

On Performance of Cache Policies in Named Data Networking*

Jianhua Ran¹, Na Lv², Ding Zhang², Yuanyuan Ma², Zhenyong Xie²

¹School of Resource and Environmental Science & Key Laboratory of Geographic Information System of the Ministry of Education, Wuhan University, Wuhan, China

²School of Information and Electronic, Beijing Institute of Technology, Beijing, China
{jzgap & mayuannihao2007 & xiezhengyong07}@163.com, lvnawaiting@gmail.com

Abstract - Named Data Network (NDN) is gaining increasingly concerns, as an important direction of the future Internet architecture research centered on content. Content caching has played a key role in NDN. Existing cache replacement policies like Least Frequently Used (LFU) and Least Recently Used (LRU) have failed to make full use of the popularity of contents, which leads to a low cache efficiency in the dynamic network. In order to make the best use of content popularity in the cache strategy, this paper has proposed a cache replacement policy based on content popularity (CCP), and designed the data structure format and replacement algorithm. For fully studying and analyzing the performance of different cache policies in NDN in terms of network throughput, server load and cache hit ratio, we have done a lot of simulations to show how they will improve the network. The simulation results show that our proposed CCP can significantly decrease the server load with a higher cache hit ratio and increase the network capacity at the same time compared with LRU and LFU. And the average throughput is reduced significantly by nearly 47% in comparison to that of the case without in-network caching. Moreover, it also shows the performance under different sizes of content store. The effectiveness of the CCP strategy is proved during the simulation.

Index Terms - Information-Centric Networking, Named-Data Networking, Cache Policy

I. Introduction

As a future Internet architecture proposal, Information-Centric Networking(ICN) intends to motivate the architectural transition from today's host-centric Internet architecture to information-centric in order to disseminate efficiently and flexibly the enormous information generated by a variety of applications. In this research direction, many approaches have been proposed such as PSIRP [1], NetInf [2, 3], PURSUIT [4], CCN [5], DONA [6] and NDN [7]. NDN [8, 9] is an ongoing research effort that aims to move the Internet into the future with a content-centric design that is capable of efficient content distribution and seamless mobility support. In this paper, we strive to study the impact of various cache policies on overall performance of NDN.

Here, we extract the NDN structuring architectural properties and assumptions as the following. We refer to the prefix as the content identifier which names a content object uniquely (e.g. a video or a named single packet). A request is sent to get a specific content object with the prefix specified in its packet. The request is routed to the original content without

awareness of any cached copy in routers by some routing mechanism (e.g. OSPF). The goal of this paper is to study the performance of a variety of cache policies, e.g. Least Recently Used (LRU), Least Frequently Used (LFU) etc., without considering the bandwidth consumption and propagation delay. Our work mainly focuses on the network performance benefit brought by in-networking caching and studying the impact of employing different cache policies over NDN. Our contributions are indicated as follows:

- (1) We design a cache policy based on content popularity (CCP), which takes into account the popularity factor and add a new data structure called Content Popularity Table (CPT) on the basis of retaining Content Store (CS), Pending Interest Table (PIT) and Forwarding Interest Base (FIB) in NDN.
- (2) With metrics of cache hit ratio, server load and the average network throughput, we find by extensive simulations that the performance benefit brought by CCP is much higher in comparison to LFU and LRU.

The rest of this paper is organized as follows. We provide an overview of NDN architecture in section 2. In section 3, we review the conventional cache policies and existing problems in NDN. Subsequently, in section 4, we introduce our proposed cache policy CCP, which adds a new data structure called CPT on the basis of retaining CS, PIT and FIB in NDN. In section 5, we evaluate our analysis findings and the performance of various cache policies with extensive simulations based on the open-source ndnSIM package, which implements NDN protocol stack for NS-3 network simulator. We conclude in section 6.

II. NDN Overview

In this section, we briefly go over a few basic concepts of NDN that are essential to our work (refer to [10], [11], [12], [13] for more details). All communications in NDN are performed using two distinct types of packets: Interest and Content. Both types of packets carry a name, which uniquely identifies a piece of content that can be carried in one Content packet. Content names in NDN are hierarchically structured and an example name for the first segment of a campus video would like: "/campus/videos/0x23jkA/0".

* This work is partially supported by National Key Technologies R&D Program of China Grant #2011BAH05B08.

NDN router maintains three major data structures: PIT, FIB and CS, which are indicated as follows:

- (1) *PIT*: It holds and keeps track of all “not yet satisfied” Interests that have been sent upstream towards potential data sources so that returned Content can be sent downstream to its requester(s). Each PIT entry contains one or multiple incoming and outgoing physical interfaces; multiple incoming interfaces indicate the same data is requested from multiple downstream users; multiple outgoing interfaces indicate the same Interest is forwarded along multiple paths.
- (2) *FIB*: It maps name prefixes to one or multiple physical network interfaces, specifying directions where Interests can be forwarded.
- (3) *CS*: It temporarily buffers Content packets that pass through this router, allowing efficient data retrieval by different consumers.

Communications in NDN are driven by data consumers. Each piece of content is cryptographically associated to a data name. To retrieve data, a consumer sends out an Interest packet, which carries a name that identifies the desired data. A router remembers the interface from which the Interest comes in, and then forwards the Interest packet by looking up the name in its FIB, which is populated by routing protocols that propagate name prefixes instead of IP prefixes. Each received Interest is kept in a PIT. If additional Interest packets for the same data name are received, the router simply records the arrival interfaces in the PIT entry for the name. Once the Interest packet reaches a node with the requested data, a data packet is sent back. A data packet carries the name and the actual data, along with a signature created by the original data producer that binds the two together.

When a router sees a data packet, it finds the matching entry in its PIT and forwards the packet to the interfaces where the corresponding Interests come from and removes the PIT entry. As a result of the Interest state that have been set up at the intermediate routers, the data packet traces the reverse paths back to all the data consumers in a multicast fashion. Each Interest is kept in the PIT until the desired data is retrieved, or until its time-to-live period expires. It is the data consumer’s responsibility to re-express an Interest when it expires, assuming the Content is still desired then. For simplicity, we assume routers keep Interests in PIT for one second in our simulations.

III. CACHE POLICIES and PROBLEMS in NDN

Now in NDN, LRU and LFU are most generally used.

LRU is an access-time-pattern-based replacement policy with using recent information to increase cache hit ratio, but the content popularity is not considered. i.e., the content with a persistent and a time interval request will most likely be requested by users again. But according to LRU strategy, it may be replaced by the recently requested content, even if less people will be interested in the new content.

LFU is also an access-time-pattern-based replacement policy with using frequency information to increase cache hit

ratio under keeping track of access frequency of a data item. However, it is not suitable for dynamic environments. i.e., some temporarily popular contents may be stored in the cache forever, even if they are no longer used, until some new more popular contents with a higher total hits occur. This may cause the cache to be occupied by data which is popular previously but recently rarely used. So the new arrival content can’t be saved.

IV. A cache replacement strategy based on content popularity

Through above introduction and analysis of cache strategies, we can conclude that in order to meet dynamic changes of user’s interests, a cache replacement strategy must follow the following two points:

- (1) Be able to adapt to rapid changes in the dynamic network transmission mode.
- (2) Can’t be affected by the contents of the temporary alternate.

To meet the aforementioned two demands, we propose a cache replacement policy based on content popularity (CCP). The calculation method of content popularity and CCP strategy will be described in details below.

A. Calculation Method of Content Popularity

CCP adds a new data structure CPT on the basis of retaining CS, PIT and FIB in NDN. CPT is used to save cache hits and content popularity. A detailed table about CPT is shown as Table I.

CPT holds all the information about the content popularity stored in CS, including content name, cache hit, previous and current popularity. When a counting cycle ends, we can get the current content popularity ranking through current cache hit and previous popularity. Then we can determine the cache replacement strategy based on the popularity ranking.

The computation formula of content popularity is described as follows:

$$P[i+1] = \frac{N[i] * \alpha + P[i]}{\alpha + 1} \quad (1)$$

$$\alpha = 1 + c * T \quad (2)$$

As in (1) and (2), $P[i]$ and $N[i]$ indicate the cached content popularity and the content cache hits in current counting cycle, respectively. α ($\alpha > 1$) is the weight coefficient of content popularity. From (2), we can see a positive correlation relationship between α and counting cycle T . c indicates the proportional coefficient of α and T . In this paper, we set $c = 0.5, T = 8$.

Through (1) to calculate the current content popularity, not only considers the influence between previous and current popularity, moreover, it highlights the influence of the latest network hot pot. Through expanding (1), we can get (3) below. We clearly see that as time is going, the previous hit number

has much smaller influence on the content popularity, which much better reflects the content popularity of current network.

$$P[i+1] = \frac{N[i]*\alpha + P[i]}{\alpha+1} \quad (3)$$

$$= \frac{N[i]*\alpha}{\alpha+1} + \frac{N[i-1]*\alpha}{(\alpha+1)^2} + \frac{N[i-2]*\alpha}{(\alpha+1)^3} + \dots$$

Then we give an example to illustrate the application of this method in our cache policy. In Table I, it records three content popularities. We can see the second content has a much higher previous popularity than the first one, but with a lower current cache hit. In this case, we can get the current popularity of the first one is much better than the second one.

TABLE I Content Popularity Table

Content name	Previous popularity	Current cache hit	Current popularity
/campus/images/v1.mpg	12.08	20	18.68
/campus/images/v2.mpg	21.28	15	16.05
/campus/music/m1.mp3	22.14	30	28.69

When $\alpha = 5$, we can get the current popularity is 18.68. Then the latest content with a higher popularity can store in the cache and non-latest content with a lower popularity will be replaced when a new content arrives.

B. Replacement Strategy

Content routing determines the state of cache replacement strategy according to the size of remaining space of current repository. When a content router receives a data packet which hasn't been stored, it needs to calculate the difference between the threshold value of the cache size and the number of cached contents to get the remaining size of cache space. If a data packet is greater than the rest of the space, the replacement will happen. On the basis of CCP, the content with minimum popularity will be replaced and its record in CPT will be deleted at the same time. Then the new arrival content will be cached in CS and set up its own record in CPT at the same time.

V. Simulation And Evaluation

In this section, we present an in-depth evaluation study, aiming to quantify the effectiveness of proposed CCP compared with other strategies. We use the open-source ndnSIM package, which implements NDN protocol stack for NS-3 network simulator (<http://www.nsnam.org/>), to run simulations. We extend ndnSIM with our proposed CCP and evaluate the effectiveness of each replacement strategy independently, including LRU, LFU and CCP.

The metrics we choose to quantify the performance are the average network throughput, cache hit ratio and server load. In our experiments, we assumed that users express Interests at constant average rates with randomized time gap between two consecutive Interests, where the random number for the gap follows a uniform distribution.

In our simulation, there are 20 content routers and users are distributed at the edge of the network. And there is only one server to satisfy the requested Interests. Fig. 1 just shows a part of our network topology. In our experiments, there are 200 different content objects in the network. The cache capability of each node is defined as the percentage of cached capacity and the total capacity, which is ranging from 20% to 60%. For example, if the percentage is 20%, it means that if there are 200 different content objects, then each node can cache 40 content objects. In the real network, the number of cached contents in each node is limited. So we hope to observe the cache efficiency by changing the cache size. At the same time, we define the simulating time is 300s.

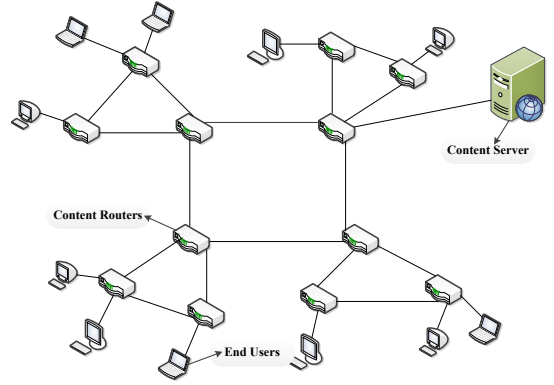


Fig. 1 The topology of the network.

For the results reported below, we assume a backhaul bandwidth of 100Mbps, and the cache size varies between 20 to 400Kbits.

Fig. 2 shows the performance of different cache policies in terms of cache hit ratio achieved for a given topology with different cache sizes. It is evident that CCP performs significantly better than LRU and LFU for all cache sizes. For example, when the cache size is 200Kbits, the cache hit ratio of CCP is approximately 0.6, compared with LRU and LFU whose cache hit ratio achieved are approximately 0.5 and 0.3 respectively.

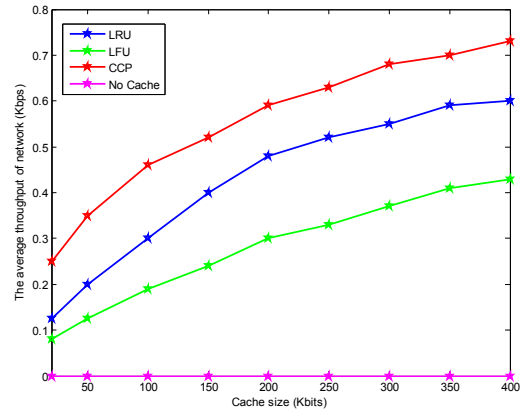


Fig. 2 Cache hit ratio vs. Cache size.

Fig. 3 shows the performance of different cache policies in terms of average network throughput with different cache sizes. It is obvious that CCP performs better than both LRU and LFU. And the average throughput is reduced significantly by nearly 47% in comparison to that of the case without in-network caching.

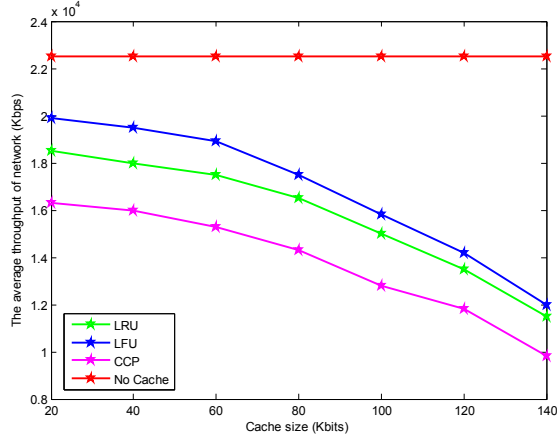


Fig. 3 The average throughput of the network vs. Cache size.

Fig. 4 shows the performance of different cache policies in terms of average server load with different percentages of cache capability. We can see the server can reach a lower load with CCP than with other cache policies. When the percentage is 20%, the server load is decreased by approximately 42.14% compared with LFU. And the effectiveness is more obvious when the cache capability is smaller.

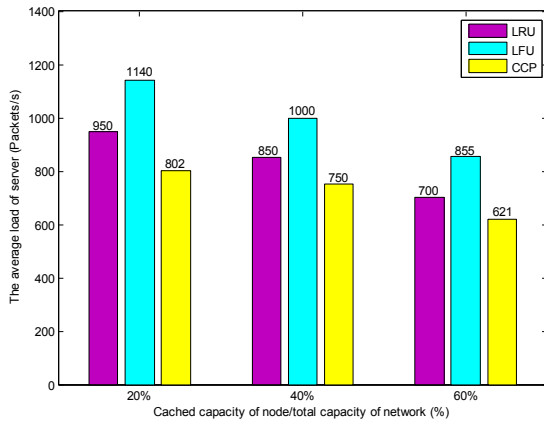


Fig. 4 The average load of the server vs. Cache capability.

VI. Conclusion

NDN being a newly proposed future Internet architecture, it's important to address the significance of in-network caching in NDN. In this paper, we demonstrated the effectiveness of proposed cache policy CCP, compared with the conventional policies like LRU and LFU. Our simulation results show that CCP can significantly decrease the server load with a higher cache hit ratio and increase the network capacity at the same time. And the average throughput is reduced significantly by nearly 47% in comparison to that of the case without in-network caching. In the future, we plan to design more intelligent cache policies and more complicated network scenarios with real services will be considered.

Acknowledgment

This research is supported by a grant from the National Key Technologies R&D Program of China.

Reference

- [1] P. Jokela, A. Zahemszky, C. E. Rothenberg, S. Arianfar and P. Nikander, LIPSIN: "Line Speed Publish/Subscribe Inter-networking," in Proc. of the ACM SIGCOMM 2009 conference on Content communication. New York, NY, USA.
- [2] B. Ahlgren, M. D'Ambrosio, C. Dannewitz, A. Eriksson, et al. "Second netinf architecture description," 4WARD EU FP7 Project, Deliverable D-6.2 v2.0, Apr. 2010, fP7-ICT-2007-1-216041-4WARD / D-6.2, <http://www.4ward-project.eu/>.
- [3] B. Ahlgren, M. D'Ambrosio, C. Dannewitz, M. Marchisio, I. Marsh, et al. "Design considerations for a network of information," in Proc. of ReArch'08: Re-Architecting the Internet, Madrid, Spain, Dec. 9, 2008.
- [4] PURSUIT, <http://www.fp7-pursuit.eu/>
- [5] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in Proc. of the 5th international conference on Emerging networking experiments and technologies, ser. CoNEXT '09. New York, NY, USA, 2009.
- [6] T. Koponen, M. Chawla, B. G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A data-oriented (and beyond) network architecture," in Proceedings of SIGCOMM'07, Kyoto, Japan, Aug. 27-31, 2007.
- [7] NDN, <http://www.named-data.net/>.
- [8] V. Jacobson et al., "Networking named content," in Proc. of CoNEXT, 2009.
- [9] L. Zhang et al., "Named data networking (NDN) project," NDN Project, Tech. Rep. NDN-0001, October 2010.
- [10] V. Jacobson et al., "Networking named content," in Proc. of CoNEXT, 2009.
- [11] L. Zhang et al., "Named data networking (NDN) project," NDN Project, Tech. Rep. NDN-0001, October 2010.
- [12] C. Yi, A. Afanasyev, L. Wang, B. Zhang, and L. Zhang, "Adaptive forwarding in named data networking," CCR, vol. 42, no. 3, pp. 62-67, July 2012.
- [13] C. Yi, A. Afanasyev, I. Moiseenko, L. Wang, B. Zhang, and L. Zhang, "A case for stateful forwarding plane," Computer Communications: Information-Centric Networking Special Issue, 2013.