http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (23): 7083-7089, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.7083.7089

Performance Assessment of Cache Strategies in Content Centric Network

Guoyin Zhang, Bin Tang, Peng Wang, Yanxia Wu and Xinyu Zhang
College of Computer Science and Technology, Harbin Engineering University,
Harbin 150001, China

Abstract: In-net caching is an important part of the content centric network architecture. So far, the default "Leave Copy Everywhere (LCE)" caching policy will cache all content without distinction, lead to a large number of cache content redundant in the nodes of network. Due to the limitation of node cache size, redundant reduced the verity of cache content cached in network, result in bad utilization of cache space. To boost the performance of the cache, many improved mechanisms have been put forward, including LCD, MCD and ProbCache caching policies. This article respectively experimented the LCE, LCD, MCD and ProbCache algorithm, judging by standards like content response delay, network link load, server stress, evaluation index, analyzed factors, such as network topology, content popularity and cache space, have what kind of effect on these policies' performance.

Key words: Content centric network, caching policy, ndnSim, cache placement

INTRODUCTION

With the development of Internet technology, as well as the growth of user's demand for large data acquisition and distribution, traditional host-to-host Internet architecture has exposed many problems, such as scalability, safety and mobility, etc. Therefore, a series of improvements have been put forward, Content Centric Network (CCN) architecture is one of them. Among next generation network Architectures such as NDN, MobilityFirst, the NEBULA, XIA, CCN has more advantages and is now getting more and more attention. CCN advantages include many features such as web caching, naming content, requester drives, data security. Among these, web caching is an important part of the CCN system structure.

At present many research institutions have begun their studies on next generation network architecture and the cache in CCN: Zhang *et al*. (2010) in University of California, Los Angeles (UCLA) purposed NDN architecture and constantly improving the architecture details, it is important to verify and guiding the development direction of NDN system structure; Xerox's Palo Alto research centre (PARC) realized an prototype protocol CCNx of the according to the NDN structure; Bell Labs (Carofiglio *et al*., 2011a; 2011b), Alcatel-Lucent and Orange Labs (Muscariello *et al*., 2011) proposed the basic theory model of CCN cache and Psaras *et al*. (2011), Chai *et al*., (2012) in the University of London also researched cache problem of CCN. Domestic institutes took research of CCN included the Institute of computing technology in Chinese academy of sciences (Yang *et al*., 2011), Beijing University of Post and Telecommunications, Shenzhen research institute of Beijing university (Li *et al*., 2012), information technology research institute of Tsinghua University (Chai *et al*., 2012), Chongqing university of Posts and Telecommunications, Huawei research institute and other research institutions, etc.

ALGORITHM DESCRIPTION

In-network caching is an important part of CCN architecture. Caching the content to CCN nodes can reduce the pressure of the server, lower the network load and data request delay, enhance mobility and thus improve the network performance. In CCN, each router has its cache space to store chunks it forwarded and the performance of data placement policy will greatly influence the whole performance of the CCN architecture. Data placement policy is mainly divided into two categories: explicit cooperative caching placement policy and implicit cooperative caching placement policy. In explicit cooperative caching placement policy, nodes share their status and information and decide what, when, where to cache through these information. Unlike explicit cooperative caching placement policy, implicit cooperative caching placement policy relies on local cache management strategy and relative location of web cache node to achieve good performance.

Corresponding Author: Bin Tang, College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China

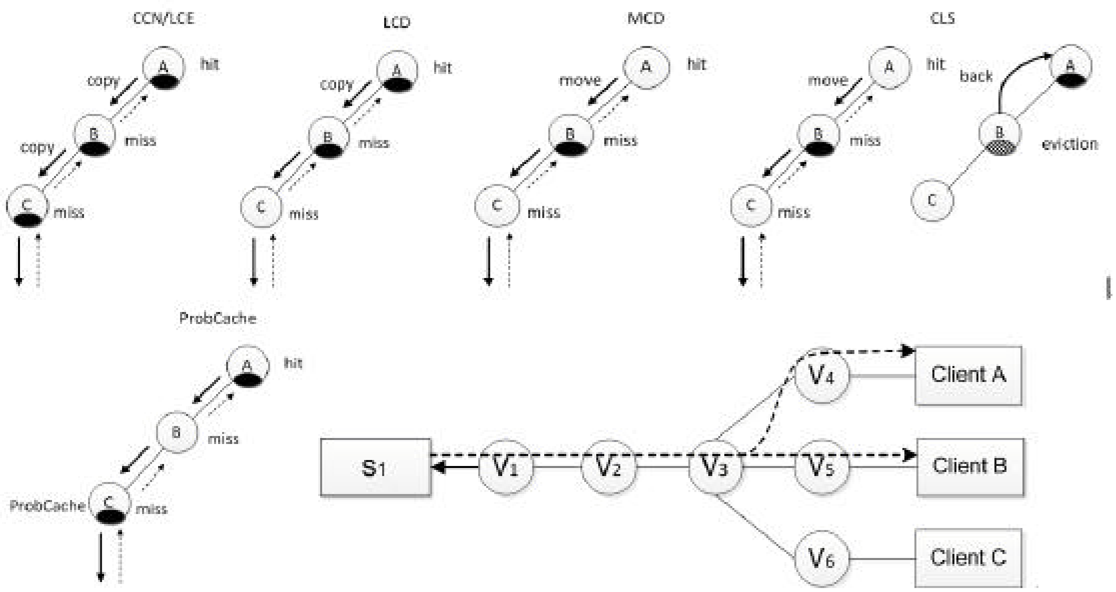

Fig. 1: Process of caching policies

Due to the high complexity algorithm of explicit cooperative caching policy, it couldn't meet the needs CCN nodes, which require high-speed content forwarding. Therefore, the research of CCN caching policy mainly concentrated on the implicit caching policy. This article mainly concerns about existed implicit cache cooperative caching placement policy such as LCE, LCD, MCD, ProbCache. The improvement mainly consisted by two things: one is by increasing the variety of content, to improve the cache hit ratio, reduce the pressure of the server; another is to make cache content marginalized, caching contents nodes closer to the requester, reduce the request delay. Through the process of single way content request, the LCD, MCD and ProbCache strategy in the default LCE improvement ideas on strategy are described in Fig. 1.

As shown in Fig. 1, in case of content request there are Interest packets going through node C, forwarded to node B, finally got the content at node A. When the node A hit request content, LCE will cache the contents on node B and C both; LCD will only cache content on the next level node B; MCD cache content on node B while delete content on node A at the same time, except the server nodes; CLS is similar with MCD, the difference is when the data on node B is deleted, the data will be stored up to the node A and trace back to the server node. Compared with LCE, LCD will cache copy of the content to the next level as well, much closer to the requester, save other nodes' cache space as well and could cache more types of content. But if some contents being requested frequently, then after a period of time LCD will results to all the nodes on the link caching these content, therefore reduced the cache content types. MCD ensured that there are only one copy of content cached on single-link path. In the process of the request, the copy of the content will be gradually cached to nodes nearer to the requester, make full use of node's cache space and ensured the variety of contents that network can cache. But after some content been replaced out of the cache on link, in case that the content is requested again, the request could only be responded by server then, therefore increased the content request delay. CLS proposed the way of recall, ensures that when content being replaced out the node, there are still nodes that cached copy of the content in the link-path. Also it purposed the cache content assist routing mode, increased the efficiency of network content distribution.

ProbCache, according to the probability of computing nodes, determines which node the data will be storage to, as following form:

$$\text{ProbCatche}(x) = \frac{\sum_{i=1}^{c-(x-1)} N_i}{T_{tw} N_x} \times \frac{x}{c}$$

where, N_i is the cache size of r_i, x is the Time Since Birth (TSB), c is Time Since Inception (TSI). Less for More, on the other hand, calculating node's central value to find the optimal cache node, as following form:

Table 1: Comparison of cache policies

Algorithm	Complexity	Topology	Verity	Popularity	Redundancy
LCE	Low	Simple	Low	High	High
LCD	Low	Average	Low	High	Average
MCD	Average	Complex	Average	Average	Average
CLS	High	Tree	High	Low	Low
Prob cache	High	Complex	High	Average	Low

Table 2: Experiment parameters

Parameter	Content quantity	Chunk size	Cache space	Node delay	Link bandwidth	Zipf parameter α	Request frequency
Value	1000/1000	1024 KB	100	10 m sec	10 Mbps	0.7	100/sec

$$C_B(v) = \sum_{i \neq v \neq j \in V} \frac{\sigma_{i,j}(v)}{\sigma_{i,j}}$$

where, $\sigma_{i,j}$ is the number of content distribution paths, $\sigma_{i,j}(v)$ is the number of paths which bypassed node v between node i and node j. Compared with LCD and MCD, ProbCache and LessForMore algorithm have higher complexity. However, due to the information been searched on many nodes of link-path, the judgment that each node makes could reflect the real situation of the network more precisely and result in better general benefits. The comparison is shown in Table 1.

PERFORMANCE EVALUATION

In this study, the experimental use simulation platform ndnSIM based on NS-3, the experiment topology include one link topology, binary tree topology network and scale-free network topology which simulated real network. Routing forward strategy is BestRoute routing strategy provided by ndnSIM, user requests follow Zipf-Like process, all nodes have the same size of cache space. Main parameters in the configuration of the experiment are shown in Table 2.

The purpose of network's cache mainly includes three aspects: first, reduce the content delivery delay, by accessing contents from cache closer to nodes, the request delay to the server can be reduced; Second, reduce the network traffic and congestion, accessing contents from cache closer to nodes could reduce the bandwidth loss of transmission between user and server in the entire link-path. Third, reduce the load on the server, network node response content request, can reduce the number of requests to the server and thus reduce the load on the server. Therefore, from performance improvement brought by the cache, we use cache hit-rate, hop decrement, server hit decrement and content average request delay these four types of performance evaluation indicators to measure the performance of caching policies.

Cache hit-rate: Network cache hit-rate could be comprehended as the average hit-rate of cached contents in the entire network, the network cache hit-rate Hit_{avg} is calculated as followed:

$$Hit_{avg} = \frac{\sum_{i=1}^{N_u}\sum_{j=1}^{C_u}(\text{req}(i, j) - \text{serv}(i, j))}{\sum_{i=1}^{N_u}\sum_{j=1}^{C_u}\text{req}(i, j)}$$

where, req(i,j) represents the number of times that user node i requested content j, serv(i,j) is the number of times that source respond to node i request for content j. N_u is the total number of user nodes, C_n is the total number of the types of content in a network.

Hop decrement: The percentage of hop decrement $\beta(t)$ is shown as follows:

$$\beta(t) = \frac{\sum_{r=1}^{R} h_r(t)}{\sum_{r=1}^{R} H_r(t)}$$

where, $H_r(t)$ is the link length of content f_r between user to server in condition of node didn't cache contents in unit-time. $h_r(t)$ is the number of hops in request content f_r using cache policy in unit-time. When no cache is hit on the path to server, $h_r = H_r$. Hop decrement is the percentage of content request hops and the path hops between user and server. In absence of cache policy, $\beta = 1.0$:

$$\gamma(t) = \frac{\sum_{r=1}^{R} w_r(t)}{\sum_{r=1}^{R} W_r(t)}$$

where, $W_r(t)$ is request frequency of content f_r in unit-time, $w_r(t)$ is the number of hits on server by content f_r in unit-time. This is a effective reflection of server's load in different policies.

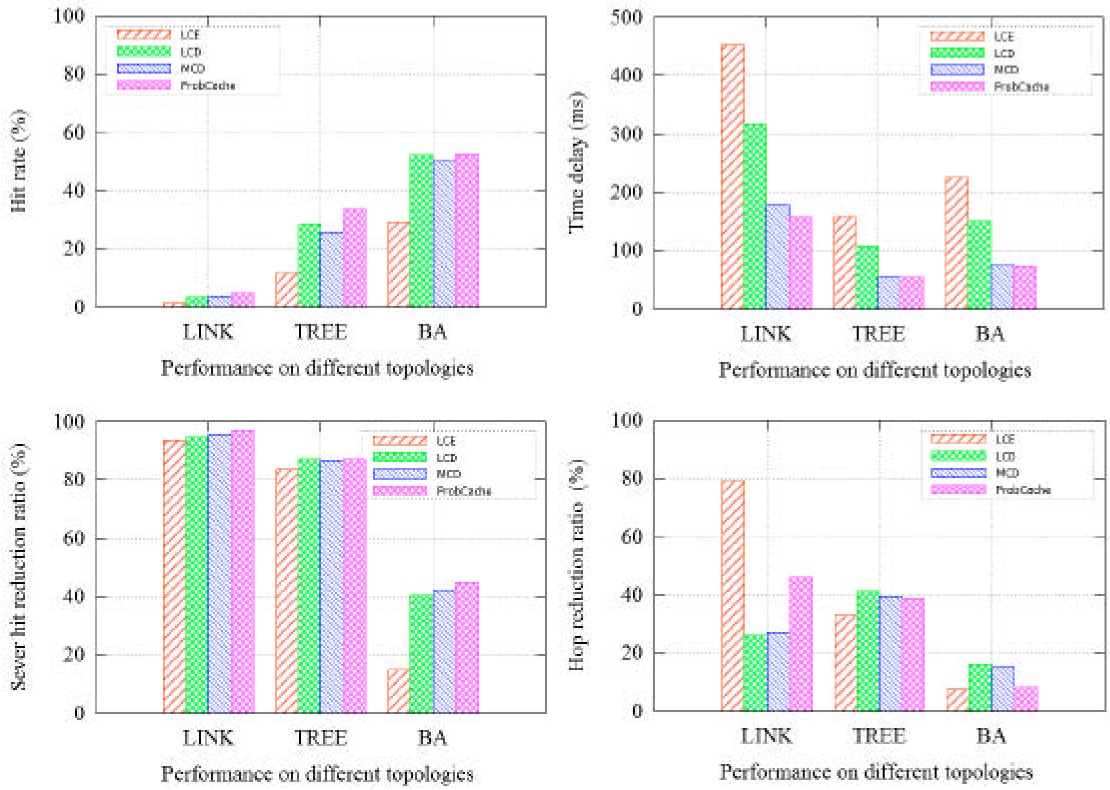

Fig. 2(a-d): Effect of topology on (a) Hit-rate (b) Delay (c) Server hit decrement and (d) Hop decrement

Content request delay: The time interval from user sent content request to the content response. In this study, the content request delay means the average request delay of content.

Effect of topology on cache policy: Under different environment of topologies, we compared performances of LCE, LCD, MCD and ProbCache. Topology respectively takes the aforementioned single LINK topology (LINK), binary TREE network topology (TREE) and scale-free network topology (BA) which simulated real network. Content distribution follow Zipf-Like distribution, α = 0.7, running time is 300 s. Other parameters are referred in Table 2.

Server hit decrement: As shown in Fig. 2a and c, LCD, MCD, ProbCache have similar hit-rate and server hit decrement in different topologies and improved 50% compared with LCE, proved that they have significant effect on improving hit-rate. As shown in Fig. 2b, four caching policies in different topologies have different network delay: in single link-path, the delay of ProbCatche is 10% lower than MCD, 35% lower than LCD, 50% lower than LCE; in binary tree, the delay of ProbCatche 5% lower than MCD, 40% lower than LCD, 65% lower than the LCE; in scale-free networks, the delay of ProbCache is 5% lower than MCD, 45% lower than LCD, 60% lower than LCE. As shown in Fig. 2d, ProbCache and LCE have decreasing hop decrement, while the LCD and MCD caching policy is much different, the performance is optimal in the binary tree, which reached 50 and 20% higher than single link-path, 35% higher than a scale-free network. This is because of the characteristic of the ProbCache is being able to make the requested content closer to the requester faster, therefore the hop decrement of ProbCache is optimal. This makes the request get the content in nodes near the requester, therefore greatly reduced the network delay.

Effect of popularity on cache policy: Under the environment of different popularity, performance of LCE, LCD, MCD and ProbCache are compared. Popularity, respectively take α = 0.3, 0.7, 0.3 and 0.7, content distribution follow the Zipf-Mandelbrot distribution, topology is general scale-free networks, running time is 300 s.

In terms of content popularity, popularity α plays an important role in network performance. three improvement

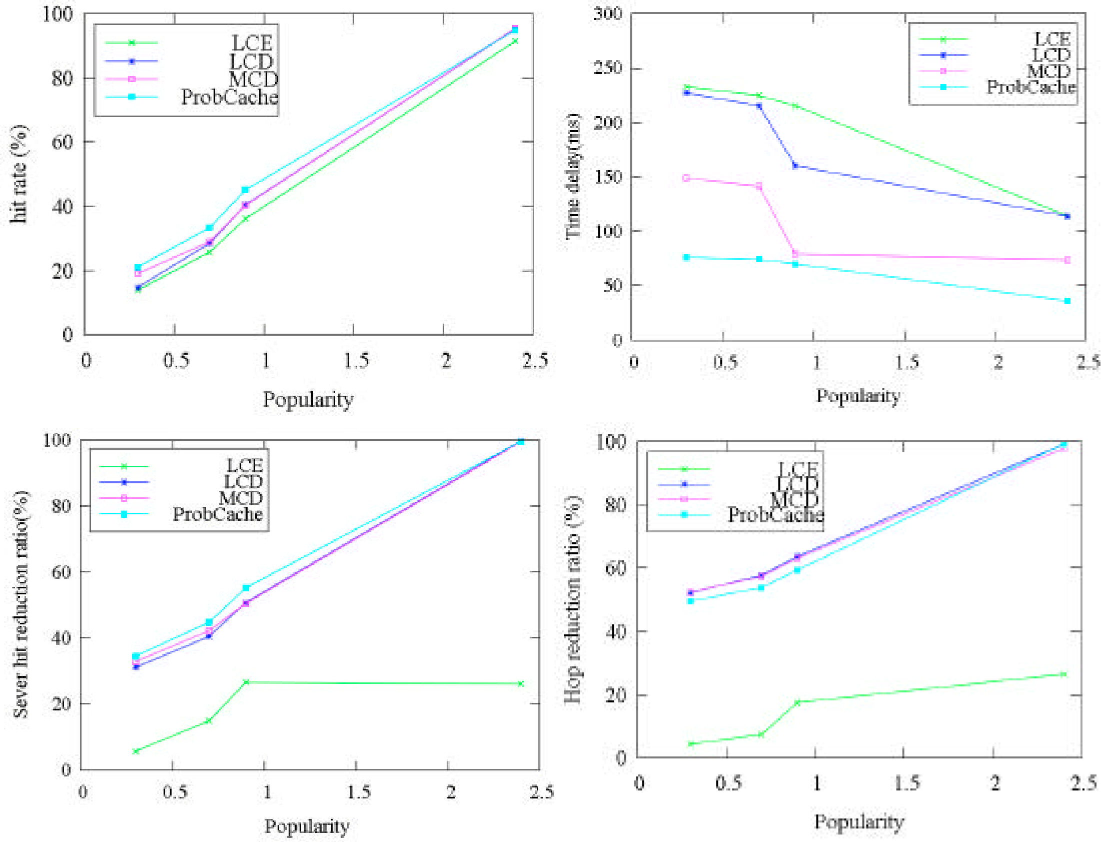

Fig. 3(a-d): Effect of popularity on (a) Hit-rate, (b) Delay, (c) Server hit decrement and (d) Hop decrement

policies, LCD, MCD, ProbCache , have better performance in hit-rate, delay, server hit decrement, hop decrement compared with the default policy LCE.

As shown in Fig. 3a: in terms of hit-rate and server hit decrement, regardless of popularity, ProbCatche is the best; MCD and LCD are similar in condition of low popularity and the LCE is lowest; When popularity is high, LCD is better than MCD and similar to ProbCache, reached 97%, improved by 78%. Network delay is shown in Fig. 3(b): ProbCache is best, followed by MCD, finally is LCE and LCD and popularity is inversely proportional with the network delay. Hop decrement is shown in Fig. 3(d): LCD and MCD is similar when popularity is low and higher than ProbCache; when popularity is high, hop decrement of ProbCache is higher than LCD and MCD; the default caching policy LCE is always lowest compared with improved policies. This is because ProbCache, compared with LCD, MCD and LCE, could deliver requested content to nodes nearer to the requester faster and the network structure in scale-free networks is complicated enough to reduce the in-node cache replacement frequency, so the ProbCache performs better in hit-rate, delay and server hit decrement. But in terms of hop decrement, due to the lack of enough request content concentration when á is low, ProbCache will frequently replace contents in nodes nearer to the requester, thus have lower hop decrement compared with LCD and MCD.

Effect of cache size on cache policy: In environments of different cache size, compare the performance of LCE, LCD, MCD and ProbCache. The cache size of each node is 10, 15, 20, 25, 30, 25, 40, 45 and 50% of content types, respectively. Content distribution follow the Zipf-Mandelbrot distribution, topology structure is scale-free network, running time is 300 s. Other parameters referred in Table 1.

As shown in Fig. 4a-d: When cache capacity is low, ProbCache performs better in hit-rate and server hit decrement than MCD, LCD and LCE; while with high cache capacity, LCD is better than ProbCache and MCD; LCE is better than ProbCache for hit-rate, but always lower than LCD, MCD and ProbCatch in server hit decrement. This is because with the increase of node cache capacity, the network can cache more contents.

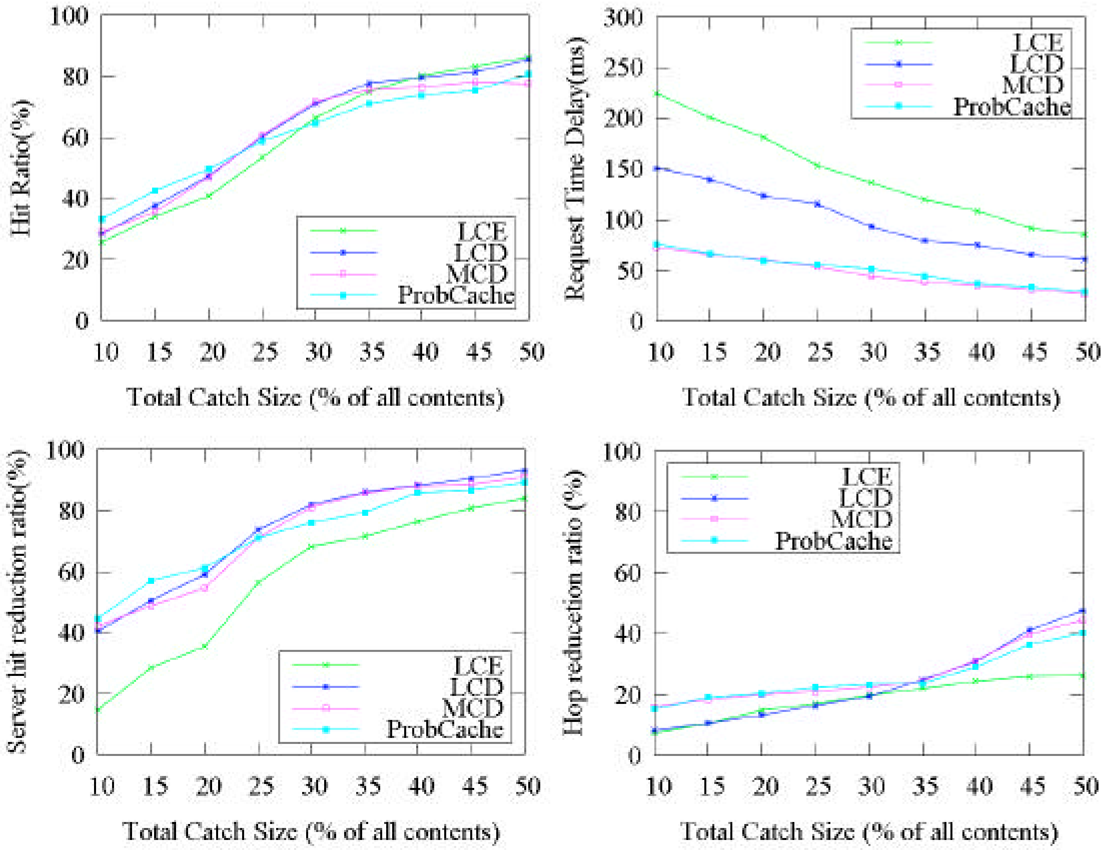

Fig. 4(a-d): Effect of cache size on (a) Hit-rate, (b) Delay, (c) Server hit decrement and (d) Hop decrement

LCD strategy can make the network store multiple copies of same requested content, therefore, with the increase of node cache capacity, LCD is significantly boosted in hit-rate and servers hit decrement and hop decrement. As shown in Fig. 4b, network delay decreases with the increasing of the cache capacity and ProbCache is always optimal, MCD second, LCE and LCD have longest delay.

CONCLUSION

This study studied the multiple network caching policies' performance in Content Centric Network. Through experiments we proved: in scale-free networks, when nodes cache space accounted for the proportion of all content is less than 25%, the caching policy ProbCache has optimal performance in network hit-rate, time delay and the server load decrement and hop decrement. And when the cache space of all content size is bigger than 25%, the comprehensive performance of LCD caching policy has greatly increased.

ACKNOWLEDGMENT

This study is supported by the National Natural Science Foundation of China (Grant No. 61073042, 61003036), the Natural Science Foundation of Heilongjiang Province, China (Grant No. F201121), Harbin Technological Innovation Talent Research Special Foundation Project of Harbin, Heilongjiang Province, China (Grant No. 2012RFQXG097), Postdoctoral Science-research Developmental Foundation of Heilongjiang province, China (Grant No. LBH-Q12134), the Fundamental Research Funds for the Central Universities (Grant No. HEUCF100606), Heilongjiang province project Education Fund (Grant No.12513054).

REFERENCES

Carofiglio, G., M. Gallo, L. Muscariello and D. Perino, 2011. Modeling data transfer in content-centric networking Proceedings of the 23rd International Teletraffic Congress, September 6-9, 2011, San Francisco, USA, pp: 111-118.

Carofiglio, G., V. Gehlen, and D. Perino, 2011. Experimental evaluation of memory management in content-centric networking. Proceedings of the IEEE International Conference on Communications, June 5-9, 2011, Kyoto, Japan, pp:1-6.

Chai, W.K., D. He, I. Psaras and G. Pavlou, 2012. Cache less for more in information-centric networks. Proceedings of the Part I, 11th International IFIP TC 6 Networking Conference, May 21-25, 2012, Springer-Verlag, Berlin, Heidelberg, pp: 27-40.

Li, J., Z. Chen and X. Shi, 2012. ICN system structure and technology research. Chinese J, Netinfo Security, 42: 75-80.

Muscariello, L., G. Carofiglio, and M. Gallo, 2011. Bandwidth and storage sharing performance in information centric networking. Proceedings of the ACM SIGCOMM workshop on Information-centric networking, August 15-19, 2011, Toronto, Canada, pp: 26-31.

Psaras, L., R.G. Clegg, R. Landa, W.K. Chai and G. Pavlou, 2011. Modelling and evaluation of CCN-caching trees. Proceedings of the 10th International IFIP TC 6 Networking Conference, May 9-13, 2011, Valencia, Spain, pp:78-91.

Yang, L., S.W. Ma and X.X. Wang, 2011. Research on the internet architecture of content centric network. J. Infor. Comm. Technol., 56: 66-70.

Zhang, L.X, E. Deborah, and B. Jeffrey, 2010. Named data networking. Technical Report 001. http://named-data.net/techreports.html