A Low-Latency Content Acquirement Scheme in NDN

Dapeng Qu, Zhaoyang Heng, Guoxin Lv and Xingwei Wang

October 24, 2022

A low-latency content acquirement scheme in NDN

Dapeng Qu[1], Zhaoyang Heng[1], Guoxin Lv[1], Xingwei Wang[2*]
[1]College of Information, Liaoning University, Shenyang 110036, China
[2]School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China
dapengqu@lnu.edu.cn, {hengzhaoy123, lnu_lvguoxin}@163.com, wangxw@mail.neu.edu.cn

***Abstract*—As a new network architecture, the content-centric model of Named Data Networking (NDN) is gradually replacing the host-centric model of the current Internet. Meanwhile, with the rapid development of Metaverse, the low-latency content acquirement attracts much attention. Thus, we propose LCAS, a Low-latency Content Acquirement Scheme in NDN. In LCAS, an ant routing algorithm is proposed to take full advantage of caching in NDN to strengthen routing paths. Specially, a Data packet lays pheromone based on the caching information of upstream nodes along the routing path from the content provider to the content requester, which can effectively guide the subsequent Interest packets. On the other side, a neighbor discovery algorithm is presented to help acquire content from the one-hop neighbors as far as possible. Extensive experiments on DFN and Deltacom have been conducted to show its performances, and the results demonstrate that LCAS has lower average hop counts and average end-to-end delay, compared with other schemes.**

***Index Terms*—Named data networking, low latency, content acquirement, ant routing, neighbor discovery**

I. Introduction

With the rapid development of Internet, network application model is evolving from early resource sharing to current content requirement. However, current Internet infrastructure is mainly based on IP address indicating end-to-end connection. Therefore, it is difficult for traditional end-to-end transmission mode to satisfy this change [1]. To solve this problem, Named Data Networking (NDN) is proposed. It changes the semantics of network service from delivering packets to a given destination to fetching content identified by a given name, thus helps the Internet evolving from the host-centric model to a content-oriented model [2].

In NDN, a content requester sends out an Interest packet carrying a name identifying its desired content. The Interest packet is routed towards the content producer based on this name. Any node which holds the matching content or the content producer will act as a content provider and provides a corresponding Data packet which carries the matching content, and the Data packet is forwarded back to the content requester through the corresponding Interest packets' routing path reversely. Therefore, how to acquire a desired content from the content providers is an important issue in NDN.

Recently, the idea of Metaverse attracts numerous attention. Network architecture is the first challenge presented by Metaverse to cyberspace, and the content-oriented model of NDN is more appropriate for Metaverse than the host-centric model of current Internet. Moreover, the immersive experience of Metaverse brings a very high requirement on latency, while the in-networking caching of NDN enables intermediate nodes to cache the content to be content providers, shortens the routing paths to the content requester and reduces content acquirement latency [3]. However, considering the importance of latency on various future Internet applications, thus, how to acquire content with a low latency from the appropriate content provider is still an important issue in NDN.

Although there are some existing works which propose content acquirement or routing/forwarding strategies for NDN, they try their best to reduce the content acquirement latency [4]. There are still two important shortcomings need to be considered. On one hand, in-networking caching is an important issue in NDN, and taking advantages of caching will greatly reduce latency than just getting content from the original content producers. Thus, how to know and get the content from content providers is crucial to reduce content acquirement latency. On the other hand, physical proximity is an important issue in Internet, and getting content from physical neighbors will have a natural advantage in latency than that from distant nodes. Thus, how to know and get the content of neighbors is also crucial to reduce content acquirement latency.

To solve the problem aforementioned, we propose LCAS, a **L**ow-latency **C**ontent **A**cquirement **S**cheme in **NDN**. LCAS consists of **ARA**(an **A**nt **R**outing **A**lgorithm) and **NDA**(a **N**eighbor **D**iscovery **A**lgorithm). ARA strengthens the good routing paths by Data packets laying pheromone based on the caching information of upstream nodes along the routing path from the content provider to the content requester, and NDA learns the caching information of one-hop neighbors to help acquire content. Both of them try to reduce the content acquirement latency by fully taking advantage of caching information of the neighbors and content providers. The main contributions of this work are summarized as follows.

- We propose a low-latency content acquirement scheme for NDN. It consists of two disparate algorithms, namely ARA and NDA. ARA takes full advantage of in-networking caching and tries to get the content from some content providers which hold the content replica. While NDA utilizes the physical proximity and tries to acquire content from one-hop neighbors. Both of them contribute to reduce the latency for getting content from nearer nodes, not the original content producers.
- We conduct extensive experiments to evaluate the performance. Experimental results show that LCAS has lower average hop counts and end-to-end delay. For example, LCAS can reduce the delay by up to 78.46% when compared with other schemes.

This work is supported by the National Key Research and Development Program of China under Grant No. 2019YFB1802800, the National Natural Science Foundation of China under Grant Nos. 62032013 and 61872073, and Natural Science Basic Research Fund of Liaoning Provincial Education Department under Grant No. LJC202002.

The rest of this paper is organized as follows. Section II summarizes related works. Section III presents the system model, the working flow, two corresponded algorithms and algorithm pseudocodes in LCAS. Section IV evaluates the performance with extensive experiments under two different network scenarios. Finally, Section V concludes the paper.

II. Related Work

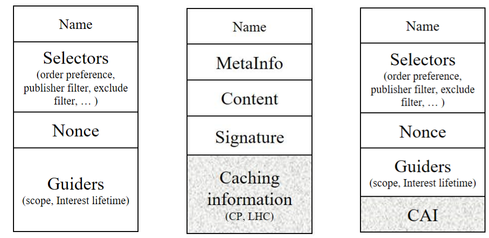
Forwarding is an important issue in networking, and an effective and efficient forwarding will help the content requester to get its required content as quickly as possible [5]. As the main performance metric, delay attracts much attention and plenty of researchers have tried their best to reduce content acquirement latency from current Internet to future ICN. In NDN, the Interest packet sent from the content requester will be forwarded by any received intermediate nodes based on the Pending Interest Table (PIT) and Forwarding Information Base (FIB), if there is no corresponding content in its Content Store (CS). Among them, CS is a content cache, PIT records all the interfaces (indexed by content names), for which it has forwarded Interest packets that have not been satisfied yet, and these future corresponding Data packets will be sent based on all the interfaces. FIB is a routing table which maps content names to interfaces for Interest routing.

As the most distinct feature of NDN, caching attracts much attentions, and plenty of researchers focus on caching to reduce the content acquirement latency [6]. However, the current NDN routing protocols still focus on the traditional problem of forwarding Interest packets to content producers, without explicit support of in-network caching, which limits NDN's potential and benefits to applications [7]. Moreover, by labeling each route update at the entry point into a network, internal routers select the same border router for the same name prefix, which enhances the hit ratio of cached contents. Fetching the Nearest Replica (FNR) [8] is a CDN-like enhancement to the NDN design. When a consumer sends an Interest packet for a popular content which will be fetched from the nearest replica in the network, regardless of whether it is on the best path from the content producer to the content requester. DIVER [9] explores the network to search and retrieve content that is closer to the content requesters. It uses probe packets to find content, and the routers answer the received probe packets by inserting their availability information in a space-efficient data structure. It keeps the most diversified as possible the acquired availability information in the explorations to raise the probability of satisfying future content requests.

Although there are plenty of related works, how to reduce content acquirement latency is always a crucial issue in computer networks, it will be more and more important with the rapid development of Metaverse.

III. System Design

In this section, we present LCAS's design in detail. We first introduce the model and the flow of our system respectively. Then we propose a content acquirement scheme LCAS based on ant routing algorithm **ARA** and neighbor discovery algorithm **NDA**. Finally, we provide algorithm pseudocodes for these algorithms respectively.



(a) Interest Packet (b) Data Packet (c) Hello Packet

Fig. 1. The structure of packets in NDN.

A. System Model

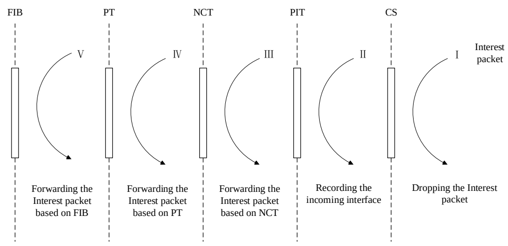
In this paper, NDN can be modeled as a connected graph $G = (V; E)$, where $V = \{v_i|1 \leq i \leq N\}$ is the set of nodes, and $E = \{e_{i,j}|v_i, v_j \in V, 1 \leq i; j \leq N, i \neq j\}$ is the set of links. N is the total number of nodes in G, and $N = |V|$. At the same time, the content can be modeled as a set C, and a content file $c_k \in C$, where $1 \leq k \leq M$, $M = |C|$ is the total number of content files in NDN.

To support the content acquirement scheme proposed in this paper, referring to the basic packet formats in NDN, we redesign the structure of a Data packet shown in Fig. 1(b). Moreover, a Hello packet to achieve neighbor discovery is also proposed. The field with shadow is the new added ones. In a Data packet, the "Caching information" field means the caching information of the content in this Data packet, "LHC" will be true if the content was cached in the last hop node, otherwise, it will be false. "CP" means the pheromone value of this content taken by the Data packet, which can be used to guide the subsequent Interest packets. Obviously, the larger the value, the higher probability of getting required content through corresponding interfaces. In a Hello packet, the "CAI" field means the changing of CS in current node and is used to let one-hop neighbors know the caching information more accurately. The detailed calculation and usage process can be referred in the later subsection.

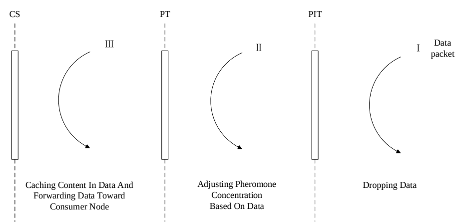
To support the newly proposed ARA and NDA in LCAS, referring to the basic node structure, we redesign node structure for NDN. That is, two new tables, namely Pheromone Table (PT) and Neighbor Caching Table (NCT) are designed to realize the ARA and NDA respectively. PT is the pheromone table held by a node and consists of three items, namely Content prefix, Interface and Pheromone. The pheromone value $\tau^{K}_{intf_j}$ reflects that the probability of the current node getting the required content prefix $/K$ from some nodes holding it through interface $intf_j$. NCT is the neighbor caching information table and consists of two items, namely Content name and Interface.

B. Working Flow

According to the basic working flow of NDN and the above analysis based on the proposed network model and the structures of packets, the working flows of receiving an



(a) Receiving an Interest packet



(b) Receiving a Data packet

Fig. 2. Working flow of process a packet

Interest packet, a Data packet and a Hello packet are designed respectively. As shown in Fig. 2(a), when receiving an Interest packet, the NCT and PT are added between PIT and FIB, that is, if there is no corresponding entry in PIT, the node will resort to NCT and PT to acquire the content through one-hop neighbors and some nearer content providers. They contribute to a high probability of getting the required content with a low latency. The reason of NCT being before PT is that one-hop neighbors usually have much nearer distance than those content providers, which will further help reduce latency. If the three schemes fail, the node will finally rely on FIB.

Similarly, as shown in Fig. 2(b), when receiving a Data packet, the new PT is added between PIT and CS, that is, if there is no corresponding entry in PIT, the current node will update PT and then take caching strategy in CS. The detailed description can be referred to Section. III-C. For the Hello packet, the CS will be retrieved to generate and the NCT will be updated to reflect the information of neighbors. The detailed description can be referred to Section. III-D.

C. Ant Routing Algorithm (ARA)

As a swarm intelligence algorithm, ant colony optimization has been successfully applied to design routing algorithms. A Data packet lays pheromone based on the caching information of upstream nodes along the routing path from the content provider to the content requester, which can effectively guide the subsequent Interest packets. The pheromone not only reflects the quality of routing path, but also reflects the caching information, and the routing path with more pheromone will have a higher probability to acquire the required content. Recall the Data packet in Fig. 1(b)

$$CP_{new} = \begin{cases} \lambda \times CP_{old} + \Delta CP & LHC\ is\ true \\ \lambda \times CP_{old} & otherwise \end{cases} \tag{1}$$

where CP_{old} and CP_{new} are the values in "CP" field in the Data packet before and after update, λ is an coefficient, and $0 \leq \lambda \leq 1$, since the farther the content provider, the less its attraction. ΔCP is the value strengthened by the last neighbor for its caching. After that, the Data packet updates its "LHC" fields based on the caching strategy. If it is cached in the current node, it will be set as True, otherwise, it will be set as False, which can tell the caching information to the next node. Then the node will update its PT table as Eq. (2):

$$\tau_{intfK}^{K} = \tau_{intfk}^{K} + \Delta\tau \tag{2}$$

where $\Delta\tau$ is the value strengthened by the received Data packet, and it can be set as CP_{new}. Moreover, to reflect the change of network and caching information, the pheromone value can be evaporated periodically as Eq. (3):

$$\tau_{intfK}^{K} = \rho \times \tau_{intfk}^{K} \tag{3}$$

where ρ is the evaporation coefficient, and $0 \leq \rho \leq 1$. The evaporation operation will be taken at the beginning of each time slot, which can remove the influence of old routing information, and new strengthened routing paths will have a higher probability to be used to guide later Interest packets.

D. Neighbor Discovery Algorithm (NDA)

Because of the naturally physical proximity, one-hop neighbors will play a huge roll in content acquirement if they hold the required ones. Recall the Hello packet in Fig. 1(c), a node generates the "CAI" field based on the change of CS. It can be a binary mapping table, and denoted as $<$ Content name, Action $>$, where content name just denotes these content which changes recently, and action is the corresponding action, namely caching and deletion. When a node receives a Hello packet, it will update NCT structure based on the "CAI" field.

E. Algorithm Pseudocode

Since LCAS consists the interactions between Interest packets and Data packets, we will provide two algorithms about the process of receiving them respectively. Due to the limited space and the simplicity, the processing of Hello packet will just be described and the algorithm will not be provided.

For convenience, it is assumed that the number of contents in CS is N_{CS}, the number of entries in PIT is N_{PIT}, the number of entries in NCT is N_{NCT}, the number of entries in PT is N_{PT}, and the number of entries of in FIB is N_{FIB}.

1) Processing of an Interest packet

As the important way of content acquirement, how to process a received Interest packet plays a crucial role in reducing the latency. Algorithm 1 shows the pseudocode of the processing an Interest packet.

Recall Fig. 3(b), the intermediate node takes operation in CS(lines 1-2), PIT(lines 4-8), NCT(lines 9-10), PT(lines 11-13) and FIB(lines 14-15) orderly. NCT and PT are the new operations which do not exist in traditional NDN. They contribute a better way to get the required content when there is no corresponding entry in PIT and before taking FIB. Furthermore, the entry finding operations are taking exact matching, thus the time complexity is linear operation, based on the above analysis, the total complexity of Algorithm 1 is $\mathbf{O}(Alg1) = \mathbf{O}(N_{CS} + N_{PIT} + N_{NCT} + N_{PT} + N_{FIB})$.

Algorithm 1 Receiving process of an Interest packet

```
Input: Received Interest packet
Output: Forwarding/Dropping Interest packet
    Generating and Sending Data packet
if the current node holds the required content then
    Generate a corresponding Data packet with the content and
    Send it to the interface of receiving the Interest packet;
else
    if there is a corresponding entry in PIT then
        Record a new interface for this entry;
        Drop the Interest packet;
    else
        Insert an entry with content name and corresponding
        incoming interface into PIT;
        if there is a corresponding entry in NCT then
            Forward the Interest packet based on NCT;
        else
            if there is a corresponding entry in PT then
                Forward the Interest packet based on PT;
            else
                Forward the Interest packet based on FIB;
            end if
        end if
    end if
end if
```

2) Processing of a Data packet

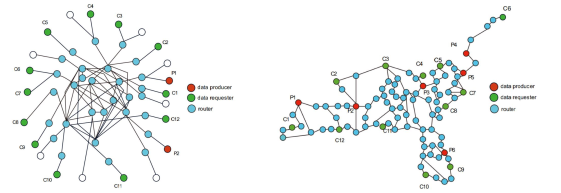
Because the required content is encapsulated in a Data packet and the important in-networking caching of NDN, how to process a received Data packet is important to the subsequent content requirement. Algorithm 2 shows the pseudocode of the processing a Data packet. Recall Fig. 3 (b), the intermediate node takes operation in PIT(lines 1-2), PT(line 4) and CS(line 5) orderly. PT is also the new operations which do not exist in traditional NDN. Moreover, without loss of generality, we just take the LCE as the caching strategy for its originality and simplicity. It relies the pheromone in the routing path just as the ant does, which can guide the subsequent Interest packets to get the required content wtih a low latency in an effective way. Similarly, the total complexity of Algorithm 2 is $\mathbf{O}(Alg2) = \mathbf{O}(N_{PIT} + N_{PT})$.

Algorithm 2 Receiving process of a Data packet

```
Input: Received Data packet
Output: Forwarding/Dropping Data packet
1: if there is not any corresponding entry in PIT then;
2:     Drop the Data packet;
3: else
4:     Update PT based on Eq. (2);
5:     Cache the Data packet with LCE;
6:     Forward the Data packet based on the entry in PIT;
7: end if
```

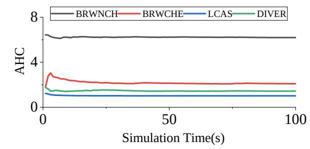
3) Processing of a Hello packet

As the new added communication way, Hello packets will help nodes know the caching information of one-hop neighbors, and the naturally physical proximity makes an important contribution to reduce content acquirement latency. The process is as follows. A node generates a Hello packet with the "CAI" field which reflects the change of CS, and sends it to all interfaces. On the other side, when a node receives a Hello packet, it gets the change of caching information in one-hop neighbors through "CAI" field in the Hello packet and updates the NCT based on the change of caching information in its one-hop neighbors. Thus, the complexities of sending process of a Hello packet and receiving process of a Hello packet are $\mathbf{O}(N_{CS})$ and $\mathbf{O}(N_{PT})$ respectively.

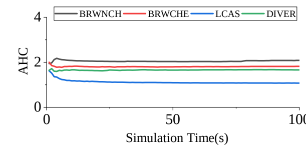


(a) DFN (b) Deltacom

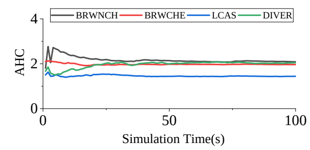
Fig. 3. Network topology



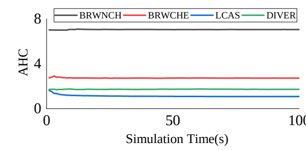
(a) C1 in DFN.



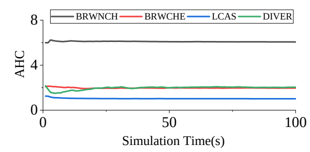
(b) C5 in DFN.



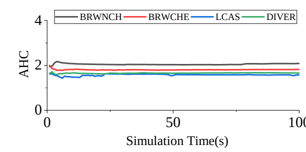
(c) C7 in DFN.



(d) C3 in Deltacom.



(e) C6 in Deltacom.



(f) C12 in Deltacom.

Fig. 4. Performance comparisons of Average Hop Counts.

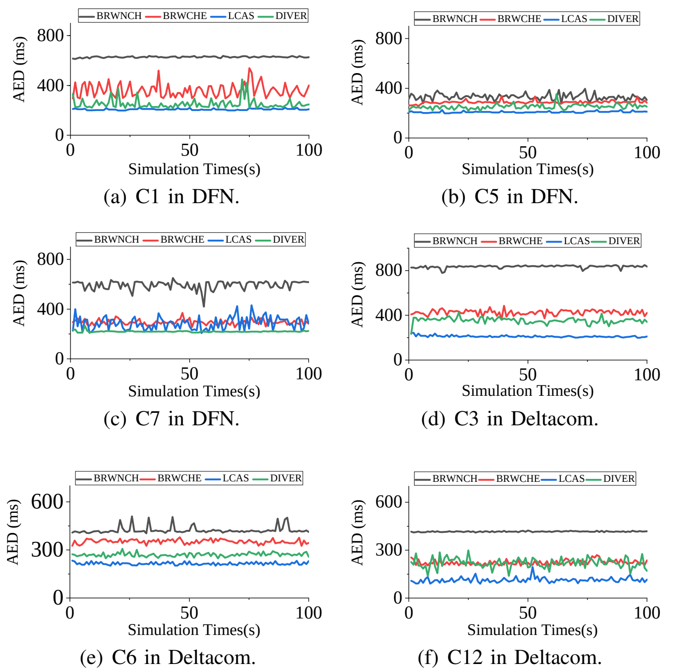
IV. Performance Evaluation

In this section, we use C++ to evaluate the performance of the proposed LCAS.

A. Experimental Settings

We use two popular and realistic topologies, namely German Research Networks, DFN and Deltacom, which are widely applied in NDN researchs [10] in our experiments as illustrated in Fig. 3. There are three kinds of nodes, namely content requesters, content producers and content providers in NDN. To calculate simply, we assume that the size of a content is one unit, and each node has the same size of CS, namely 30. In DFN, there are 12 content requesters, namely C1, C2 ... C12 (in green), 2 content producers (in red), namely P1 and P2, and 24 routers (in blue). While in Deltacom, there are also 12 content requesters, namely C1, C2 ... C12, 6 content producers (in green), namely P1, P2 (in red), and P6, and 83 routers (in blue). A content requester generates content requests following zipf-like distribution with 50 packets/s. The cache strategy and cache replacement strategy are LCE and LRU respectively.

For the content acquirement strategy, recall that DIVER [9] is a very closely related work. It explores the network to search



(a) C1 in DFN. (b) C5 in DFN. (c) C7 in DFN. (d) C3 in Deltacom. (e) C6 in Deltacom. (f) C12 in Deltacom.

Fig. 5. Performance comparisons of Average End-to-end Delay.

and retrieve contents that are closer to the content requesters. It uses probe packets to find contents, and the routers answer the received probe packets by inserting their availability information in a space-efficient data structure. It keeps the most diversified as possible the acquired availability information in the explorations to raise the probability of satisfying future content requests. To demonstrate the performance, we further choose Basic Routing Without Caching (BRWNCH) and Basic Routing With LCE (BRWCHE) as comparisons to show the influence of caching on routing schemes.

B. Experimental Results

We first evaluate *AHC*. To demonstrate clearly, we choose C1, C5 and C7 which have different distances to the nearest content producer in DFN. Specially, C1 has more hop counts than C5 and C7 which have similar hop counts. Furthermore, C7 has a neighbor C6, and they share the same access router, while C5 does not have this kind of neighbors. As shown in Figs. 4 (a), (b), (c), LCAS achieves the best performance than the other three comparisons for its two important algorithms, ARA and NDA. They contribute an important performance improvement for getting the required content from one-hop neighbors and near content providers. While DIVER gets the second best performance, since it retrieves content from the content providers which are close to the content requester. Ignoring the importance of one-hop neighbors makes it consume a higher AHC than our work. BRWCHE has the third performance since it uses LCE to utilize the in-network caching to get the required content from the intermediate nodes to be content providers. Obviously, BRWNCH is the least one, because it abandons in-networking caching and only gets content from the original content producers. The C6 which shares the same access router with C7 will play as a new content provider for it, thus C7 has a lower and steady AHC (getting content from C6) after the higher fluctuation (getting content from original content producer) in the beginning. Similarly, we choose C3, C6 and C12 which have different hop counts to their nearest content producer in Deltacom. The performance comparisons are similar with those in DFN.

To demonstrate the performance comparison better, we further choose *AED* as the second metric. As shown in Fig. 5, recall the analysis in the last paragraph, they have a similar and more accurate values in delay than those in hop counts. The reason of fluctuation is that the limited cache space cannot hold all received content and the simple LRU replaces out some content which maybe required later. Therefore, as analyzed in Section II, an effective caching strategy will combine with the forwarding scheme, and both improve the performance.

V. Conclusion

In this paper, we propose a low-latency content acquirement scheme for NDN. It consists of an ant routing algorithm ARA and a neighbor discovery algorithm NDA. They fully utilize the in-networking caching of NDN to acquire content from some cached content providers and one-hop neighbors respectively. Especially, ARA strengthens some effective routing paths by Data packets laying pheromone along their routing path from the content provider to the content requester to get the required content quickly. NDA undertands the caching information of one-hop neighbors by exchanging information periodically to get the required content from physical proximity. Extensive experiments on DFN and Deltacom have been conducted to show the performances of LCAS in terms of average hop counts and average end-to-end delay.

References

[1] Vasilakos A V, Li Z, Simon G, You W. Information centric network: research challenges and opportunities[J]. Journal of Network and Computer Applications, 2015, 52(1): 1–10.
[2] Zhang L, Afanasyev A, Burke J, et al. Named data networking[J]. ACM SIGCOMM Computer Communication Review, 2014, 44(3): 66-73.
[3] Chen C, Wang C, Qiu T, Atiquzzaman M, Wu D. Caching in vehicular named data networking: architecture, schemes and future directions[J]. IEEE Communications Surveys & Tutorials, 2020, 22(4): 2378-2407.
[4] Ariefianto W T, Syambas N R. Routing in NDN network: a survey and future perspectives[C]//Proceedings of the 11th International Conference on Telecommunication Systems Services and Applications (TSSA), 2017: 1-6.
[5] Ahdan S, Nurhayati A, Nurkahfi G N, Syambas N R. Adaptive forwarding strategy in named data networking: a survey[C]//Proceedings of the 15th International Conference on Telecommunication Systems, Services, and Applications (TSSA), 2021: 1-9.
[6] Din I U, Hassan S, Khan M K, Guizani M, Ghazali O, Habbal A. Caching in information-centric networking: strategies, challenges, and future research directions[J]. IEEE Communications Surveys Tutorials, 2018, 20(2): 1443-1474.
[7] Ghasemi C, Yousefi H, Shin K G, Zhang B. Routing meets caching in named data networks[C]//Proceedings of IEEE International Conference on Computer Communications (INFOCOM), 2018: 1-2.
[8] Cao J, Pei D, Zhang X, Zhang B, and Zhao Y. Fetching popular data from the nearest replica in NDN[C]//Proceedings of International Conference on Computer Communication and Networks (ICCCN), 2016:1-9.
[9] Bastos I V, Moraes I M. A diversity-based search-and-routing approach for named-data networking[J]. Computer Networks, 2019,157(5): 11-23.
[10] Qu D, Lv G, Qu S, Shen H, Yang Y, Heng Z. An effective and lightweight countermeasure scheme to multiple network attacks in NDN[J]. IEEE/ACM Transactions on Networking, 2022, 30(2): 515-528.