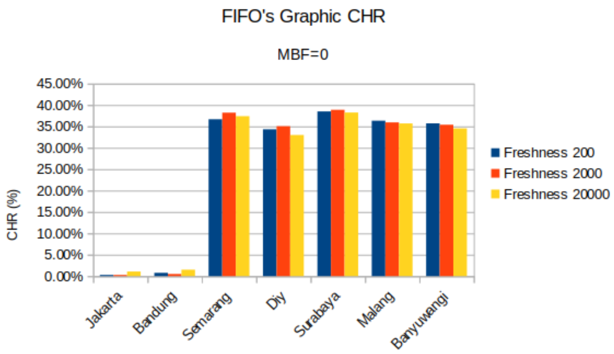

Fig 12. CHR results on the influence of freshness when MBF=0 on FIFO

V. CONCLUSION

Based on the research results from the simulations that have been tested, freshness can be applied to IoT data which requires that the data is always updated and the Cache Replacement Policy LRU and FIFO influence the decision to replace content on the Named Data Network. The lower the RTT and the higher the CHR when the freshness is 20.000 ms, it means that the longer the freshness is, the higher the chance for data to be cached. In addition, if the freshness time is set to long, i.e., 20.000 ms or more, the freshness performance is almost the same when freshness is not active because it gives a long time for one data content to be stored in the NDN router node. The shorter the freshness time is set, the stored data will always be updated but the resulting RTT and CHR values are not good. For further research, the freshness method will be applied in the Indonesian topology using LRU, FIFO, and Least Frequently Used (LFU) policies. The addition of LFU policy aims to determine the impact and influence of content popularity on Named Data Network.

REFERENCES

[1] M. Yu, R. Li, Y. Liu, and Y. Li, “A caching strategy based on content popularity and router level for NDN,” *Proc. 2017 IEEE 7th Int. Conf. Electron. Inf. Emerg. Commun. ICEIEC 2017*, pp. 195–198, 2017, doi: 10.1109/ICEIEC.2017.8076542.

[2] R. M. Negara and N. Rachmana Syambas, “Caching and machine learning integration methods on named data network: A survey,” 2020, doi: 10.1109/TSSA51342.2020.9310811.

[3] L. V. Yovita, N. R. Syambas, and I. Y. Matheus Edward, “CAPIC: Cache based on Popularity and Class in Named Data Network,” in *Proceedings - 2018 International Conference on Control, Electronics, Renewable Energy and Communications, ICCEREC 2018*, 2018, pp. 24–29, doi: 10.1109/ICCEREC.2018.8712105.

[4] M. P. Pamungkas, S. A. Ekawibowo, and N. R. Syambas, “Priority Based Multilevel Cache LRU on Named Data Network,” in *Proceeding of 2019 5th International Conference on Wireless and Telematics, ICWT 2019*, 2019, pp. 28–31, doi: 10.1109/ICWT47785.2019.8978270.

[5] N. R. Syambas *et al.*, “Research Progress On Name Data Networking To Achieve A Superior National Product In Indonesia,” in *8th International Conference on Wireless and Telematics (ICWT)*, 2022, pp. 1–6, doi: 10.1109/icwt55831.2022.9935364.

[6] F. Zharfan, L. D. Hasnaa, R. Muldina Negara, and N. R. Syambas, “Comparison of Caching Replacement Policies in Changing the Number of Interest Packets on Named Data Networks Using Mininet-NDN,” 2021, doi: 10.1109/TSSA52866.2021.9768267.

[7] S. Alduayji, A. Belghith, A. Gazdar, and S. Al-Ahmadi, “PF-ClusterCache: Popularity and Freshness-Aware Collaborative Cache Clustering for Named Data Networking of Things,” *Appl. Sci.*, vol. 12, no. 13, 2022, doi: 10.3390/app12136706.

[8] M. Amadeo, C. Campolo, G. Ruggeri, and A. Molinaro, “Beyond Edge Caching: Freshness and Popularity Aware IoT Data Caching via NDN at Internet-Scale,” *IEEE Trans. Green Commun. Netw.*, vol. 6, no. 1, pp. 352–364, 2022, doi: 10.1109/TGCN.2021.3124452.

[9] M. Meddeb, A. Dhraief, A. Belghith, T. Monteil, K. Drira, and S. Alahmadi, “Cache freshness in named data networking for the internet of things,” *Comput. J.*, vol. 61, no. 10, pp. 1496–1511, 2018, doi: 10.1093/comjnl/bxy005.

[10] E. T. da Silva, J. M. H. de Macedo, and A. L. D. Costa, “NDN Content Store and Caching Policies: Performance Evaluation,” *Computers*, vol. 11, no. 3, pp. 1–17, 2022, doi: 10.3390/computers11030037.

[11] 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, p. 4456, 2018, doi: 10.11591/ijece.v8i6.pp4456-4466.

[12] M. Meddeb, A. Dhraief, A. Belghith, T. Monteil, K. Drira, and H. Mathkour, “Least fresh first cache replacement policy for NDN-based IoT networks,” *Pervasive Mob. Comput.*, vol. 52, pp. 60–70, 2019, doi: 10.1016/j.pmcj.2018.12.002.

[13] L. Zhang *et al.*, “Named data networking,” *Comput. Commun. Rev.*, vol. 44, no. 3, pp. 66–73, 2014, doi: 10.1145/2656877.2656887.

[14] A. Aboodi, T. C. Wan, and G. C. Sodhy, “Survey on the Incorporation of NDN/CCN in IoT,” *IEEE Access*, vol. 7, pp. 71827–71858, 2019, doi: 10.1109/ACCESS.2019.2919534.

[15] M. Amadeo, C. Campolo, A. Iera, and A. Molinaro, “Named data networking for IoT: An architectural perspective,” in *EuCNC 2014 - European Conference on Networks and Communications*, 2014, no. June, doi: 10.1109/EuCNC.2014.6882665.

[16] J. Quevedo, D. Corujo, and R. Aguiar, “Consumer driven information freshness approach for content centric networking,” in *Proceedings - IEEE INFOCOM*, 2014, pp. 482–487, doi: 10.1109/INFCOMW.2014.6849279.

[17] “Mini-NDN.” [Online]. Available: https://minindn.memphis.edu/introduction.html#what-is-mini-ndn. [Accessed: 17- Mei-2022].

[18] A. Gawande and L. Wang, “Need for Mini-NDN.” .