NCR-BN Cooperative Caching for ICN Based on Off-Path Cache

Tianyou Wu
Laboratory for Future Networks
Department of Automation
University of Science and Technology of China
Hefei, China
bz1109@mail.ustc.edu.cn

Quan Zheng
Laboratory for Future Networks
Department of Automation
University of Science and Technology of China
Institute of Artificial Intelligence
Hefei, China
qzheng@ustc.edu.cn

Qianbao Shi
Laboratory for Future Networks
Institute of Advanced Technology
USTC
Hefei, China
shiqianbao@iai.ustc.edu.cn

Feng Yang
Institute of Advanced Technology
USTC
Institute of Artificial Intelligence
Hefei, China
youngf@email.ustc.edu.cn

Zhenghuan Xu
Laboratory for Future Networks
Institute of Advanced Technology
USTC
Hefei, China
zhxu86@ustc.edu.cn

2022 5th International Conference on Hot Information-Centric Networking (HotICN) | 979-8-3503-3520-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/HOTICN57539.2022.10036233

***Abstract*—In recent years, Information-Centric Networking (ICN) has attracted great interest as an architecture specifically designed for future networks. Its deployment of caches in network nodes, rather than only servers, allows popular cached content to be stored closer to users, thus effectively improving network performance. Most of the existing research on ICN cache performance has been conducted on the strategy of placing on-path caches, little considering the caches of neighbour nodes, resulting in a large number of off-path caches not being utilised. Therefore it is valuable work to investigate the cooperative caching strategies and performance of off-path caches.**

In this paper, we propose the Neighbor Caching Region Strategy with Backbone Node (NCR-BN), which is used to address the problem that the default routing mechanism of the ICN can only sense the on-path cache. Then, we conduct simulation experiments on the ndnSIM to analyse the caching performance of different caching strategies and compare them to verify the effectiveness of our strategy.

***Index Terms*—Information-Centric Networking, Caching, Off-path resources, Cooperative**

I. Introduction

With the development and progress of society, the Internet has become an integral part of people's lives. Due to the exponential growth in the number of Internet users and the increased demand for multimedia applications, the threat of a data tsunami is approaching step by step [1].

This work was supported by the Key R&D Program of Anhui Province under Grant No. 202004a05020078. We also would like to acknowledge the support from the Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, and the Institute of Advanced Technology, University of Science and Technology of China.

The existing TCP/IP protocol is unable to meet the demands of mass content distribution and faces problems such as insufficient addresses, low security, and insufficient bandwidth. As the function of the Internet has evolved from simple data transmission to secure, efficient, large-scale content acquisition and distribution, it is difficult to radically improve the performance of networks based on existing TCP/IP protocol improvements (**e.g.** IPv6, CDN, etc.). Therefore, as a revolutionary and future-oriented network framework, Information-Centric Networking (ICN) [2] has become a popular direction for network research, and Named Data Network (NDN) [3] is the most used architecture in ICN, in which data transmission is driven by the consumer. Each piece of content or data has a unique name, and consumers request data by sending an interest packet.

NDN naturally designs cache placement along the path from the server to the user side (on-path), meaning that requests from the user side are likely to be satisfied on the path, thus helping to reduce transmission latency and server load. But off-path caches may be ignored. Take the example network in Fig. 1 for a brief explanation, since routers have a finite cache size, for an interest packet from user B, if can't hit cache in Router4, it will only be forwarded to Router5, thus probably missing the potential off-path caches in Router3.

Therefore, recent related research has generated significant interest in off-path cache collaboration, such as MuNCC [4], multi-hop neighbour collaborative caching using Bloom filters; Intra-AS [5], intra-autonomous system collaborative caching. These strategies' target is similar, making use of off-path caches to reduce the redundancy of on-path caches and further improve the efficiency of content delivery. Current research on off-path caching policies such as policy design and performance analysis is still in its infancy, so research and performance analysis of cooperative off-path caching policies is very interesting work.

Based on the above, in the rest of the paper, we illustrate related work, describe the optimal placement planning problem, and briefly explain existing ICN caching strategies in Part II; and in Part III, we propose a cooperative strategy named Neighbor Caching Region Strategy with Backbone Node (NCR-BN); then in Part IV, we conduct experimental analysis with the ndnSIM platform and compare the performance with strategies such as LCE [6] and Intra-AS to further validate the effectiveness of our strategy. The last part is our summary of the paper.

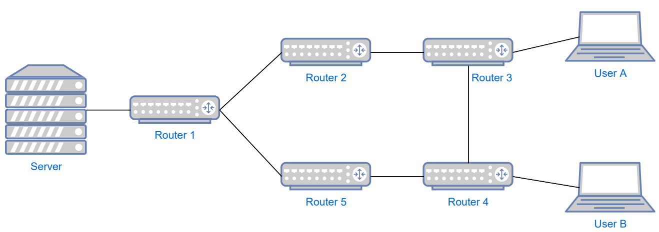


Fig. 1. Simple network for the explanation about on-path and off-path caches

II. RELATED WORK

A. Problem Statement

NDN is a typical representative of the ICN architecture, so we use NDN as an example to illustrate the optimisation problem. The relevant notation is illustrated in Table. I.

TABLE I
NOTATION DESCRIPTION

Notation	Meaning
V	set of cache nodes
v_i	the i_{th} cache node
M	set of content chunk
m_i	the i_{th} content chunk
$p(m)$	popularity of chunk m
C_i	maximum capacity of node v_i
λ_v^m	request rate of chunk m at node v
b_m, v	benifit that chunk m cached at node v

Consider a network region based on NDN, we can describe it by $G(V, E)$, where $V = \{v_1, v_2, ..., v_{|V|}\}$, denotes the NDN cache nodes in the region, and $E = \{e_1, e_2, ..., e_{|E|}\}$ defines the links between cache nodes. And the i_{th} node's cache capacity is C_i. By the definition of popularity, simply get $\sum_{m_i} p(m_i) = 1$. Here we assume that external requests obey Poisson distribution. We construct the problem from the above description as follows.

$$\begin{aligned} \max \quad & \sum_{m \in M} \sum p(m) x_{m,v} b_{m,v} \\ s.t. \quad & \begin{cases} x_{m,v} \in \{0,1\} \\ \sum_m x_{m,v} \le c_v \\ \sum_m p(m) = 1 \end{cases} \end{aligned} \tag{1}$$

The above problem can be described as how to reasonably place content on each node to obtain the maximum benefit given the network topology. It can be seen that the problem is an optimal placement problem under multiple constraints. To solve the problem, global information such as network topology, content popularity, and every node's cache information is required. However, such global information brings a large overhead in terms of collection and computation, and is difficult to adapt to the dynamic network environment. Therefore, to solve the placement problem, strategies of neighbourhood cooperation are generally adopted to achieve improved performance with less overhead.

B. Related Caching Strategies

The default cache-placement strategy used in NDN is LCE (Leave Copy Everywhere), which, as the name suggests, is based on the idea that data packets are cached at every node they pass on their way back to the consumer. LCE is a simple algorithm to implement and is effective in small caching systems. The default cache-lookup strategy for NDN is on-path fetching so that the off-path cache is not visible to the interest packet.

In the Intra-AS strategy, firstly, each node is handled independently by a specific cache management scheme (typically LRU [7] strategy). Ideally, the cache space of an AS should contain the popular items requested by the consumers it serves. At the same time, each cache node periodically passes its stored cache information to its one-hop neighbouring nodes via broadcast. When an interest packet reaches a cache node and the content store (CS) is not hit, it will find out if there is corresponding cached content in the neighbourhood by querying the neighbourhood cache information held by the node, and once there is a match, the node forwards the packet of interest to the corresponding neighbourhood node. To reduce the redundancy of the neighbourhood cache, the policy periodically removes the redundant cache content through periodic neighbourhood de-duplication, thus reducing the redundancy of its cache content based on the neighbourhood cache information. Intra-AS brings good cache performance. But it delegates the neighbourhood de-duplication to each node, making it costly for the link to deliver the neighbourhood information, especially for the links of edge nodes.

There are some other cooperative caching strategies, for example, MuNCC, multi-hop neighbour collaborative caching using Bloom filters, CLS [8] policies, which support out-of-path caching by creating cache traces indicating the location of the cache and the number of hops of the corresponding content, DCM [9] policies, by installing a cache manager to exchange cache information with other node routes, etc. In short, they all make good use of off-path caches.

III. NCR-BN STRATEGY

A. Strategy Overview

Based on the existing cooperative caching strategies summarized above, we designs a new strategy, Neighbor Caching Region Strategy with Backbone Node (NCR-BN). The essence of this strategy is to collect all nodes in a neighbourhood into a Neighbor Caching Region (NCR) and treat it as a whole, then select a suitable node in the Neighbor Caching Region as the backbone node (BN) for cache replacement based on the hierarchical relationship of the nodes. If no backbone node is introduced, and all nodes in the Neighbor Caching Region are considered to be of the same level, NCR-BN degenerates to the Neighbor Caching Region Strategy (NCR[1]).

Each cache node in the NCR will periodically broadcast its cache content information to its one-hop neighbourhood nodes so that all nodes in the NCR will keep the whole cache information from this domain.

In the NCR, the backbone node is selected step by step from the edge of the NDN network using the selection algorithm. In the example network shown in Fig. 2, blue nodes are edge nodes, red nodes like 1, 2, and 3 are backbone nodes, and they form the NCR together with their respective child node. Meanwhile, red node 4 also forms the NCR together with its child node (red nodes 1, 2, and 3) and acts as the backbone node. This shows that the backbone node is relative to the NCR, so that the same node may be both a backbone node in one NCR and a normal node in another NCR. It can be concluded that every node, except for the edge nodes in the network, will be a backbone node for one NCR.

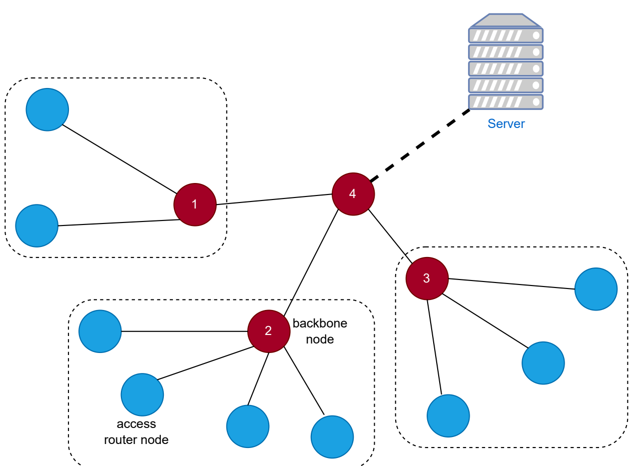


Fig. 2. Example network for explaining backbone node

For our strategy, two major issues need to be addressed. One is the share of cache information between nodes in the same NCR, and the other is how to ensure that when cache replacement occurs, the replaced cache is the least popular data in its NCR. Our designs are as follows.

[1]NCR is regarded as a strategy degenerated from NCR-BN only during the experimental validation(Part IV), and for the rest of this paper it is regarded as the Neighbor Caching Region

B. Cache Information Sharing

We chose interest packets that nodes can send autonomously as the medium for cache information sharing. By adding a field in the interest packet, we separate interest packets into the normal packet and messenger interest packet (MIP). Nodes in one NCR share their cache information by periodically broadcasting messenger interest packets carrying their cache information. Also, considering that too large NCRs may lead to complexity in sharing, the node broadcasts its cache information only to neighbour nodes within one hop.

Every node maintains the Cache Information Table of Region Nodes (CITRN) concerning the cache information of its neighbours. CITRN is a specific map structure consisting of the correspondence of content names and interfaces. When a node receives a messenger interest packet carrying cache information of its neighbour, it will update its CITRN. The CITRN records the current cache content of each neighbouring node. As shown in Fig. 3, when a node receives a normal interest packet and does not hit its cache store and pending interest table, it queries the CITRN, and if it hits the CITRN, it forwards the interest table directly to the corresponding neighbour node. If the content on the neighbor node is not updated, **i.e.** the content is valid, the neighbor node will pack a copy of the content into a data packet and return it, otherwise, the neighbor node will call back the interest packet and inform the node that the requested content has been replaced, and the node will continue to forward the interest packet again according to the FIB table through the original NDN forwarding strategy.

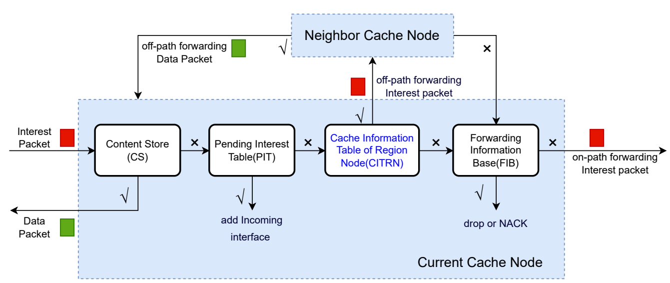


Fig. 3. Interest packet processing for nodes in the NCR-BN strategy

C. Cache Replacement

The default cache replacement policy for NDN is the LRU (Last Recently Used), but since the LRU does not distinguish between popular and non-popular content, during the actual cache replacement process, it is likely to result in popular content being replaced prematurely while non-popular content occupies cache for a long time, resulting in cache imbalance. This situation exists even in the on-path caching policy. Therefore, like the frequency of content in the LFU (Least

Frequently Used) [10], we consider the popularity of content in the NCR to reduce cache imbalance through the backbone node (BN) in our strategy.

The backbone node in the neighbourhood cache region maintains a table named Minimal Popularity Table of Region Nodes (MPTRN), as shown in Table. II. The MPTRN table is a specific map structure consisting of the correspondence of interfaces and minimum popularity. The minimum popularity information is broadcast periodically to the one-hop nodes with the MIPs carrying cache information. When the cache of a node in the NCR is full and the cache replacement occurs, the replaced content is forwarded to its backbone node in the form of a special replaced data packet (RDP), the RDP also piggybacks the popularity of this content from the node.

Upon receiving the RDP, the backbone node compares the content popularity P_{data} piggybacked in the RDP with the lowest local content popularity P_{min} from the MPTRN. If $P_{data} > P_{min}$, the backbone node will forward this RDP to the node with the least popularity in MPTRN and notify it to cache this content. So that a new cache replacement occurs, replacing the lowest popularity content in the NCR with the content in the RDP. Otherwise, $P_{data} \leq P_{min}$, the RDP is discarded.

TABLE II
MPTRN EXAMPLE TABLE

interface	minimum popularity
127	7
128	13
126	15
...	...

Algorithm 1 NCR-BN Interest Packet's forwarding

Arrival: Interest Packet
initialization;
onIncomingInterest:
if normal interest packet **then**
  if data in cache **then**
    return data;
  else if node v is server and provides data **then**
    $send(data)$;
  else if content name is found in CITRN **then**
    $FaceID = getFaceFromCITRN(interest.Name)$;
    $face = getFacewithID(FaceID)$;
    forward interest packet through face;
  else
    forward interest packet to the next hop;
  end if
else if messenger interest packet **then**
  refresh CITRN and MPTRN with Control Interest;
end if

Algorithm 2 NCR-BN Data Packet's forwarding

Arrival: Data Packet
initialization;
onIncomingData:
if normal data packet **then**
  handle by default NDN strategy;
else if replaced data packet **then**
  $DataPopularity = data.getPopularity()$;
  $MinPopularity = CS.getMinPopularity()$;
  if $DataPopularity > MinPopularity$ **then**
    cache this data on the appropriate node;
  else
    drop this data packet;
  end if
end if

D. Forwarding Algorithm

In our NCR-BN strategy, interest packets are divided into two categories: normal interest packets for requesting content and messenger interest packets (MIP) for broadcasting information about cache information; and data packets are also divided into two categories: normal data packets in response to the normal interest packet and replaced data packets (RDP) for delivering replaced content and its popularity to the backbone node. For the four types of packets, we designed the following forwarding algorithm. Algo. 1 and Algo. 2 respectively show the forwarding of interest and data packet.

For the normal interest packet, they are processed according to the process shown in Fig. 3, **i.e.** when an interest packet arrives, it will query the CS and PIT table in order. If neither has a matching item, it will query the CITRN table first for the off-path caches in the NCR, if there has a matching item then forward the interest packet through the corresponding interface, otherwise querying the FIB table and forwarding the interest packet through on-path policy.

When a messenger interest packet arrives, the node will update its CITRN and MPTRN table by comparing the information about the cache information and popularity information in the interest packet with the incoming interface.

For normal data packets, the process is the same as the NDN default policy. When a replaced data packet (RDP) arrives, by comparing the minimum popularity locally in the MPTRN with the popularity of the content piggybacked in the packet, the backbone node chooses to notify another node to cache the replaced content or just discard it.

IV. SIMULATION EXPERIMENT

A. Experimental Eenvironment

In this part, we choose the ndnSIM [11] experimental platform based on the ns-3 for simulation experiment and performance verification, and the ndnSIM version is 2.7.

We choose two types of network topologies for simulation experiments to better analyse the performance of the strategies.

- **Generated-Tree topology:** Generated by the topology generation tool BRITE [12], with 7 layers in total, where the number of child nodes is randomly generated between 0 and 5 starting from the root node. A total of 628 nodes

are generated, with the root node of the tree set as the server and the 377 leaf nodes set as consumers.
- **AT&T-ISP topology:** This realistic network topology is proposed by Heckmann [13], et al. It is similar to one real network of AT&T. Fig. 4 shows this topology, which includes 154 nodes in total with one node set as the server and the 104 edge nodes set as consumers.

To demonstrate the effectiveness of NCR-BN, we set up the experiment to compare with LCE and Intra-AS. The NCR strategy (without setting the backbone node) is also added to this comparison to show the superiority of the backbone node.

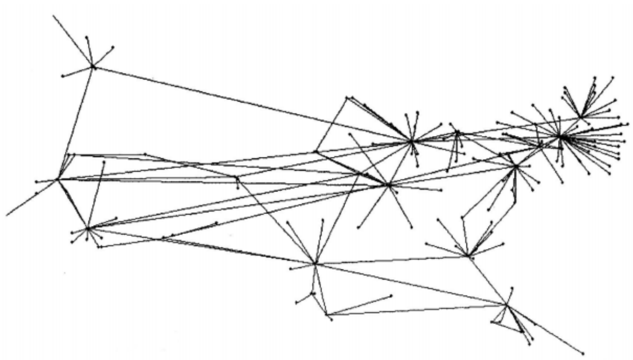


Fig. 4. AT&T-ISP topology

Then, the performance metrics we selected are the global cache hit ratio and the server load [14].

- **Hit_ratio:** As the Eq. 2 shows, it is the ratio of the total number of normal[2] interest packets requested that are hit and responded correctly on non-server nodes. This metric visually demonstrates the caching system's ability to respond to consumers' requests in advance, which is one of the advantages of the NDN network.

$$Hit\_ratio = \frac{\sum_V Hit\_counts}{\sum_V Request\_counts} \quad (2)$$

- **Server_load:** As the Eq. 3 shows, it is the number of normal interest packets that the server needs to process per second, measured in req/s, reflects the pressure on the server to handle the requests. When the load is too high, it will cause the server to be busy and may cause significant network latency or even packet loss, so it is necessary to keep the server load as low as possible.

$$Server\_load = \frac{HitServer\_counts}{Total\_Time} \quad (3)$$

At last, the experimental parameters and variables are set as follows. There are $N = 2000$ types of content in the network, with different data. The size of all the content is $1MB$. The content's popularity obeys the $Zipf-Mandelbrot$ distribution, and the Zipf's exponent α is set between 0.6 and 1.5. Each consumer's request for data obeys Poisson distribution with $\lambda = 10req/s$. Each node has the same cache size C, and the value of C is set between 10 and 100, so that the ratio of cache size to total content $R = \frac{C}{N}$ is set from 0.005 to 0.05, to ensure that the experiment is close to the real situation.

[2]Distinguished from the messenger interest packet (MIP)

When verifying the effect of Zipf's exponent on caching performance, we fix the cache capacity at $C = 50$. And when verifying the effect of cache capacity, we fix the Zipf's exponent at $\alpha = 0.7$ for AT&T's isp topology and $\alpha = 1$ for the generated tree topology.

B. Results and Analysis

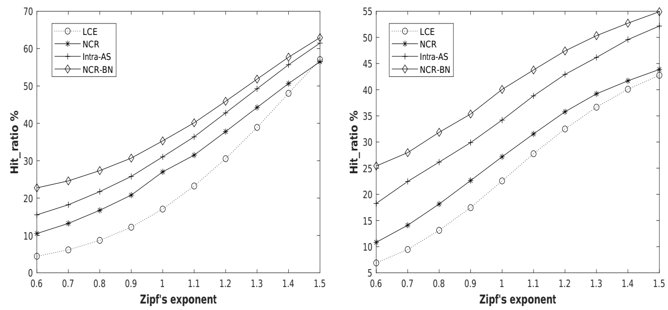
The following figures will show the trends in hit ratio and server load with Zipf's exponent and cache capacity for the two topologies.

Fig. 5 shows the trends in hit ratio with Zipf's exponent for the two topologies. The NCR, Intra-AS, and our NCR-BN strategy all achieve a higher hit ratio than the LCE strategy, with the NCR-BN strategy performing better. As Zipf's exponent increases, requests for highly popular content become more concentrated and the cache tends to store more popular content, making the effect of the off-path caching strategies weaker, thus slowing down the increase in hit ratio from the three off-path strategies. For the generated tree topology, when the Zipf's exponent reaches $\alpha = 1.5$, the improvement of all three off-path strategies relative to LCE is insignificant. But for AT&T's isp topology, even when the Zipf's exponent reaches $\alpha = 1.5$, there is still a good performance increase for NCR-BN and Intra-AS strategies, but NCR's improvement becomes not visible.

Fig. 6 shows the trends in server load with Zipf's exponent for the two topologies. The three off-path strategies all achieve a lower server load than the LCE strategy, with the NCR-BN strategy performing better. With $\alpha = 1$, NCR-BN can reduce the server load by almost 42% for the generated tree topology and 37% for AT&T's isp topology compared to the LCE. Similar to the hit ratio, as Zipf's exponent increases, the improvement of all three off-path strategies relative to LCE is little.

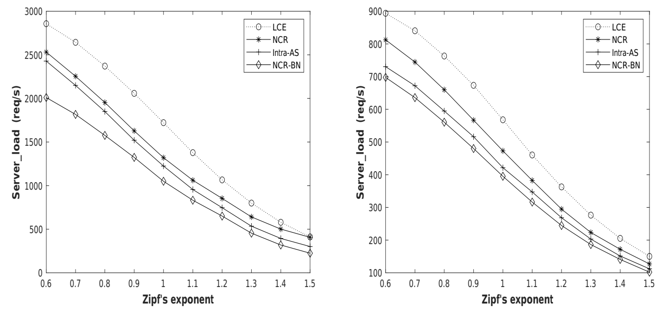
It is clear to see that the off-path strategies are more sensitive to cache capacity. Fig. 7 shows the trends in server load with Zipf's exponent for the two topologies. All three off-path caching strategies have great improvement in hit ratio compared to LCE, with NCR-BN performing better. And the improvement becomes more significant as the cache capacity increases. This is because our NCR-BN strategy makes full use of the cache in the neighbor caching region. For the AT&T isp topology, when the cache capacity is too small, i.e. $C = 10 - 20$, the advantage brought by the backbone node is difficult to be realized, and the NCR-BN's performance is closer to NCR.

The trends in server load with Zipf's exponent for the two topologies are shown in Fig. 8. Whether for the generated tree topology or the AT&T isp topology, three off-path strategies all work well to reduce server load, and our NCR-BN strategy performs better.



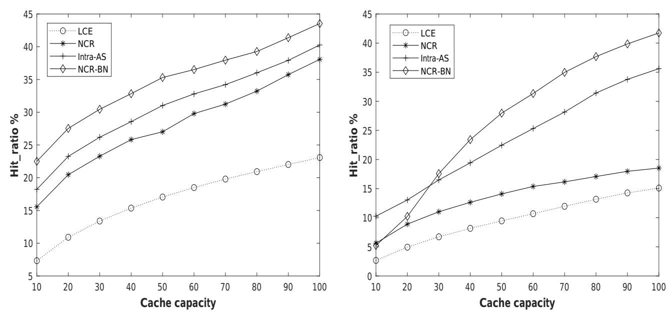
(a) Generated-Tree (b) AT&T-ISP

Fig. 5. Trends in Hit_ratio with Zipf's exponent



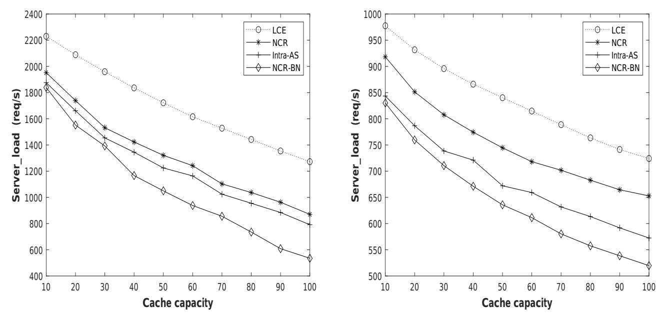
(a) Generated-Tree (b) AT&T-ISP

Fig. 6. Trends in Server_load with Zipf's exponent



(a) Generated-Tree (b) AT&T-ISP

Fig. 7. Trends in Hit_ratio with cache capacity



(a) Generated-Tree (b) AT&T-ISP

Fig. 8. Trends in Server_load with cache capacity

V. CONCLUSION

This paper investigates related work in NDN networks, pointing out that most current caching strategies only consider on-path caching while ignoring the benefits of horizontal off-path cooperative caching. Then we design a new strategy to make full use of the off-path caches. That is the Neighbor Caching Region Strategy with Backbone Node, NCR-BN strategy. Its innovation lies in the adoption of the Backbone node structure, which reduces the redundancy of caches in the neighbour caching regions and thus improves the overall performance of the caching system. Afterward, simulation experiments were conducted on the ndnSIM platform to analyse NCR-BN's caching performance and compare it with the other strategies such as LCE, and Intra-AS. The results showed that our NCR-BN strategy has satisfactory performance in improving the global cache hit ratio and reducing the server load.

REFERENCES

[1] Cisco visual networking index: global mobile data traffic forecast update, 2015–2020 3 Feb 2016. [Online]

[2] Ghodsi A, Shenker S, Koponen T, et al. Information-centric networking: seeing the forest for the trees. ACM Workshop on Hot Topics in Networks, 2011, pp.1-6.

[3] Zhang L, Estrin D, Burke J, et al. Named data networking (NDN) project[J]. Transportation Research Record Journal of the Transportation Research Board, 2010, 1892(1), pp:227-234.

[4] Mick T, Tourani R, Misra S. Muncc: Multi-hop neighborhood collaborative caching in information–entric networks[C]. Proceedings of the 3rd ACM Conference on Information-Centric Networking. 2016, pp:93-101.

[5] Wang J M, Zhang J, Bensaou B. Intra-AS cooperative caching for content-centric networks[C]. ACM SIGCOMM Workshop on Information-Centric NETWORKING. 2013, pp:61-66.

[6] Jacobson V, Smetters D K, Thornton J D, et al. Networking named contentInternational Conference on Emerging NETWORKING Experiments and Technologies. 2009, pp:1-12.

[7] Laoutaris N, Che H, Stavrakakis I. The LCD interconnection of LRU caches and its analysis. Performance Evaluation, 2006, 63(7), pp:609-634.

[8] Li Y, Lin T, Tang H, et al. A chunk caching location and searching scheme in content–entric networking[C]. 2012 IEEE International Conference on Communications (ICC). 2012, pp:2655-2659.

[9] Sourlas V, Gkatzikis L, Flegkas P, et al. Distributed cache management in information-centric networks. IEEE Transactions on Network and Service Management, 2013, 10(3), pp:286-299.

[10] H. Che, Y. Tung, and Z. Wang, Hierarchical Web caching systems: Modeling, design and experimental results, IEEE J. Sel. Areas Commun, vol. 20, no. 7, Sep. 2002, pp. 1305–1314.

[11] Mastorakis S, Afanasyev A, Zhang L. On the evolution of ndnsim: an open-source simulator for NDN experimentation. Computer Communication Review, 2017, pp:19-33.

[12] Medina, Alberto, Lakhina, et al. Brite: An approach to universal topology generation. International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems-Mascots. 2001, pp:346-353.

[13] Heckmann O, Piringer M, Schmitt J, et al. Generating realistic ISP-level network topologies[J]. Communications Letters IEEE, 2003, 7(7), pp:335-336.

[14] Shailendra S, Sengottuvelan S, Rath H K, et al. Performance evaluation of caching policies in ndn-an icn architecture[C]. 2016 IEEE Region 10 Conference (TENCON). IEEE, 2016, pp:1117-1121.