Based on Content Relevance Caching Strategy in Information-Centric Network

Xueqin Xiong[1](✉), Zhanjun Liu[1], Yuan Zhang[2], and Qianbin Chen[1]

[1] School of Communication and Information Engineering,
Chongqing University of Posts and Telecommunications, Chongqing, China
1710382196@qq.com
[2] School of Computer and Internet of Things,
Chongqing Institute of Engineering, Chongqing 400056, China

Abstract. Information-Centric Networking (ICN) replaces identifying endpoints with identifying content. One of the most common and important features of ICN architectures is in-network caching, which can significantly reduce content request latency and improve user quality of service(QoS) and network performance. Therefore, how to efficiently utilize cache resources and optimize cache performance has become one of the research hotspots in ICN. This paper proposes a local popularity caching strategy based on content relevance caching (CRC). In this strategy, the local popularity of the content is calculated based on the relevance of the content requested. Routing nodes make caching decisions based on local popularity. In the forwarding process of the interest packet, the cache node ID table in the interest packet is updated according to the local popularity, and the cache decision is made. In the backhaul process of the data packet, content caching and replacement strategy need to be performed according to the caching parameters and local popularity parameters. In cache replacement, if the remaining cache capacity of the current routing node cannot satisfy the required capacity of the cached content, the local popularity of the content needs to be used to replace the cached content. In this way, repeatedly requested content ends up being cached in routing nodes closer to the user. The simulation results show that CRC has better network performance than several other classical caching strategy.

Keywords: Information-Center Networking · Content Relevance · Local Popularity · Caching Strategy

1 Introduction

With the development of information technology, the traffic in the network continues to increase. The number of users who obtain information from the network

---

Q. Chen—This work was supported by the Chongqing Municipal Education Commission under Grant KJQN201900645.
Chongqing University of Posts and Telecommunications Doctoral Start-up Fund Project (A2019017).

F. Gao et al. (Eds.): ChinaCom 2022, LNICST 500, pp. 122–134, 2023.
https://doi.org/10.1007/978-3-031-34790-0_10

is gradually increasing, and users are more concerned about the content itself [1]. This has led to the Information-Centric Networking (ICN) [2] architecture as the research direction for future networks. ICN emerged as an architecture that includes caching at the edge of the network. Unlike traditional methods, users in ICN request content directly from the network only by their Named Data Object (NDO) names [3]. As shown in Fig. 1, ICN includes cache management modules in all routing nodes to realize the decoupling of content and address [4], and each routing node in the network can selectively store a copy of the content [5]. When the same content request occurs again, the corresponding content can be returned directly from the routing node without the need to obtain it from the source server. However, the cache capacity of each routing node is limited, and only reasonable cache deployment [6,7] can improve the cache utilization of nodes.

The current caching strategies in ICN include: Leave Copy Everywhere (LCE) [8], Leave Copy Down (LCD) [9], Probability [10], ProbCache [11] etc. LCE is the routing node caches every piece of content it passes through. It does not consider the influence of content popularity and network topology, which will lead to frequent replacement of cached content in routing nodes, resulting in low cache utilization of routing nodes. LCD is next-hop routing node that caches content at the hit routing node, which helps to gradually push popular content caches to edge routing nodes. It considers the network topology, however, popular content needs to move many times to reach the edge routing node, so the convergence speed is slow. Probability is that the routing node caches the arriving content with a fixed probability. It without considering the distance between the user and the routing node. Therefore, the QoS for users cannot be improved. ProbCache is the probability that content can be cached to a routing node inversely proportional to the distance between the routing node and the content requester. It enables content to be quickly cached to routing nodes closer to the user. The research found that in practical applications, there is a relevance between the content requested. However, some existing caching mechanisms [12] do not consider the relevance between the content requested.

Aiming at some existing caching mechanisms, this paper proposes a caching decision based on content relevance caching (CRC). Considering the relevance between the contents requested, the local popularity of the content is calculated according to the relevance between the contents [13]. Therefore, according to the local popularity of the content, the routing node decides whether to cache the content requested. During the forwarding of interest packet, make caching decisions based on local popularity and update the table of cache node IDs in interest packet. In the process of backhauling data packet, content caching and replacement strategy need to be performed according to cache parameters and local popularity parameters. In cache replacement, if the remaining cache capacity of the current routing node cannot satisfy the required capacity of the cached content, the local popularity of the content needs to be used to replace the cached content. This paper considers the user's QoS from the content request delay in the network [14,15], and the content copy is cached in the node closer

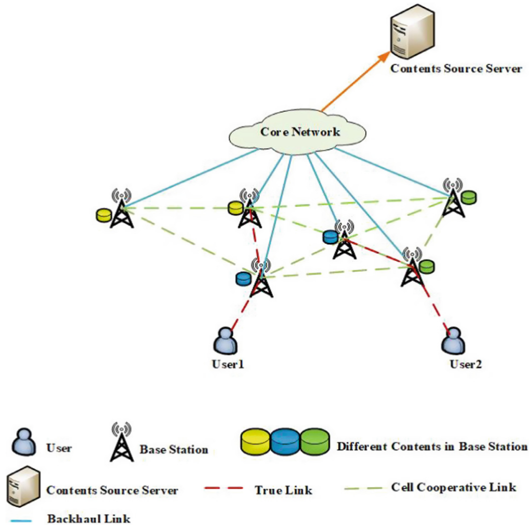


Fig. 1. Cache model in ICN.

to the user side, so as to realize the network performance and improve the user's QoS [16].

In this paper, it is assumed that all routers in the network have the same computing power and the same cache capacity. Furthermore, to simplify the model, this paper considers everything contents in the network to have the same capacity. Considering the content service model of the network, it is assumed that the requests sent by users satisfy the Poisson distribution, and the popularity of the content satisfies the Zipf-like distribution [17].

2 Cache Model and Modified Structure

2.1 Cache Model

In order to simultaneously satisfy the user's QoS and improve the cache utilization, the optimal cache location needs to be adopted when content caching. The optimization problem model is derived for the content cache location problem as follows.

$$\begin{aligned} &\min \sum_{c\in C}\sum_{n\in N} d_n^c\left(1-S_n^c\right) \\ &\text{s. t}\begin{cases} \sum\limits_{c\in C} S_n^c < C_n & \forall n\in N \\ S_n^c=\{0,1\} & \forall n\in N, \forall c\in C \end{cases} \end{aligned} \tag{1}$$

where d_n^c represents the distance from the content request initiated by node n to the routing node that provides content c. The S_n^c represents whether node n

caches content c, the value is 0 or 1. The content cached by each node cannot exceed the cache capacity of the routing node itself. The caching strategy proposed in this paper is a caching strategy based on content relevance, and the caching decision is made in the forwarding stage of the interest packet. In order to make the content relevance-based caching strategy proposed in this paper feasible, it is necessary to add the cache routing node ID table field to the interest packet and data packet, and add the cache parameter S_n^c and the popularity parameter *pop* to the routing node.

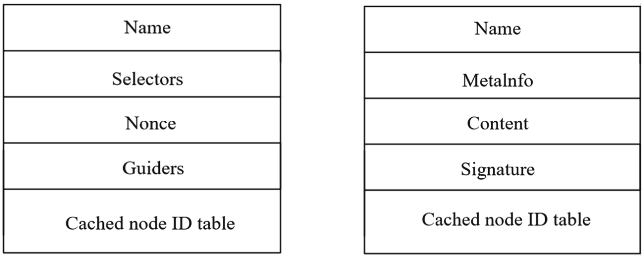
When the request interest packet sent by the user arrives at the routing node, the cache parameter S_n^c in the routing node needs to be queried first. If $S_n^c = 1$, it means that the content has been cached in the routing node, and the data packet is returned directly. If $S_n^c = 0$, it means that the routing node does not cache the content, and the local cache cannot satisfy the user's request. Continue to forward the interest packet upwards. In the forwarding process, a cache decision needs to be made in order to update the cache ID table in the interest packet. The caching decision proposed in this paper is a caching decision based on a greedy algorithm. If the remaining cache space of the current routing node is sufficient, the arriving content can be directly cached. At this time, the ID of the routing node needs to be added to the cache node ID table in the interest packet, the cache parameter S_n^c corresponding to the cache information table in the routing node is set to 1, and the interest packet continues to be forwarded upward. If the remaining space of the current routing node is insufficient, it is necessary to determine whether to add the ID of the current routing node to the interest packet after updating the local popularity. Assuming that $\mathrm{ID} = n1, n2...$ represents the list of nodes on the forwarding path of the content request that have decided to cache the data packet, there are:

$$ID^c = \begin{cases} ID^c \cup \{n\} & pop_n^c > \min\left(\left\{ \text{ pop } _n^i \mid i \in C_n \right\}\right) \\ ID^c & \text{otherwise} \end{cases} \tag{2}$$

where C_n represents the set of all content that have been cached in routing node n. The i represents the least popular content cached in the current routing node.The pop_n^i represents the content that has been cached in routing node the minimum popularity of the current routing node. If the local popularity c of the currently requested content is greater than the minimum local popularity among the cached content in the current routing node. Then the current routing ID of the node needs to be added to the node ID table of the cached interest packet. Set the cache parameter S_n^c corresponding to the cache information table in the routing node to 1, and the cache parameter of the modified content i to 0. Continue to forward interest packet up.

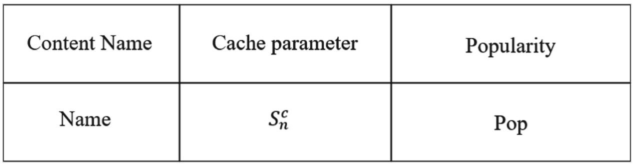
2.2 Modified Interest Packet, Data Packet and Routing Node Models

The modified the fromat of interest packet is shown in Fig. 2(a), the fromat of data packet is shown in Fig. 2(b), and the structure of the routing node is shown in Fig. 3.



(a) interest packet (b) data packet

Fig. 2. Modified packet of interest and data



Content Name	Cache parameter	Popularity
Name	S_n^c	Pop

Fig. 3. Structure of the modified routing node.

3 Local Popularity Caching Mechanism Based on Content Relevance

3.1 Calculation of Content Relevance

In practical applications, each user has its own access domain, which means that there is a high relevance between the content requested by the user, so the content with high relevance in the routing node is more likely to be accessed by the user in turn. In this paper, the proposed caching strategy exploits the semantic relevance of content as the relevance between content. The first step in semantic relevance analysis is to extract attributes that have an impact on relevance, and the more attributes with the same or similar semantics, the greater the relevance. When calculating the semantic relevance, this paper mainly considers the content name of the interest packet, and calculates the content relevance according to the content name attribute of the interest packet.

Relevance coefficient parameter: $\beta_{c'}^{c}$ represents the relevance between the content c and the content of the node cache c', $\beta_{c'}^{c}$ value of is 0, 1.

$$\beta_{c'}^{c} = \frac{|M \cap M'|}{|\max(\mathrm{M}, \mathrm{M}')|} \tag{3}$$

In this equation,M and M' represent the semantic vector spaces of content c and content c', respectively. Where $M = M_1, M_2......M_k$ denotes the set of name strings of content c and $M' = M'_1, M'_2......M'_m$ denotes the set of name strings of content c'. The number of elements in the intersection between the semantic vectors M and M' is used as the numerator. The maximum number of ideas of M and M' (i.e., the maximum number between k and m) is used as the denominator.According to the above formula, the relevance between the content c and the content c' is obtained.

3.2 Local Popularity Calculation

Since the content requested by the user is high relevance, data content with high relevance is more likely to be requested by the same user. Taking relevance into account when calculating local popularity can improve network performance. The local popularity of content c at routing node n is updated as follows.

$$pop_n^c = \frac{Ret_n^c}{\sum_{c_j \in C_n} Ret_n^{c_j}} + \sum_{c' \in C_n} \beta_{c'}^c \tag{4}$$

where Ret_n^c indicates the number of requests for content c in routing node n, and C_n indicates all content cached by routing node n. The $\sum_{c_j \in C_n} \mathrm{Ret}_n^{c_j}$ represents the total number of requests for all content at routing node n. The $\beta_{c'}^c$ indicates the relevance between content c and content c'. The $\sum_{c' \in C_n} \beta_{c'}^c$ indicates the sum of the relevances between the content c and all content already cached in the routing node.

3.3 Caching Decision Process

The caching decision proposed in this paper is made in each routing node along the forwarding path of interest packet sent by users. The decision-making process algorithm is as follows.

The algorithm proposed in this paper is based on the greedy algorithm. When the routing node receives the interest packet requested by the user, it firstly queries whether the remaining available cache capacity of the current routing node can satisfy the cache capacity for caching the content requested by the user. If possible, directly add the ID of the routing node to the cache node ID table of the interest packet, calculate the popularity of the interest packet in the current routing node, and update the cache node ID table interest packet and the cache information table in the routing node. If not, it is necessary to query the minimum popular content and its popularity in the cache information table of the routing node, and compare the popularity of the arriving content with the minimum popularity of the cached content in the node. If it is greater than, add the ID of the routing node to the cache node ID of the interest packet table, and update the cache node ID table interest packet and the cache information table in the routing node, and then forward the interest packet. The cache parameter of the replaced Interest in the routing node is set to 0. If the popularity of the content is less than the minimum popularity of the content in the routing node, the interest packet is forwarded directly.

Algorithm 1. Caching decisions based on content relevance

```
Input: interest packet for request content $c$ received by node $n$
        $pop_n^c = 0$
Output: $S_n^c$
if $CacheAvailablecapacity > CacheContentCapacity$ then
    $S_n^c = 1$
    Update the current local popularity of content $c$ in the routing node $pop_n^c$
    $ID^c \cup \{n\}$
    $CacheAvailablecapacity = CacheAvailablecapacity - CacheContentCapacity$
else
    Calculate the current local popularity of content $c$ in the routing node $pop_n^c$
    if $pop_n^c > pop_n^i$ then
        $S_n^c = 1$
        $ID^c \cup \{n\}$
        $S_n^i = 0$
    else
        $S_n^c = 0$
        $ID^c = ID^c$
    end if
end if
```

4 Simulation and Performance Analysis

In order to verify the performance of the caching strategy proposed in this paper in the network. This paper uses the ndnSIM2.7 simulation tool to compare the caching strategy with several other existing caching strategies. And verify the effectiveness of the strategy through simulation processing. MATLAB is used for the result data.

4.1 Experimental Environment and Parameter Settings

Simulation is performed in Linux (ubuntu 18) environment using NS3-based ndnSIM2.7 [18] emulator. The main modified parts are:

(1) The format of interest packet is modified, and the cached message ID table is added to the interest packet.
(2) The format of data packet is modified and the cached message ID table is added to the data packet.
(3) Modify the structure of the routing node and add the cache message ID table in the routing node.

The cache strategy proposed in this paper is simulated and implemented, and compared with other cache strategies. The experimental simulation parameters are shown in Table 1.

The network structure used in this paper is a tree structure, divided into 7 layers, each internal node has 2 child nodes, and there are 127 nodes in total, including 64 leaf nodes, 62 intermediate nodes, and 1 root node. In the simulation process, the leaf node is set as the user request node, the intermediate node is

Table 1. Experimental parameter setting.

Experimental parameters	Experimental setup
Frequency of content requests per node	100
Cache capacity per node	50–150(100)
Content Capacity	1
Content Popularity Parameter (Zipf)	0.6–1.4(1)
Simulation time	10 s

set as the routing node that needs to make caching decisions, and the root node is set as the content source node. It is assumed that the content source node can provide all the content requested by users in the network, and the cache required to cache each content is 1 unit. The content popularity obeys Zipf distribution, the parameter of Zipf is set between 0.6 and 1.4, the default value of Zipf parameter is 1. The cache capacity of each node is set to 50–150 units, and the default value of node cache capacity is 100 units.

4.2 Experimental Performance Indicators

In order to be able to effectively compare the results of the caching decision proposed in this paper with several other caching decisions, several common network performance in the network are selected as evaluation metrics.

(1) Cache hit ratio: cache hit ratio is the ratio of the number of content requests satisfied by a routing node to the total number of content requests made by all content requesters over a period of time. cache hit ratio can be used to measure the cache performance metrics in the network, and the cache utilization of a routing node can be reflected by the cache hit ratio. The higher the value of cache hit ratio, the higher the cache utilization of the routing node. Suppose the number of cache hits of a routing node is R_{hit}, the total number of requests for all contents sent by a requester is R_{total}, and the cache hit ratio is $cache_{hit}$, then there is a cache hit ratio of:

$$Cache_{hit} = \frac{R_{hit}}{R_{total}} \tag{5}$$

Figure 4 reflects the variation of cache hit rate with Zipf parameter when the cache capacity is the default value, from the figure, It can be seen that in different cache decision strategies, with the increase of the Zipf parameter, the cache rate increases, but the cache hit rate in CRC is better than the cache hit rate in other caching strategies, which is because when the cache capacity of the routing node is certain, the cache hit rate in the same routing This is because when the cache capacity of a routing node is certain, users are more likely to request content with high relevance in the same routing node, and the relevance of users' requested content is taken into account when calculating the content popularity in CRC, which can improve the cache hit ratio. Moreover, the local popularity calculation method is introduced in the CRC strategy proposed in this paper, which can more accurately sense the dynamic changes of the popularity of interest packet in the routing nodes, so the cache hit rate changes more obviously. From the figure, It can be seen that when the Zipf parameter is equal to 1, the cache hit rate of the routing node increases significantly, and the cache hit rate at this time is better than several other cache strategies.

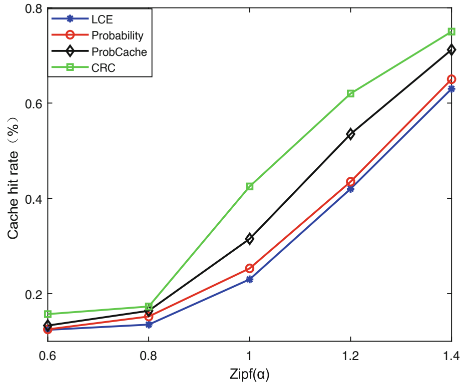


Fig. 4. Variation of cache hit rate with Zipf parameters.

Figure 5 shows the variation of the cache hit rate with the change in the cache capacity of the routing node when the Zipf parameter is set to its default value. From the figure, it can be seen that the cache hit rate of the routing node increases with the increase of the cache capacity of the routing node. The higher cache hit ratio of CRC compared to several other caching policies is due to the real-time update of the popularity of the cached content in the CRC strategy, which makes the content cached in the routing node reach the real-time update with the popularity of the content.

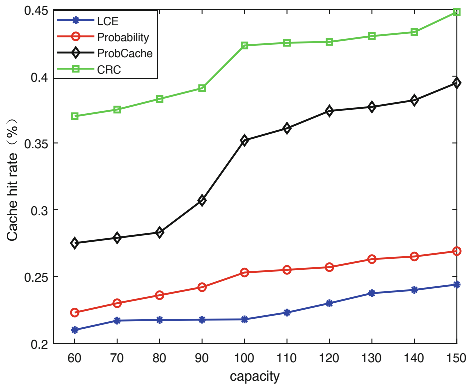


Fig. 5. Variation of cache hit rate with cache capacity.

(2) Average delay: The average delay is the average time for a user to send a content request and receive a content response. The average delay can reflect the retrieval efficiency of the content in the network and the QoS of the user. The lower the average delay, the higher the retrieval efficiency., the better the QoS of the user.

Figure 6 shows the average delay as a function of Zipf parameters. It can be seen that as the Zipf parameter increases, the average latency decreases across several caching decisions. When the Zipf parameter is 1, the caching decision proposed in this paper is quite different from other caching decisions. In this experiment, the delay of one-hop routing node is set to 10 ms. From the simulation results, it can be seen that the average delay is about 35 ms, indicating that most of the user's requests are responded between two hops, which improves the user's QoS.

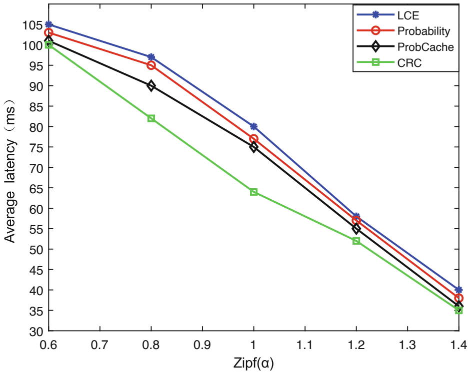


Fig. 6. Variation of average access latency with Zipf parameters.

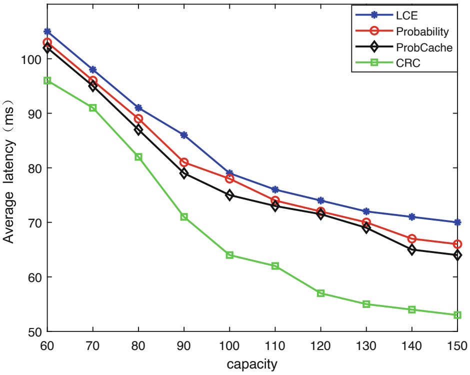


Fig. 7. Variation of average access latency with cache capacity.

Figure 7 shows the average delay with the cache capacity is recorded, indicating that the average delay decreases with the increase of the cache capacity. Compared with other caching strategies, the CRC average delay reduction is more obvious. Because the CRC considers the relevance between the content requested by the user, the content copy is cached in the routing node closer to the user, which can respond to the user's request faster.

(3) Hop reduction rate: Assuming that $hop_r(t)$ is the number of hops elapsed in a single request and $Hop_r(t)$ is the number of hops elapsed by the corresponding user of this request to obtain content from the content-providing source, the hop reduction rate is $H(t)$:

$$\mathrm{H(t)} = \frac{\sum_{r=1} \mathrm{Hop}_r(t) - \sum_{r=1} \mathrm{hop}_r(t)}{\sum_{r=1} \mathrm{Hop}_r(t)} \tag{6}$$

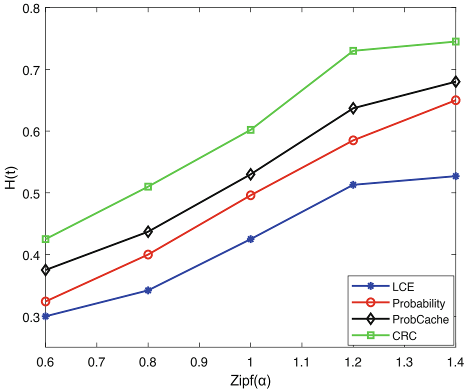


Fig. 8. Variation of hop count reduction rate with Zipf parameters.

Figure 8 shows the hop reduction rate as a function of the zipf parameter. When α is small, the popularity of the content is not concentrated, and the residence time of the cached content in the node is small. At this time, the cached content in the routing node is frequently replaced, making it difficult for the content of subsequent requests to be responded by the routing node, and the effect of shortening the content acquisition path is not obvious, and the reduction rate of hops is low. As α increases, requests for content are more concentrated, and at this time, the replacement of cached content in routing nodes is reduced. Because the caching strategy proposed in this paper makes full use of the local characteristics of content popularity, the performance can be better than several other caching strategies.

Figure 9 shows the variation of hop count reduction rate with the cache capacity of the routing node. When the cache capacity of the routing node is small, most of the content requests need to be forwarded to the content-providing source server to get a response, and the hop count reduction rate is lower at this time. The performance improvement of CRC is more stable compared to the other caching strategies.

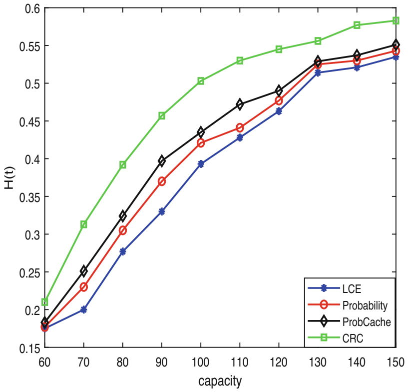


Fig. 9. Variation of hop count reduction rate with cache capacity.

5 Conclusion

In order to effectively utilize the cache space of routing nodes in ICN and improve the service quality of users, a cache strategy based on CRC is proposed. Based on the local popularity of the content, the ultimate goal is to minimize the forwarding distance of content sent by users. The simulation results show that CRC can cache content in routing nodes closer to the user, obtain a higher cache hit rate, reduce the user's request delay, and improve the user's QoS and network performance.The next step will be based on the research of this paper, combined with the diversity of user requests in the network, to analyze some key problems existing in the current caching strategy. Combined with the relationship between routing nodes, the cache strategy is studied to further optimize the cache performance of the network.

References

1. Li, J., Xue, K., Liu, J., Zhang, Y.: An ICN/SDN-based network architecture and efficient content retrieval for future satellite-terrestrial integrated networks. IEEE Netw. **34**(1), 188–195 (2019)
2. Ngaffo, A.N., El Ayeb, W.: Information-centric networking challenges and opportunities in service discovery: a survey. In: 2020 IEEE Eighth International Conference on Communications and Networking (ComNet), pp. 1–8. IEEE (2020)
3. Lederer, et al.: Adaptive multimedia streaming in information-centric networks. IEEE Netw. Mag. Comput. Commun. **28**(6), 91–96 (2014)

4. Nour, B., et al.: A unified hybrid information-centric naming scheme for IoT applications. Comput. Commun. **150**, 103–114 (2020)
5. Qian., H.U., et al.: Random cache placement strategy for content-centric networking. In: Journal of Xidian University (2014)
6. Din, I.U., et al.: Caching in information-centric networking: strategies, challenges, and future research directions. IEEE Commun. Surv. Tutorials **20**(2), 1443–1474 (2017)
7. Hagikura, J., Nakamura, R., Ohsaki, H.: On the optimal cache allocation in information-centric networking. In: 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC) IEEE (2020)
8. Jacobson, V., et al.: Networking named content. Springer International Publishing (2015)
9. Laoutaris, N., Che, H., Stavrakakis, I.: The LCD interconnection of LRU caches and its analysis. Perform. Eval. **63**(7), 609–634 (2006)
10. Wu, H., et al.: Design and evaluation of probabilistic caching in information-centric networking. IEEE Access **6**, 32754–32768 (2018)
11. Naeem., M.A., Ullah, R., Meng, Y., et al.: Caching content on the network layer: a performance analysis of caching schemes in ICN-based Internet of Things. IEEE Internet of Things J. **9**, 6477–6495 (2022)
12. Zhang, G., Yang, L., Tao, L.: Caching in information centric networking: a survey. Comput. Netw. **57**(10), 3128–3141 (2013)
13. Kang, M.W., et al.: A content caching optimization scheme based on content popularity in ICN. In: Proceedings of Symposium of the Korean Institute of communications and Information Sciences (2018)
14. Qian, H.U., et al.: Random cache placement strategy for content-centric networking. In: Journal of Xidian University (2014)
15. Jiang, Y., et al.: User preference learning-based edge caching for fog radio access network. IEEE Trans. Commun. **67**(2), 1268–1283 (2019)
16. Ibrahim, et al.: Survey on caching approaches in information centric networking. J. Netw. Comput. Appl. **56**, 48–59 (2015)
17. Yang, et al.: Optimal caching and scheduling for cache-enabled D2D communications. IEEE Commun. Lett. Publ. IEEE Commun. Soc. **21**, 1155–1158 (2017)
18. Afanasyev, A., Ilya, M., Zhang, L.: ndnSIM: NDN simulator for NS- 3. In: Tech. Rep. 4, (2012)