

Popularity and gain based caching scheme for information-centric networks

Zhandong Fan, Qingtao Wu*, Mingchuan Zhang and Ruijuan Zheng

Information Engineering College, Henan University of Science and Technology, China

Received: 09-February-2017; Revised: 24-March-2017; Accepted: 27-March-2017

©2017 ACCENTS

Abstract

In information-centric networks (ICN), each node is equipped with a cache and it can improve content access and transmission efficiency. However, the built-in cache capacity is small, and it can't completely store huge amounts of content transmitted. The existing caching schemes lack choices in content placement and balanced distribution, thus, leading to the problem of low cache hit ratio and high user access time delay. To solve the problem, this paper proposes a popularity and gain based caching scheme (PGBCS). It refines the content object from the content file to the chunk, thus achieves fine-grained cache management. The strategy makes use of the feature of content chunk popularity, and takes the factors that affect the overall caching gain into consideration, and realizes the placement and replacement of content chunks through comprehensive measure. The simulation results show that compared with other schemes, this method can effectively promote node's cache hit ratio, reduce user request delay and further raise network service quality.

Keywords

Information-centric networks, Content chunk, Popularity, Caching gain.

1.Introduction

With the rapid development of internet, several of new applications have appeared. The traditional centered on host network architecture has been difficult to adapt to new application requirements, which faced with a series of challenges in mobility support, network scalability and security, etc. Although caching technology in application of content delivery networks (CDN) and peer to peer (P2P) [1] has alleviated the pressure of application requirements to a certain extent. The current caching strategies are aimed at a specific application and the technology with shortcomings of high complexity and costs and can't apply to all applications. In order to solve these problems, researchers have focused attention on research of new Internet architecture [2]. Information-centric networks (ICN) emerges at the right moment and has received extensive research and development, among which; the respective schemes under this background include the content-centric networking (CCN)/ named-data networking (NDN) [3], data-oriented network architecture (DONA) [4], NetInf [5], PDX [6], etc. In ICN, each node is equipped with a built-in cache, and its core idea is that storing content through internal routers of the network, then the user requests forward in the network.

The caches will respond directly if hit along the way, rather than from the content source (server) each time [7].

Due to cache content closer to users, thus can reduce the consumption of network bandwidth, server load and user access delay [8]. It is considering these advantages, this thought is extended to the entire network and all applications in the design of the new Internet.

Caching mechanism is the core of the ICN technology, the pros and cons of a caching system directly affect the efficiency of ICN content distribution, and a reasonable caching policy enables users easier to access, to content from the near cache nodes without from original content server (OCS) every time. The content placement and replacement strategy are two important aspects of caching technology research, however, the major problems of current caching policies include low space utilization rate, and low cache hit ratio and large access latency. In order to solve the above problems, this paper proposes a PGBCS Scheme. The main contributions of this paper are as follows.

- 1) Refining the level of popularity to content chunk. The existing caching strategies are mostly based on content popularity, which is not the same as

*Author for correspondence

content chunk popularity. This article refines the level of popularity to the chunk and realizes fine-grained caching management.

- 2) Based on comprehensive earnings to find the best placement node. Comprehensively measure factors that influence caching gain, finding the best node to cache the content chunk, and maximizing the comprehensive income of nodes.
- 3) The substitution method based on the comparison of the content chunk value. Comparing the value of the content chunk to be replaced with the new arrival content chunk and determining whether it is necessary to cache new content chunk.

2.Related works

In ICN, if it is needed to establish collaboration domain and interactive state information of collaboration between the nodes. ICN caching strategies can be roughly divided into explicit collaboration and implicit cooperation [9]. In explicit collaboration mechanism, a kind of common way is through the interaction access mode between nodes in collaboration domain, the cache state, etc. to improve the caching diversity in the cooperative domain, but the strong dynamic interaction of ICN cannot bear the huge costs.

In implicit cooperative caching mechanism, nodes can achieve cache decision-making with little or less interaction, at the same time; have an advantage in terms of execution speed and complexity. The implicit cooperative caching mechanism can be divided into the following two categories according to whether need to know accurate information about content popularity beforehand.

2.1 Considering potential content popularity caching mechanism

In [10] authors proposed an RCOne mechanism, which stores content objects in a node randomly selected along the way transmission path. ProbCache [11, 12] takes the node position in the data transmission path and cache capacity into account while making probability caching decisions, making the content with greater probability cached in the node closer to the requester. In [13] authors proposed a WAVE scheme which provides caching advice for the subsequent nodes by setting the mark, and stores content chunks of file in the exponential growth way along the transmission path when the number of requests for a unit of the file increases. Caching decisions according to node and importance in the community have been suggested [14, 15]. Such decisions are relatively simple, but lack of accurate

information about content access frequency, and exist low speed of copying the content to the edge, the excessive competition in the edge nodes, the disorder competition of cache space between different transmission paths, etc.

2.2Caching scheme based on content popularity

In [16] authors proposed a caching scheme based caching age, which calculates the caching age of content according to the level of content popularity and the location of the storage nodes, the higher the content popularity, the longer the cache resides. A prob-PD scheme based on the probability of popularity which makes decision of probability caching according to content popularity and the distance from the content source have been proposed [17]. A caching mechanism based on income perception, which uses the filtering effect of request flow on the cache, and achieves collaboration between nodes and cache diversity while maximizing single node earnings have been proposed [18]. A probabilistic storage based heuristic caching method, the content with higher popularity and greater placement earnings can have a greater caching probability have been proposed [19]. Badov et al. [20] proposed a congestion perception and lookup mechanism, which caches the more helpful content to alleviate congestion and considers low bandwidth load of links preferentially when forwarding request messages. These schemes have a better performance in the cache hit ratio and bandwidth saving, however, most of them, assuming content popularity is a known parameter, besides, there is no clear description information that how to obtain and maintain content popularity by the router.

In ICN caching systems, the request arrival process is related to the caching state of upstream nodes [21], so the arrival rate of content requests has a strong dynamic feature, and static offline popularity statistics clearly can't will meet the needs of caching decisions. A direct method is to use a hash table to count the visits of content; however, the storage cost of this method is unbearable [22]. Some methods through the statistics of coarse-grained content access frequency to reduce the complexity, the specific methods include dividing the content into several levels in accordance with content popularity [23, 24] and using the content file popularity instead of content chunk popularity [25], etc. But users may show different degrees of interests towards the content chunks of same content file; these methods ignore the differences which influence the performance of the granular cache management.

Different with the above works, this paper comprehensively measures the main factors which influence the caching gain by means of implicit cooperation and finding out the best placement node to cache content chunks through the calculation of the server or nodes. At the same time, the regular of rise and fall is considered in caching mechanism and using the value of the content chunk comparison approach to decision new content chunk replacement. Besides PGBCS algorithm is simple and has a less cost of computing resources.

3. PGBCS operation mechanism

In our system model, the network is made up of OCS and some routers equipped with built-in cache as well as some users. Among which, only OCS can produce content, and store content permanently. Let $G(V, E)$ be an undirected network with $V = \{v_1, \dots, v_n\}$ nodes and E the set of links. For the sake of simplicity, we assume that v_{i-1} is on the downstream side of v_i , which means v_{i-1} is closer to the user.

This article does not consider the situation of the flood routing, which means the nodes choose the next-hop only according to routing protocols and not broadcast to all ports when receive interest packet.

3.1 The basic principle of PGBCS

In order to make the cache in ICN more efficient, this paper design's content chunk popularity and caching gain based caching scheme. The PGBCS basic idea is based on the value of a content chunk in the cache space is changing dynamically and decaying exponentially with time. At the same time, in order to ensure that the high popularity content chunks have high caching value, the value of content chunks increases exponentially with the requested count, so the value of high popularity content chunks can be far higher than the low popularity content chunks. The occupied space is marked as "idle" when the content chunks value below the threshold value set, however, the content chunks in the free space are not deleted immediately, and will be replaced only when new content chunks arrive, in this way, the content chunks in the free space can continue to provide service for users. When the content chunks of the free space pop up again, so that the value is higher than the threshold set, and then cancel its "idle" tag of the occupied space. Interest request packets will forward in the form of labels, and if hit along the way, then the OCS or router will packet and back to user along Interest path. If there are spare nodes on interest path, then store the content chunk on the spare node closest to the user; if there are no spare nodes, and then select a

node which can maximize caching gain as the best placement node.

3.2 The implementation of PGBCS caching policy

This paper aims to design an in-network caching scheme that can obtain high gain. Symbols are defined in Table 1.

Table 1 Table of notation

Notation	Meaning
O	Content chunk
v_i	The identify of i node
\overline{w}_i	The average value of content chunks cached in v_i
S_i	The occupied space of v_i
r_i	The space of v_i expected to take up
f_i	The free space of v_i
C_i	The overall cache space of v_i
p_i	The usage ratio of v_i
d_i	The distance between v_i and user
D	The total length of Interest path
q_i	The ratio of d_i and D
$M(v_i)$	Caching gain of v_i
w_0	The initial value of content chunk
$m_i(o)$	The number of requests for O in v_i
$W_i(o)$	The value of O in v_i
$W_t(o)$	The value of O at time t

3.2.1 Caching gain measurement

In ICN, if the high popularity content chunks are stored in nodes closer to users, which can reduce forwarding paths of interest packets and data packets, thus can decrease users' access time effectively and improve the user's experience. However, not all content chunks with high popularity can be stored on the node closest to the user due to the cache capacity constraints. Taking the information of nodes which includes the value of content chunks, the free space of nodes and the distance between nodes and users and etc. into account when searching the node along the interest path that can maximum caching gain if cache the content chunk.

Defining a measure indicator $M(v_i)$ for a caching gain of router nodes, the bigger value of $M(v_i)$ shows that placing the new content chunk on the node will make overall nodes gain maximal on the interest path. The main factors influencing the value of $M(v_i)$ are: The smaller the average value \overline{w}_i of the content cached by the v_i , indicate that the higher possibility content

chunks in the node will be replaced and the greater the value of $M(v_i)$. The smaller the cache usage rate p_i of v_i , shows that the greater the free space of the node, which can make full use of the cache space if cache the new content chunk on the node, and the greater the value of $M(v_i)$. The smaller the distance ratio q_i of v_i , shows that the closer distance between the node and user, which can reduce the user's access delay if cache the new content chunk on the node, and the greater the value of $M(v_i)$. Comprehensive calculate $\overline{w_i}$, p_i , q_i and obtain $M(v_i)$.

Step 1: calculating the average value of the content chunks cached in v_i .

$$\overline{w_i} = \frac{1}{n} \sum_{j=1}^n (o_j)$$

Among them, $\overline{w_i}$ indicates the average value of content chunks cached in v_i , n indicates the number of content chunks cached, and o_j indicates content chunk j .

Step 2: Computing the utilization rate of v_i .

$$p_i = \frac{S_i + r_i}{C_i}$$

Among them, S_i indicates the amount (not including the content chunk in free space) of space occupied of v_i , r_i indicates the amount of space that expected to be occupied, the value of r_i is equal to the number of Interest packets forwarded across v_i which not yet return data packets. The maximum r_i equals $C_i - S_i$ (the size of the free space of v_i), thus it can guarantee $0 < p_i < 1$.

Step 3: The distance factor.

$$q_i = \frac{d_i}{D}$$

Among them, d_i indicates the distance (the number of hops) between v_i and user along the interest path and D indicates the total length of Interest path.

The calculation formula of v_i caching gain index.

$$M(v_i) = \frac{1}{\overline{w_i}} \cdot \frac{1}{p_i} \cdot \frac{1}{q_i}$$

When v_i has free space, then OCS or router nodes choose the best cache node according to f_i in Interest packet, at this time, does not calculate the value of $M(v_i)$, so there is no meaningless situation that denominator equals 0.

3.2.2 The calculation of cache value

If the Interest packet hit at OCS, then set an initial value w_0 for the content chunk (the value is greater than the threshold), and if the requested content chunk comes from router node, then the initial value of the content chunk is the value that cached in the node. Assuming that the distance between v_i and OCS is 1, and v_{i-1} located on the downstream side of v_i , so the value of content chunk o when it returned to v_{i-1} is calculated as follows.

$$\begin{cases} w_i(o) = w_0 \cdot a^{m_i(o)-1} \\ W_{i-1}(o) = w_i(o) \cdot a^{m_{i-1}(o)-1} \end{cases}$$

Among them, $m_i(o)$ indicates the requested times of o in v_i , and in order to avoid repeating calculation requests for the first time then minus 1. $a(a > 1)$ is the base of the index, and the greater its value shows that the faster the value increases of content chunk, we assume that $a=2$.

The value growth rule of the content chunks in v_i node is.

$$W(o) = w_i(o) \cdot a$$

Among them, $W(o)$ is the value of o after request, $w_i(o)$ is the value of o before request, and the content chunk o will growth $a(a > 1)$ times after every request, and the greater a shows that the faster the content chunk value increases, we assume that $a=2$.

The value attenuation rule of content chunks in v_i node is.

$$W_t(o) = w_i(o) \cdot b^{\frac{t-t_0}{\tau}}$$

Among them, $W_t(o)$ is the value of o at time t , $W_i(o)$ is the value before attenuation, t_0 indicates the last updated attenuation time, $t-t_0$ indicates the distance from the last update time, and τ is a constant. $b(0 < b < 1)$ is the base number, the smaller its value shows that the faster attenuation of the content chunk, we assume that $b=0.5$, $\tau=1$.

3.3 The main algorithm

PGBCS caching policy is based on single Interest packet request, this paper will give the strategy implementation with the process of Interest packet and data packet.

When user sends a request, if the Interest packet does not hit and there is no PIT entry for content request at the node, then the 4 group $(v_i, d_i, f_i, M(v_i))$ will be recorded and forwarded to the next node. If the Interest packet hit at the node, then take out the set of 4 groups and judge whether there is any nodes with $f_i > 0$ along Interest path. Finding out v_i that closest to the user with $f_i > 0$. If exist, then caching o on the v_i directly in the process of data packet back to user, and at the same time replacing the minimum value of the content chunk; If not exist, then choose v_i with the maximum $M(v_i)$ as the best cache node along Interest path. When the data packet back to the node, comparing with the content chunk with the minimum value, if the value of the content chunk not less than the minimum value, then replacing the minimum value content chunk; if the value of the content chunk less than the minimum value which shows the content chunk has low popularity, then giving up cache the content chunk.

If the Interest packet hits at OCS after forwarding, then OCS will pack content chunk into packet and back to user along Interest path. First of all, OCS will assign an initial value w_0 to the content chunk o and the value of o will increase with the number of request times increases. Then, the OCS determines the location of the content chunk o , the process is similar to the above process of when Interest packet hit at the node, so we will not repeat it here. PGBCS strategy is only associated with the value of f_i and $M(v_i)$ of the node, so the node that Interest packet hit can judge independently and there is no additional information interaction between nodes.

In order to illustrate the PGBCS strategy implementation process more clearly, the pseudo code of PGBCS algorithm is given in Table 2.

Table 2 PGBCS algorithm

1. Initialization
2. if exist nodes with $f_i > 0$ along Interest path
3. Case1: exist
4. IF (the number of nodes is equal to 1)

5. caching content chunk o in the node and replacing the content chunk with the lowest value cached in free space;
6. ELSE
7. caching content chunk o in the node closest to the user and replacing the content chunk with the lowest value cached in free space;
8. End IF;
9. Case2: not exist
10. selecting the node with the largest $M(v_i)$ as the best cache node;
11. IF (the value of o not less than the lowest value of content chunk in v_i)
12. caching o in the node and replacing the content chunk with the lowest value;
13. ELSE
14. discarding the content chunk o and giving up caching;
15. End IF;

4. Experiment simulation and analysis

In order to verify the performance of PGBCS proposed in this paper, we carried out the simulation and performance evaluation based on ndnSIM [26] simulator and used Matlab software draw graphics with simulation data. The main performance parameters considered include cache hit ratio, the hop count and content diversity ratio.

4.1 The setup and method of simulation

Topological structure is based on any intention structure, as shown in Figure 1; the experimental network topology consists of 36 nodes.

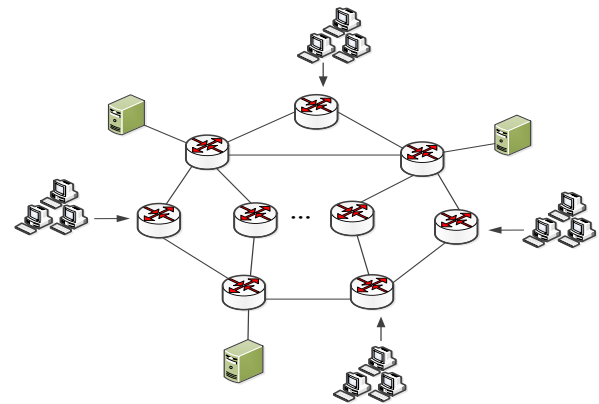


Figure 1 Simulation topology

At the same time, we assume that the number of the content is 10000, each content has the same size 10 Mbytes, and has been divided into $k=10$ chunks; Each router buffer space is 100 Mbytes, which can cache 100 content chunks. In addition, each user will

send 1000 requests for content to connect routers in the simulation process, besides the arrival of the requests accord with Poisson distribution and the user requests for content accord with Zipf distribution [27], in which, we assume $0.5 \leq \alpha \leq 1$ and $\alpha = 0.7$. We selected representative strategy as PGBCS performance comparison objects: one is the traditional leave copy everywhere (LCE) strategy, another is the strategy based on the betweenness strategy. This paper used multiple performance parameters that PGBCS compared with LCE and Betw, which includes cache hit ratio, hop count and content diversity ratio. In addition, we also studied the effect of various network parameters on the performance, such as cache size, the number of content and Zipf parameters. The main parameters and default values of this paper are given in Table 3.

Table 3 Experimental parameters

Parameter	Default value	Range
Number of nodes	36	
Number of users	1000	
Number of content	10000	5000~20000
Content size (MB)	An average of 10	
Node cache size (MB)	100	50~200
Access pattern	Zipf: $\alpha = 0.7$	0.5~1

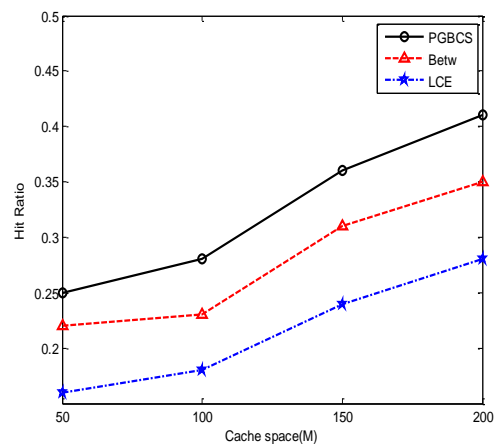
4.2 The result of the experiment

The experiment results are given in this section, in order to observe the network performance is affected by some parameters, we only make one parameter change and other parameters remain unchanged, its default value is shown in Table 3.

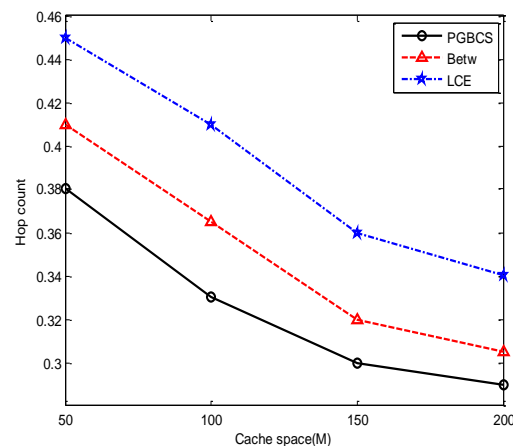
4.2.1 The influence of the cache size

In this section, we first observed the effect of cache size on system performance. Figure 2 shows the system performance changes over the cache size, among them, cache size have been increased from 50MB to 200MB.

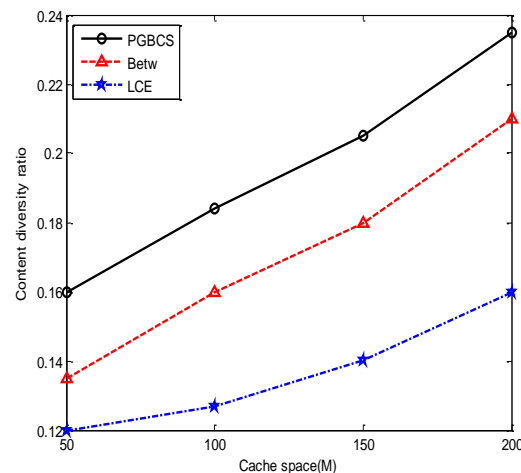
As expected, from Figure 2 (a) you can see that the cache hit ratio of these three mechanisms increases with the increase of the cache size, and in this process, the performance of PGBCS has been always better than LCE and Betw. Figure 2 (b) shows that the hops user access to the content of these three mechanisms decreases with the increase of the cache size. Because of the hit ratio has improved the hop count also fell naturally. However, the hop count of PGBCS is always lower than LCE and Betw. As is shown in Figure 2 (c) that the CDR of these three mechanisms increases with the increase of the cache size, and PGBCS is always higher than LCE and Betw.



(a) Hit ratio changes over the cache size



(b) Hop count changes over the cache size

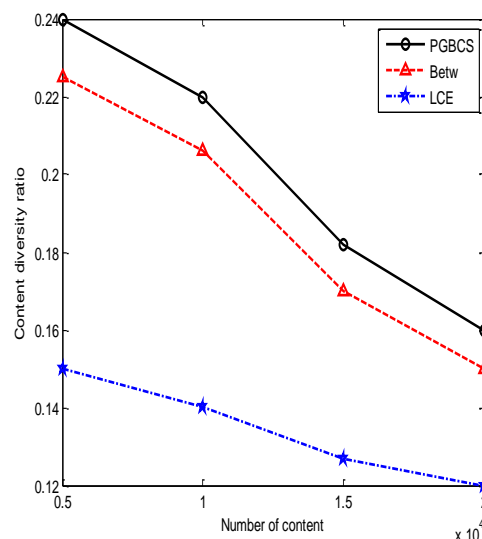


(c) Content diversity ratio changes over the cache size

Figure 2 The influence of the cache size

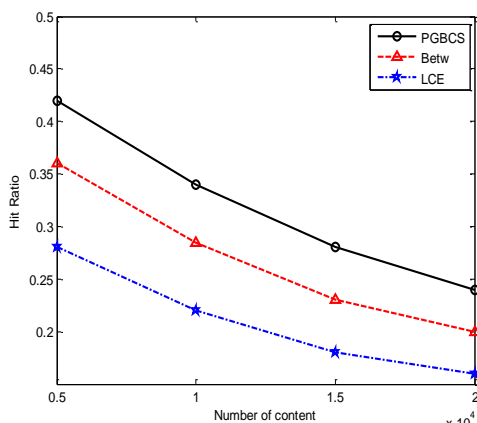
4.2.2 The influence of content quantity

Figure 3 shows the situation that system performance changes over content quantity, the number of content changes from 5000 to 20000 in a wide range. As shown in Figure 3(a) that the cache hit ratio of these three mechanisms reduces significantly with the increase of content amount. This is mainly because the cache size hasn't changed but the content quantity increased, thus, the cache will become increasingly scarce and the probability of Interest packet be responded is reduced naturally. However, compared with LCE and Betw, PGBCS always shows an obvious advantage. In Figure 3(b), as expected, the hop count of these three mechanisms increases with content amount increases, but PGBCS is always lower than LCE and Betw. As shown in Figure 3(c) that the CDR of these three mechanisms decreases with the content amount increases and PGBCS is always better than LCE and Betw in various cases. However, the relative advantage of PGBCS is gradually shrinking with the increase of content amount.

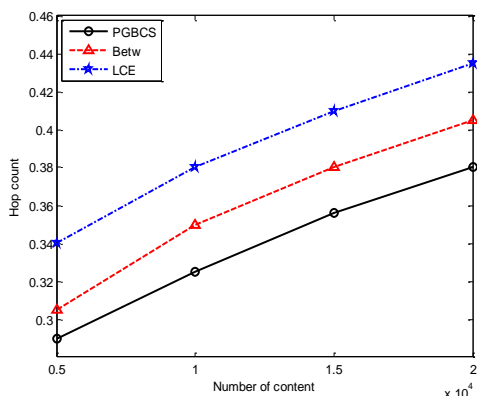


(c) Content diversity ratio changes over content quantity

Figure 3 The influence of content quantity



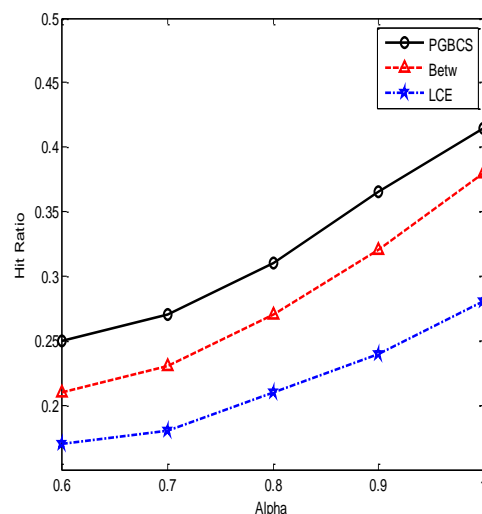
(a) Hit ratio changes over content quantity



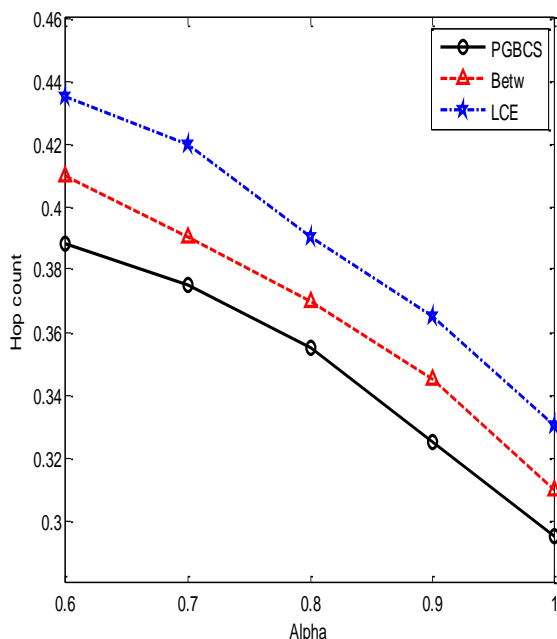
(b) Hop count changes over content quantity

4.2.3 The influence of Zipf parameters (α)

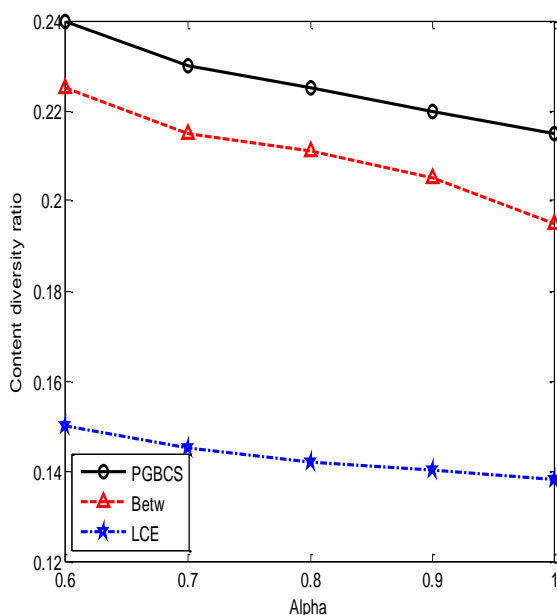
Now, everybody has a widespread belief that user preferences (i.e., access mode) for the content follows Zipf distribution. Different applications has different distributed parameters, the greater Zipf parameter (α) the more concentrated the user preferences. In this section, we study the influence of user preference on caching mechanism performance, more accurately; we'd like to know the caching mechanism performance for different applications. Figure 4 shows the results of the experiment, in which, α varies from 0.5 to 1.



(a) Hit ratio changes over parameter α



(b) Hop count changes over parameter α



(c) Content diversity changes over parameter α

Figure 4 The influence of Zipf parameter (α)

Figure 4(a) clearly shows that the cache hit ratio of these three mechanisms increases with the increase of α , the performance improves with the strengthen of local time.

This is because these three mechanisms take advantage of the local time of content in place or replacement strategy, which is further proved that the importance of using local time in designing caching mechanism. Figure 4(b) shows that the hop count of these three mechanisms decreases with the increase of α , but PGBCS is better than LCE and Betw. Figure 4(c) shows that the CDR of PGBCS and Betw decreases slowly with the increase of α , while the CDR of LCE almost kept unchanged, but PGBCS is always higher than LCE and Betw. The above experimental results show that PGBCS performance is better than LCE and Betw under various experimental conditions which include cache hit ratio, hop count and content diversity ratio.

5. Conclusion

In this paper, we propose a content chunk popularity and gain based caching scheme. It refines content object to the chunk, and realizes fine-grained caching decisions. The strategy makes full use of content chunk popularity, and takes the access frequency of content chunk as a measure of popularity. The popularity of a content chunk increases exponentially with the requested count, at the same time, it decays exponentially over time. PGBCS also takes node's overall caching gain into account, the new arrival content chunk is cached in the node which can maximize the overall caching gain. The simulation results show that the scheme can improve the cache hit ratio of routing nodes and reduce users' requests time delay. In the future, we will study the correlation between different content chunks of a same content object, and apply it to the PGBCS to improve the performance of our algorithm.

Acknowledgment

This work is partially supported by the National Natural Science Foundation of China (NSFC) under Grants no. U1404611, no. U1604155, and no. 61370221, in part by the Program for Science & Technology Innovation Talents in the University of Henan Province under Grants no. 16HASTIT035 and no. 14HASTIT045, in part by Program for Science & Technology Innovative Research Team in University of Henan Province under Grant no. 14IRTSTHN021, and in part by Henan Science and Technology Innovation Outstanding Talent under Grant no. 164200510007.

Conflicts of interest

The authors have no conflicts of interest to declare.

References

- [1] Rhea S, Godfrey B, Karp B, Kubiawicz J, Ratnasamy S, Shenker S, et al. Open DHT: a public DHT service and its uses. In ACM SIGCOMM computer communication review 2005 (pp. 73-84). ACM.
- [2] Rexford J, Dovrolis C. Future internet architecture: clean-slate versus evolutionary research. *Communications of the ACM*. 2010; 53(9):36-40.
- [3] Jacobson V, Smetters DK, Thornton JD, Plass MF, Briggs NH, Braynard RL. Networking named content. In proceedings of the 5th international conference on emerging networking experiments and technologies 2009 (pp. 1-12). ACM.
- [4] Koponen T, Chawla M, Chun BG, Ermolinskiy A, Kim KH, Shenker S, et al. A data-oriented (and beyond) network architecture. *ACM SIGCOMM Computer Communication Review*. 2007; 37(4):181-92.
- [5] Dannewitz C, Golic J, Ohlman B, Ahlgren B. Secure naming for a network of information. In INFOCOM IEEE conference on computer communications workshops 2010 (pp. 1-6). IEEE.
- [6] Fotiou N, Nikander P, Trossen D, Polyzos GC. Developing information networking further: from PSIRP to PURSUIT. In international conference on broadband communications, networks and systems 2010 (pp. 1-13). Springer Berlin Heidelberg.
- [7] Amble MM, Parag P, Shakkottai S, Ying L. Content-aware caching and traffic management in content distribution networks 2011(pp. 2858-66). IEEE.
- [8] Wang J. A survey of web caching schemes for the internet. *ACM SIGCOMM Computer Communication Review*. 1999; 29(5):36-46.
- [9] Guo-qiang Z, Yang L, Tao L, Hui T. Survey of in-network caching techniques in information-centric networks. *Ruan Jian Xue Bao/Journal of Software*. 2014; 25:154-75.
- [10] Eum S, Nakauchi K, Murata M, Shoji Y, Nishinaga N. CATT: potential based routing with content caching for ICN. In proceedings of the second edition of the ICN workshop on information-centric networking 2012 (pp. 49-54). ACM.
- [11] Psaras I, Chai WK, Pavlou G. Probabilistic in-network caching for information-centric networks. In proceedings of the second edition of the ICN workshop on information-centric networking 2012 (pp. 55-60). ACM.
- [12] Psaras I, Chai WK, Pavlou G. In-network cache management and resource allocation for information-centric networks. *IEEE Transactions on Parallel and Distributed Systems*. 2014; 25(11):2920-31.
- [13] Cho K, Lee M, Park K, Kwon TT, Choi Y, Pack S. Wave: popularity-based and collaborative in-network caching for content-oriented networks. In IEEE conference on computer communications workshops 2012 (pp. 316-21). IEEE.
- [14] Chai WK, He D, Psaras I, Pavlou G. Cache “less for more” in information-centric networks (extended version). *Computer Communications*. 2013; 36(7):758-70.
- [15] Cai J, Yu SZ, Liu WX. Caching strategy based on node’s importance to community in information-centric networks. *Sūrikaiseikikenkyūsho Kōkyūroku*. 2015.
- [16] Ming Z, Xu M, Wang D. Age-based cooperative caching in information-centric networking. In international conference on computer communication and networks 2014 (pp. 1-8). IEEE.
- [17] Ioannou A, Weber S. Towards on-path caching alternatives in information-centric networks. In IEEE conference on local computer networks 2014 (pp. 362-5). IEEE.
- [18] Long CH, Hongbo TA, Xingguo LU, Yi BA, Zhang Z. Gain-aware caching scheme based on popularity monitoring in information-centric networking. *IEICE Transactions on Communications*. 2016; 99(11):2351-60.
- [19] Wu HB, Li J, Zhi J. Probability-based heuristic content placement method for ICN caching. *Journal on Communications*. 2016; 37(5):62-72.
- [20] Badov M, Seetharam A, Kurose J, Firoiu V, Nanda S. Congestion-aware caching and search in information-centric networks. In proceedings of the international conference on information-centric networking 2014 (pp. 37-46). ACM.
- [21] Melazzi NB, Bianchi G, Caponi A, Detti A. A general, tractable and accurate model for a cascade of LRU caches. *IEEE Communications Letters*. 2014; 18(5):877-80.
- [22] Dai H, Wang Y, Wu H, Lu J, Liu B. Towards line-speed and accurate on-line popularity monitoring on NDN routers. In IEEE international symposium of quality of service 2014 (pp. 178-87). IEEE.
- [23] Kim Y, Yeom I. Performance analysis of in-network caching for content-centric networking. *Computer Networks*. 2013; 57(13):2465-82.
- [24] Zeng Y, Jin M, Luo H. LICA: A segment-popularity based caching scheme in ICN. *Acta Electronica Sinica*. 2016; 44(2):358-64.
- [25] Lanlan R, Hao P, Haoqiu H, Xuesong Q, Ruichang S. Popularity and centrality based selective caching scheme for information-centric networks. *Journal of Electronics & Information Technology*. 2016; 38(2):325-31.
- [26] Afanasyev A, Moiseenko I, Zhang L. ndnSIM: NDN simulator for NS-3. University of California, Los Angeles, Technical Report 2012.
- [27] Breslau L, Cao P, Fan L, Phillips G, Shenker S. Web caching and Zipf-like distributions: evidence and implications. In annual joint conference of the IEEE computer and communications societies 1999 (pp. 126-34). IEEE.



ZhanDong Fan received the bachelor degree in computer science and technology from Henan University of Science and Technology, Luoyang, Henan province, China, in 2014, and he is currently pursuing the master's degree in the computer application technology, Henan University of Science and Technology, Luoyang, Henan province, China. His research interests include cloud computing, Internet of things and information-centric Network.
Email: 1612847595@qq.com



Qingtao Wu was born in Jiangsu Province, PRC in March 1975. Qingtao Wu studied in East China University of Science and Technology (Shanghai, PRC) from March 2003 to March 2006, majored in computer application and earned a Doctor of Engineering Degree in three years' time. He works as a Professor in Henan University of Science and Technology. His research interests include component technology, computer security and future Internet security.



Mingchuan Zhang was born in Henan Province, PRC on May 1977. Mingchuan Zhang studied at Beijing University of Posts and Telecommunications (Beijing, PRC) from September 2011 to July 2014, majored in Communication and information system and earned a Doctor of Engineering Degree in three years' time. He works as an Associate Professor at Henan University of Science and Technology. His research interests include bio-inspired networks, Internet of Things, future Internet and computer security.



Ruijuan Zheng was born in Henan Province, PRC in March 1980. Ruijuan Zheng studied in Harbin Engineering University Technology (Harbin, PRC) from March 2005 to March 2008, majored in computer application and earned a Doctor of Engineering Degree in three years' time. She works as an Associate Professor at Henan University of Science and Technology from March 2008. Her research interests include bio-inspired networks, Internet of Things, and computer security.