

Content-Centric Network Caching Strategy based on Node Situational Degree

Jianwei Zhang^{a,*}, Chunfeng Du^b, Zengyu Cai^b, Wenqian Wang^b, and Zuodong Wu^a

^aSoftware Engineering College, Zhengzhou University of Light Industry, Zhengzhou, 450002, China

^bSchool of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou, 450002, China

Abstract

In order to meet the requirement of efficient network utilization and cache data availability of content-centric networks, a content-centric network caching strategy based on the node situational degree is proposed. It considers user preferences, content popularity, and node caching. Specifically, the active nodes in CCN network statistics involve user requests, hit rates, time intervals, and cache statuses of nodes according to the period. Through these data, the user preferences around the active node, the content popularity of each type of content in the node, and the cache degree of the node are calculated. Finally, the situational degree of the node is obtained. In the process of user request data and content packet information interaction with active nodes or content source servers, the situational threshold of nodes is calculated. According to dynamic changes in the node situational degree value and the matching of byte value information contained in the interest package and data package, the best caching node can be judged and selected for content replica. The optimized configuration of cached content can be realized. Simulation results show that, compared with the traditional caching scheme, this caching strategy can effectively improve the average cache hit ratio and reduce the request delay.

Keywords: content-centric network; situational factors; situational threshold; situational degree

(Submitted on June 11, 2019; Revised on July 10, 2019; Accepted on August 5, 2019)

© 2019 Totem Publisher, Inc. All rights reserved.

1. Introduction

With the development of network technology and the popularization of various network services, the main body of information obtained by users is also changing. It has changed from traditional text-oriented information to mass audio and video-oriented content information. According to a Cisco forecast [1], by 2021, global Internet users will reach 4.1 billion, and video traffic as the main content category will account for 82% of all IP traffic. With the changes in users' main demands and the transmission of massive content data in the network, the traditional TCP/IP network architecture shows many shortcomings, which make it difficult to meet the needs of the future development of the new generation network. In order to keep the quality of service (QOS) unchanged and adapt to the future network development, the CCN network, as a new generation of content-based network architecture, has been proposed [2] and has received extensive attention. CCN networks realize the transformation of communication from "host-host" to "request content-obtain content". The main body of service is more content than storage location. Different from traditional IP networks, CCN network node routing can cache content information in order to shorten the information request time and improve communication efficiency.

CCN networks have the advantage of Intranet caching [3]. They are the basis and prerequisite for routing and forwarding the data of interest packages requested in the network. At the same time, they are ubiquitous, transparent, fine-grained, and process line speeding, which prevents traditional caching technologies such as Web, CDN, and P2P from being directly and effectively applied to CCN networks [4]. Faced with the massive network transmission of content information, in order to improve the efficiency of CCN network content distribution, avoid the redundant transmission of data in the domain, alleviate the pressure of the network, and realize the efficient use of content resources, the following two points must be achieved: the first is to find suitable cache nodes in the whole network communication link, which can quickly and effectively respond to the same interest request again. Secondly, there is a contradiction between the size of the buffer space

* Corresponding author.

E-mail address: zhangjw@zzu.edu.cn

and the line speed forwarding of the device. For the relatively limited buffer space of the node, it is necessary to replace the contents of the cache in time in order to maintain the diversity of the cache and improve the quality of the cache.

Since the CCN network was proposed, the research of its cache has been a topic of concern to researchers. In order to improve the CCN network cache performance advantages, the existing design methods are considered from the "user", "node", and "content" aspects. "User" is the requester of content and the driver of CCN network communication. Considering the user's preferences, we can select and cache the content we need. "Node" considers the location of content cache and chooses a better location for content cache, which helps to reduce user request latency. "Content" is the content cached in the node. On the one hand, according to the importance of the content, it can cache more important content. It can control the redundancy of the cache and improve the diversity of the cache. The existing cache strategy design is too one-sided when considering the factors affecting the caching strategy, and it fails to fully consider various factors.

To further improve cache efficiency and optimize cache performance, this paper proposes a caching strategy based on node context (CSNC) for CCN networks. The strategy adopts the method of considering situational factors in situational perception. The situational factors such as user preference, content popularity, and node caching are taken into account. The situational degree of nodes is defined, and the formula is given. Through the interactive process of interest package requests and packet responses, content replicas that meet the requirements of the situation degree range of nodes are cached on the necessary nodes in order to optimize the cache performance and enhance the diversity of system caches.

2. Related Work

With the further exploration of CCN networks by scholars, more and more attention has been paid to research on the caching characteristics of CCN networks. Literature [5] studied the default caching strategy LCE (leave copy everywhere) for CCN networks, which requires all nodes on the intra-domain communication link to perform content caching. The feature of caching everywhere generates a large number of redundant content copies in the network. When the remaining space of the node is insufficient, frequent cache content replacement will also cause processing consumption. The focus of literature [6], LCD (leave copy down) is a caching strategy only for the next hop routing node that hits the node. This strategy can control cache redundancy in a short time. As time goes on, repeated requests push content to the edge of the network. At the same time, all the upstream nodes of the link cache redundant content. This strategy cannot achieve optimal redundancy control.

In literature [7], a content popularity-based caching strategy MPC was proposed to store important and popular content in order to improve the "quality" of cached content. Literature [8] proposed a WAVE caching strategy based on the popularity of cached content in order to make the content cached with high popularity.

In literature [9], an online real-time cache strategy RBC-CC was proposed. This caching strategy combines time factors and then proposes weighted content popularity based on content time. Then, the selection of cache placement nodes is realized by using the median centrality and content popularity. However, this strategy lacks consideration of the caching capacity of nodes, and it is prone to early replacement when the caching space of nodes is about to be full. In literature [10], Lee et al. first selected the node with larger available buffer space to cache according to the size of the available buffer capacity of each node on the communication link, which not only reduced the redundancy of the node cache, but also realized the uniform utilization of the node cache. However, when the network is heavily loaded and the buffer space of each node is nearly saturated, it is impossible to decide whether to cache only by the size of the buffer space.

Literature [11] proposed a cache strategy that combines time and content popularity to determine whether or not to cache in a node and how long to store in a node. Literature [12] proposed a probabilistic caching strategy that combines content popularity and the importance of node centers. The strategy classifies the central importance and content popularity of nodes in advance, so that content is cached in the central important nodes in an orderly manner according to the popularity. Although the popularity of content and the importance of nodes were taken into account in literature [11-12], the marginal storage was neglected to save time for users to obtain content, and the costs of content transmission and cache processing were not taken into account.

The existing CCN caching strategy has some shortcomings in considering the factors that affect caching. It considers the popularity of content, caching advantages of important nodes, and importance of nodes, but it fails to take into account user preferences and the capacity of cache space. It also does not make full use of the advantages of nodes in different locations.

3. CSNC Strategy

3.1. CSNC Related Definitions

Because the situational degree value at each node changes with time and period, when the data packet returns, the cache decision will occur at every routing node. Content replicas are cached only at nodes that satisfy certain conditions. In order to save network resources effectively and improve the quality of node caching, CSNC caching decision-making considers all kinds of situational factors of nodes synthetically. User preference, content popularity, and node caching degree are used as evaluation criteria. On this basis, the situational degree of nodes is proposed. User preference, content popularity, and node caching are defined as follows.

Definition 1 User preference. The user preference represents the extent to which users in the region need cached content in the node. It involves the request ratio of users to certain content in a certain period. Considering the whole network topology, users can be divided into ordinary users and node users (all other users directly or indirectly connected with the object node except the object node). Firstly, we need to use a clustering algorithm to cluster user requests. The user preferences of content ci in the statistical period are given in Equation (1).

$$UP_m = \frac{N_{ci}}{N_{all}} \quad (1)$$

Where N_{ci} represents the number of requests for M-class content ci by the user of the connecting node and N_{all} represents the number of requests for all content by the user of the connecting node. The analysis shows that the range of user preferences is $0 \leq UP_m \leq 1$ and the value in the normal range is very small.

Definition 2 Content popularity. The content popularity represents the popularity of content in intra-domain networks at a certain time T , including the frequency of user requests for this content and similar content requests. The time-varying characteristics of user requests for content are considered. The content popularity of each file in the network node is always changing. The popularity is different in different periods, so it is not feasible to consider only a single period. In order to predict content popularity more accurately, we use EWMA [13], which is based on deviation correction, to calculate content popularity in the current cycle according to content popularity in different periods and content hit rate in nodes.

$$P_{ci}(T) = (1 + UP_m) \frac{CN_{ci}}{CN_{all}} \quad (2)$$

$$CP_{ci}(T) = \frac{\beta \times CP_{ci}(T-1) + (1 - \beta)P_{ci}}{1 - \beta^T} \quad (3)$$

In Equation (2), $P_{ci}(T)$ represents the popularity of content in a single period. CN_{ci} represents the number of requests received locally for content ci during the statistical period. CN_{all} represents the total number of requests received during the statistical period. In Equation (3), $CP_{ci}(T)$ represents the content popularity in the current periodic node. CP_{ci} represents the popularity of each content in the previous cycle node. β is the attenuation factor of content popularity during the cycle, and its range is $0 < \beta < 1$.

Definition 3 Node caching. The node caching degree is an important basis to characterize a node's caching state. Whether or not the traditional caching strategy caches at a node depends to a certain extent on the state of the cache space of the node. If the node is saturated or about to be saturated, only relying on the buffer space state of the node cannot accurately describe the actual state of the node cache. There are two parts in the node cache: one is the cache occupancy rate when the cache space is sufficient, and the other is the cache replacement rate when the cache space is saturated or about to be saturated.

$$SC(vi) = \frac{\sum_{i=1}^n O(c_i) + \sum_{j=1}^m S(c_i)}{C(vi)} \quad (4)$$

In Equation (4), $\sum_{i=1}^n O(c_i)$ represents the size of the actual cache content in the node. $\sum_{j=1}^m S(c_i)$ represents the size of the content replica that has been replaced in the node. $C(vi)$ represents the total capacity of the node cache. For the node caching degree, when there is a large amount of memory space, the node caching degree mainly depends on $\sum_{i=1}^n O(c_i)$. As

time goes on, a large number of content replicas will be cached in the nodes. At this time, $\sum_{j=1}^m S(c_i)$ can be used as a measure to compensate for the node cache.

Definition 4 Node situational degree. The node situational degree is a measure of the comprehensive state of nodes. Combined with the consideration of situational factors in situational awareness, it can be integrated into user preferences, content popularity, and node caching as the node attribute category. The nodal situational degree formula is expressed as Equation (5).

$$R_{ci_SITU(vi)} = (CP_{ci}(T) + 1) \times SC(vi) \quad (5)$$

3.2. CSNC Strategy Implementation

The implementation of the CSNC strategy needs to expand the structure of interest packages and data packages in the CNC network. The extended fields are shown in Figure 1. The Interest Hops field is added to the Extended Interest Pack to record the route hops of requests, which is abbreviated as IH. The Interest Situation Degree field is added to record the threshold of scenario degree of each node in the process of forwarding interest packages, which is abbreviated as I_ISD. The field of Data Hops is added to the extended data package as the number of hops of the returned data. The Data Situation Degree field value is used to record the scenario threshold in the return packet, which is abbreviated as D_DSD.

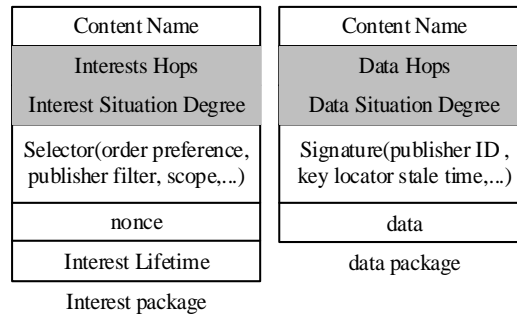


Figure 1. Package structure field extension

Assuming that the requester sends an interest request packet through n -hop node routing to node V with cached content, there is no cached content at each node in the middle. In the process of requesting the forwarding of interest packages, each node is routed and the situational factor value of the node is calculated according to the content information statistics table at the node. The statistical table of content information is shown in Table 1.

Table 1. Content information statistics

Content	User preferences	Content popularity	Cache count	Replacement number
C_1	$UP1$	$CP1$	$N1$	$S1$
C_2	$UP2$	$CP2$	$N2$	$S1$
...
C_n	UPn	CPn	Nn	Sn

Assuming that there are n hops between requesters and cached content nodes, we default here to establish an active path for that content. There are active nodes and active users on the active path. In order to make use of situational factors, we assume that each node in the CCN network has a field $R_{ci_SITU(vi)}$, which can store bytes of situational degree values of nodes. At the same time, set up a triple $(c_i, CP(T)_{ci_max}, SC(v)_{ci_min})$. c_i represents content. $CP(T)_{ci_max}$ stores the maximum popularity of node content. $SC(v)_{ci_min}$ stores the minimum cache. Initialize $CP(T)_{ci_max} = 0$, and set $SC(v)_{ci_min}$ to a maximum, such as $SC(v)_{ci_min} = 1,000$.

In the process of forwarding the requester's interest packages, it is necessary to compare the byte values in the interest packages. If the CP_i in the node is greater than $CP(T)_{ci_max}$, $CP(T)_{ci_max} = CP_i$ is executed. If $SC(v) < SC(v)_{ci_min}$ is obtained from the information in the node, then $SC(v)_{ci_min} = SC(v)$ is executed. $CP(T)_{ci_max_SITU}$ computed by each

node is assigned to I_ISD . If the content is not hit, the IH value is added to 1 and forwarded after each node is routed. At the same time, the corresponding request entry should be added to the PIT.

When the content Ci arrives at the caching node Vi along the established active path, the node calculates the situational threshold $R_{ci_max_SITU}$ for the content Ci based on the data in the triple. The specific calculation is shown in Equation (6).

$$R_{ci_max_SITU} = (CP(T)_{ci_max} + 1) \times SC(v)_{ci_min} \quad (6)$$

The response packet is forwarded along the "reverse path" and returns to the user request node. In the process of accessing each node, it will decide whether to cache at the node according to the real-time situational degree value of the intermediate node and the value of D_DSD bytes in the data packet. When the content is hit at the node, the $R_{ci_min_SITU}(vi)$ value calculated at the cache node is assigned to the D_DSD byte value, and $DH=IH-1$ is set at the same time.

When the response packet returns to the intermediate node passing through the active path, it is discussed according to the node and D_DSD byte value:

1) If $R_{ci_SITU}(vi) < D_DSD$ is satisfied, this node is considered suitable for storing content and performing caching at this node. At the same time, the DH byte value minus 1 forwards the data packet and then deletes the corresponding PIT entries in the node.

2) If $R_{ci_SITU}(vi) \geq D_DSD$ is satisfied, this node is not suitable for storing cached content. At the same time, when the DH byte value is reduced by 1, the data packet is forwarded directly, and the corresponding PIT entries in the nodes are deleted.

Above are the interest packet forwarding, packet response, and caching decision-making processes. In the packet return phase, the path node only needs to compare the size of the D_DSD byte value and $R_{ci_SITU}(vi)$ value of the packet. Based on the comparison results, it is decided whether to perform the caching operation at the node. Then, the DH value is reduced by 1 and forwarded to the next hop until the data packet is returned to the user. Finally, the whole request-response process is completed.

3.3. Algorithm Description

In order to further describe and understand the implementation process of the CSNC strategy, the pseudo code of the initialization process, interest package process, data package process, and caching process are given here.

Algorithm 1 Initialization

1. $UPi = \text{null}, CPi = \text{null}, SC(Vi) = \text{null};$
 2. $IH = \text{null}, DH = \text{null};$
 3. $D_DSD = \text{null}, I_ISD = \text{null};$
 4. $CP(T)_{ci_max} = 0, SC(v)_{ci_min} = 1000;$
 5. Empty the intermediate node cache;
 6. End
-

Algorithm 2 Forwarding Process of Interest Packet

Input: Request for interest package

Output: Hit the cached content or forward the next hop

1. Receive the interest packet sent by the user;
 2. If (CS missed the cache);
 3. Calculate $CPi, SC(vi);$
 4. Compare the size of $CP(T)_{ci_max}$ and CPi and assign the maximum value to;
 5. Compare the size of and CPi and assign the minimum value to;
 6. $IH ++;$
 7. Forwarding interest packet to the next node;
 8. Else;
 9. $R_{ci_max_SITU} = (CP(T)_{ci_max} + 1) \times SC(v)_{ci_min} ;$
 10. $I_ISD = R_{ci_max_SITU} ;$
 11. Perform algorithm 3;
 12. End
-

Algorithm 3 Data Packet Forwarding Process**Input:** Request for interest package**Output:** Whether to cache or continue forwarding

```

1.  Read  $IH, I\_ISD$ ;
2.   $DH = IH - 1, D\_DSD = I\_ISD$ ;
3.  Return the packet along the reverse path and calculate;
4.  If(  $R_{ci\_SITU}(vi) < D\_DSD$  )
5.    If( $CS$  is full)
6.      LRU replacement algorithm and cache;
7.    Else
8.      Cache data;
9.    Else
10.    $DHi$  -- forward the packet after;
11. End

```

In Algorithm 1, any node, interest package, and data package in the network are initialized. The operation assigns null to interest packages and packet hops. At the same time, the initial values of situational factors such as user preference, content popularity, cache package, and degree are set to null. Each newly added request interest package sets the I_ISD byte value to null. In Algorithm 2, if the user requests the interest package to hit the access node, the packet will be returned directly along the reverse path. Otherwise, the information in the node information table and the bytes in the triple are used for matching and exchanging values. After reaching the hitting node, it satisfies the maximum content popularity on the active path in field $CP(T)_{ci_max}$ and the minimum cache on the active path in field $SC(V)_{ci_min}$. At the same time, the IH value is added to 1 and forwarded. In Algorithm 3, when the data packet returns along the communication link, it compares the D_DSD byte value with the node situational degree value and judges whether the replica of the content needs to be cached at the node by comparing the node situational value with the threshold value.

4. Simulation Experiment

4.1. Experiment Setup

The simulation experiment adopts ndnSIM [14], which is an open source network simulation platform based on NS-3 [15] for content-centric networks. In the experiment, a network topology with 50 nodes is generated randomly by using modules in GT-ITM. In the simulation experiment, it is assumed that the request arrival node obeys the Poisson distribution. The user access obeys the Zipf [16] distribution with parameters. Suppose the CCN network prototype consists of 20 requesters, 1 content source server, and 1,000 network content blocks of the same size. The content source server contains all copies of content in the network and will not be deleted. The content popularity attenuation factor A is set to 0.4. Each cache replica block is the same size, and the total cache capacity of each node is the same size. By default, nodes can cache 60 copies of content. The simulation time is 100 seconds. The specific parameters are shown in Table 2.

Table 2. Simulation parameters

Parameter	Default value	Change range
Popularity Attenuation Factor β	0.4	
Network Topology Node Number N	50	
Cache capacity/slot	60	20-100
Zipf parameter range α	0.7	0.2-1.2

4.2. Simulation and Evaluation

In order to make a comparison, simulation experiments are carried out on LCE, Betw, ProbCache, and the proposed strategies. The simulation experiment mainly focuses on the quantitative analysis and comparison of the differences among the cache strategy indicators under the condition that the size of the buffer capacity and the change of parameter α obey the Zipf distribution. The evaluation indexes of this simulation experiment are as follows.

(1) The cache hit rate refers to the response rate of the node to the requester's request for cached content, which can be expressed by the ratio of the number of cache hits in the network domain to the total number of user requests. The higher the cache hit rate in the domain, the fewer requests received by the corresponding original server, the smaller the corresponding load, and the higher the system efficiency.

$$CH = \sum_{i=1}^n N_{I_H} / N_{I_{ALL}} \quad (7)$$

In Equation (7), N_{I_H} represents the number of cache hits in a single node's routing, $N_{I_{ALL}}$ represents the total number of interest requests in the intra-domain network, and N represents the number of routing nodes shared by the intra-domain network.

(2) The average request latency is the ratio of the total time that all users request content to the total number of users receiving the packet in the network. The smaller the average request latency of network users in the domain, the shorter the time that users request to obtain data content, and the better the user's experience.

$$ARD = \sum_{i=1}^n U_{T_D} / U_{all} \quad (8)$$

In Equation (8), U_{T_D} represents the request latency of a single user. U_{all} represents the total number of users and is equal in value to n .

4.3. Result Analysis

In order to facilitate comparison and analysis, simulation experiments are carried out on LCE, Betw, ProbCache, and the strategies proposed in this paper. As can be seen from the graph, with the increase in the node cache capacity, the cache hit rate of several caching strategies is improved, and the average request latency of users is reduced accordingly.

4.3.1. Impact of Cache Capacity

As can be seen in Figure 2(a), the cache hit rates of the LCE and Betw cache strategies increase with the increase in the cache capacity. LCE's ubiquitous caching strategy and Betw's node-based caching strategy do not fully consider the impact of the caching capacity. Although the cache hit rate will be improved to some extent, there is little difference between them. As can be seen from Figure 2(a), the cache hit rate of each strategy varies with the same cache size. Under the same conditions, the cache hit rates of four strategies increased by 0.016, 0.023, 0.037, and 0.043% for each 10 (slot) increase. At the same time, CSNC has the best effect on improving the cache hit rate among several caching strategies. As can be seen from Figure 2(b), the average request latency of each strategy varies with the same cache size. In the same case, the proposed caching strategy is superior to other caching strategies in reducing the average request latency of users. The comparison results show that the proposed CSNC caching strategy is superior to other caching strategies in the above two indicators.

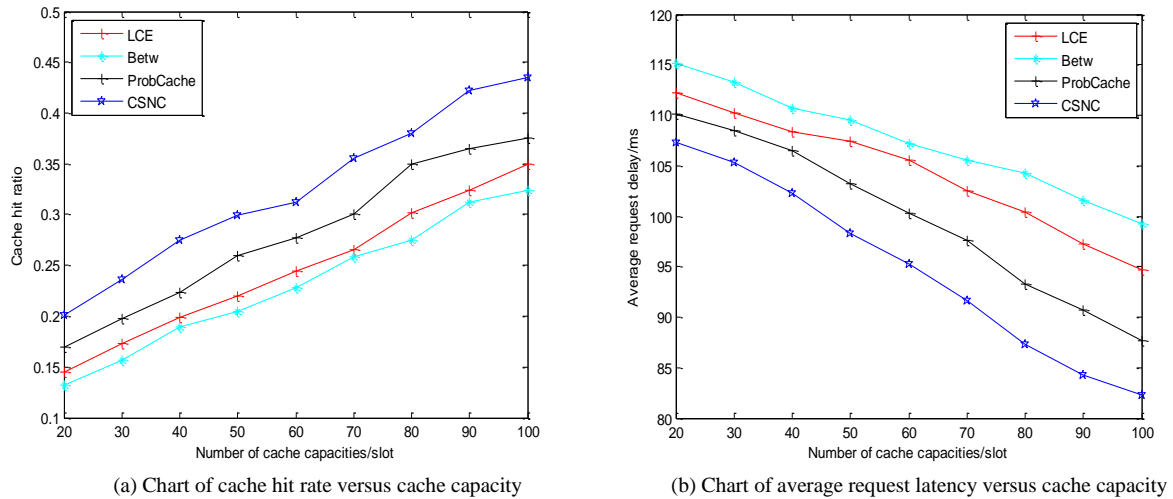


Figure 2. Cache capacity impact

4.3.2. Effect of Zipf Distribution Parameters α

As can be seen from Figure 3, with the increase in the Zipf distribution parameter α , the performance of the corresponding

four caching strategies is improved in terms of cache hit rate and average request delay. In Figure 3(a), with the increase in the parameter α , requests for instructions become more and more centralized. Among the four corresponding strategies, the intranet cache utilization increases, so the cache hit rate also increases. The centralization of requests makes the latency of requests smaller and smaller. Due to the limitation of redundancy in the network and the size of buffer space, the performance change rate increases first and then decreases, but the overall performance is improved. In Figure 3(b), as the parameters increase and requests become more concentrated, Betw outperforms the LCE strategy first followed by both approaches. This is because the Betw policy based on high-median caching outperforms the LCE policy in the case of fewer requests. However, as the request concentration increases, the latter is not very different. By comparing and analyzing these two indicators, the CSNC strategy proposed in this paper is superior to the other three strategies in performance. It can effectively improve the cache hit rate and reduce the average request delay.

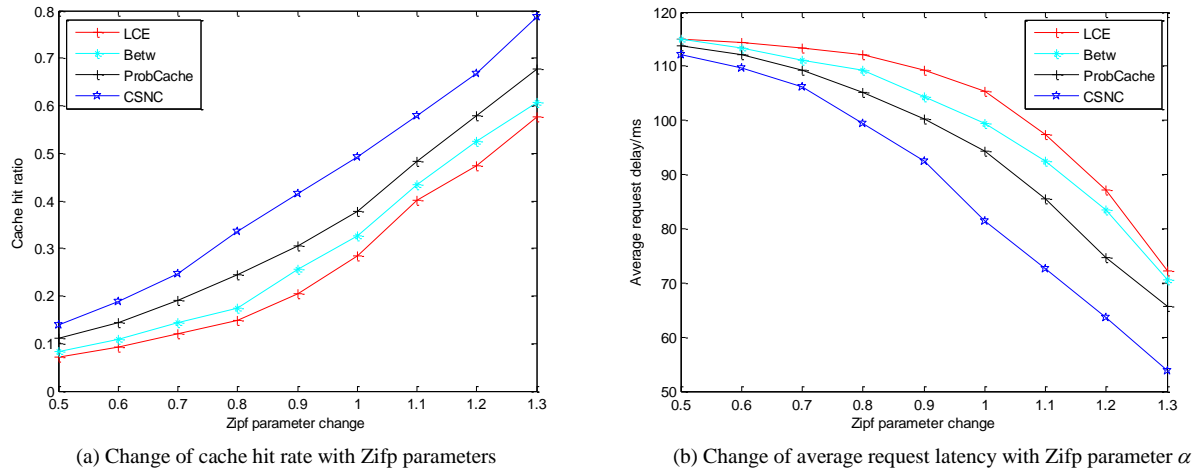


Figure 3. Obey the influence of parameter α of Zifp

5. Conclusions

One of the big advantages of content-centric networks from other traditional networks is the intra-network cache. The traditional cache method does not improve the network cache advantage because of its inadequacies in many aspects. In order to improve the performance of CCN network caching, this paper proposes a CCN network caching strategy based on node situational degree. The location of the cache is determined by comparing the changes in situational degree and situational threshold in different node cycle times. The simulation results show that the CSNC cache strategy has good advantages of improving the cache hit rate of nodes and reducing the average request latency of users.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 61672471 and 61502436), Fundamental Research Funds for the Henan Province University (No. 17KYYWF0202), Henan Province University Science and Technology Innovation Team (No. 18IRTSTHN012), and Plan for Scientific Innovation Talent of Henan Province (No. 184200510010).

References

1. Cisco Systems, Inc., "Cisco Global Cloud Index: Forecast and Methodology (2016-2021) White Paper," (http://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/Cloud_Index_White_Paper.html, Last accessed on February 01, 2018)
2. V. Jacobson, D. K. Smetters, and J. D. Thrnton, "Networking Named Content," *Communications of the ACM*, Vol. 55, No. 1, pp 1-12, 2009
3. B. Ahlgren, C. Dannewitz, and A. C. Imbrend, "A Survey of Information-Centric Networking," *IEEE Communications Magazine*, Vol. 50, No. 7, pp. 26-36, July 2012
4. M. J. Dong, "Based on the Popularity Prediction Center Network Research Content Caching Technology," Beijing University of Posts and Telecommunications, Beijing, 2015
5. D. Yao, Z. Ji, and G. Chen, "ICN Cooperative Caching based on Node Heat and Cache Replacement Rate," *Computer Engineering*, Vol. 44, No. 2, pp. 56-60, 2018
6. N. Laoutaris, H. Che, and I. Stavrakakis, "The LCD Interconnection of LRU Caches and Its Analysis," *Performance Evaluation*,

- Vol. 63, No. 7, pp. 609-634, 2006
7. R. Zhang, J. Liu, and T. Huang, "Popularity based Probabilistic Caching Strategy Design for Named Data Networking," in *Proceedings of IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 476-481, Atlanta, June 2017
 8. W. K. Chai, D. He, and I. PSARAS, "Cache 'Less for More' in Information-Centric Networks (Extended Version)," *Computer Communications*, Vol. 36, No. 7, pp. 758-770, 2013
 9. J. Li, H. Wu, and B. Liu, "RBC-CC: RBC-based Cascade Caching Scheme for Content-Centric Networking," *Journal of Network and Systems Management*, Vol. 25, No. 2, pp. 375-396, 2017
 10. I. Psatsa, W. K. Chai, and G. Pavlou, "Probabilistic in-Network Caching for Information-Centric Networks," in *Proceedings of the 2nd ICN Workshop on Information-centric Networking*, pp. 55-60, ACM, New York, 2012
 11. X. Zhou and Z. Ye, "Popularity and Age based Cache Scheme for Content Centric Network," in *Proceedings of International Conference on Information Management*, pp. 322-326, IEEE, Piscataway, April 2017
 12. Y. Li, T. Zhang, and X. Xu, "Content Popularity and Node Level Matched based Probability Caching for Content Centric Networks," in *Proceedings of IEEE/CIC International Conference on Communications in China*, pp. 1-6, IEEE, Piscataway, July 2016
 13. Q. Ling, L. Xu, and J. Yan, "An Adaptive Caching Algorithm Suitable for Time-Varying User Accesses in VOD Systems," *Multimedia Tools and Applications*, Vol. 74, No. 24, pp. 11117-11137, August 2015
 14. J. Bi, "MCBS: Matrix Computation based Simulator of NDN," *Journal of Advances in Information Technology*, Vol. 9, No. 24, pp. 36-39, 2014
 15. A. Afanasyev, I. Moiseenko, and L. Zhang, "NdNSIM: NDN Simulator for NS-3, NDN-0005," University of California, Los Angeles, 2012
 16. R. Huo, J. Liu, and G. Huan, "Cooperative Caching Strategy based on Correlation Probability in Information Centric Networking," *Journal of Beijing University of Posts and Telecommunications*, Vol. 38, No. 1, pp. 16-20, 2015

Jianwei Zhang received his Ph.D. in computer application technology from PLA Information Engineering University in 2010. He is currently a professor at Zhengzhou University of Light Industry. His research interests include broadband information networks and network security.

Chunfeng Du is a master's student in the School of Computer and Communication Engineering at Zhengzhou University of Light Industry. His research interests include network engineering and big data.

Zengyu Cai received his master's degree in computer application technology from Northeast Normal University in 2006. He is currently an associate professor at Zhengzhou University of Light Industry. His research interests include trusted computing, plan recognition, and information security.

Wenqian Wang is a graduate student in the School of Computer and Communication Engineering at Zhengzhou Light Industry University. Her research interests include content centric networks.

Zuodong Wu is a graduate student in the School of Computer and Communication Engineering at Zhengzhou Light Industry University. His research interests include content-centric networks and network security.