

On-Path Caching in Information-Centric Networking

Hongseok Jeon*, Byungjoon Lee*, Hoyoung Song*

*Smart Node Research Team, ETRI, Korea

jeonhs@etri.re.kr, bjlee@etri.re.kr, hsong@etri.re.kr

Abstract— On-path caching is a key component of information-centric networking (ICN) for reducing content download time, network traffic, and server workload. In this paper, we explore the impact of representative caching strategies on the performance of the on-path caching. Simulation results show that the cache placement strategy is important in achieving good cache performance. On the contrary, cache replacement strategies do not show obvious performance improvement. In particular, stand-alone caching strategies show limitations on their capability of reducing content download time.

Keywords— Caching, on-path caching, ICN, information-centric networking, content-centric networking

I. INTRODUCTION

Caching is to place storage in the data path to store data temporarily so that subsequent requests of the data can be served faster. As the successful growth of Web increases Internet traffic, the caching is also used to deliver the Web content quickly. The Web caching was implemented as a proxy server in the early stages and it has developed into a cooperative Web caching in which multiple Web cache servers are located at different level of network to improve overall performance. Since then En-Route caching was introduced as an advanced cooperative Web caching architecture in which Web cache servers are associated with routing nodes along the routing path [1] [2]. Recently, the concept of En-route caching is being adopted by information-centric networking (ICN) which is becoming an important direction of the Future Internet architectures [3] [4] [5] [6].

ICN architectures consider caches as an integral part of the network to keep popular contents resident throughout the network and such caches are referred to as on-path caches. Routing nodes store copies of the passing content into their on-path caches and subsequent content requests can be satisfied from the on-path caches. Such an on-path caching differs from en-route Web caching in that a cached object is a unit of data that is routed on network, like a packet. Hence, the size of cached objects should be smaller than the size of maximum transmission unit (MTU) offered by links in ICN. This implies that contents should be fragmented into a number of chunks, as typically done in P2P, and each chunk should be named [4]. Also it means that caching operations must happen at line-speed, which imposes restrictions on implementing the variety of caching strategies in practice. Unlike Web caches possessing its own disk storage, on-path caches usually utilize limited storage, typically RAM-based, in routing nodes. Therefore, in designing the caching strategy, what to place

(i.e., cache placement strategy) as well as what to replace (i.e., cache replacement strategy) is important. Our goal here is to explore the impact of representative caching strategies on the performance of the on-path caching.

The remainder of the paper is organized as follows. Section II and III introduce existing cache placement strategies and cache replacement strategies, respectively. Section IV presents the performance evaluation of existing caching strategies and compares them. Section V concludes this paper with our future work.

II. CACHE PLACEMENT STRATEGY

Cache placement refers to the decision about what to cache when new content chunks arrive. At first, cache was useful for mostly system and database areas. Within the application, the cache placement strategy was not a central problem for performance. However, as caches are employed on network such as en-route Web caching and on-path caching, the cache placement strategy also becomes a critical player for better performance.

Basic cache placement strategy is the ALWAYS that caches any incoming content chunks. Many caching approaches use the ALWAYS because it is characterized by simplicity, however, it cannot avoid for certain popular contents monopolizing the limited cache storages throughout network.

FIX (P) is another simple strategy to reduce such monopolization. In FIX (P), routing nodes determine to cache the passing chunks with the fixed probability of P whose value is depending on the size of network or content popularity. In general, the optimum value of P should be acquired through empirical studies.

Laoutaris [7] first proposed a cache placement strategy based on popularity of content for Web caching in hierarchical networks where caches are located at different level of the network such as the regional level, national level, and so on. WAVE [8] generalized the popularity-based cache placement strategy regardless of specific network topologies. In WAVE, an upstream routing node explicitly suggests what content chunks to be cached at its next downstream routing node. Consequently, as content is getting popular, copies of the corresponding chunks are getting close to clients.

Dong [9] developed a mathematical model for the optimized cache placement on the condition that routing nodes know exactly which content chunks are cached in neighboring nodes. In [9], each routing nodes make a decision as to cache

or not to minimize the average content retrieval latency subject to limited capacity of their caches.

III. CACHE REPLACEMENT STRATEGY

Cache replacement is a decision about which to remove from the cache to make space for a new content chunk when the cache is full.

Most cache replacement strategies take advantage of locality of reference. The Least Recently Used (LRU) and the Least Frequently Used (LFU) are classical but popular locality-based cache replacement strategies. The LRU takes into account temporal locality which claims that a content chunk recently used will probably be reused soon. With this principle, the LRU evicts the content chunk from cache memory that has not been accessed for the longest time. The LFU is based on advanced locality principle in which the probability of reusing will increase if the number of the use is bigger. Over the years, much research has been studied to expand the classical strategies [10]. However, little work exists that expands the LRU and LFU in the context of on-path caching.

While above the caching replacement strategies use only historical information, Famaey [11] showed the theoretical gain that can be achieved by prediction-based caching strategies under assumption of perfect prediction about popularity is possible within a certain period of time in the future.

The locality-based replacement strategy basically assumes the skewed popularity of contents and it keeps the popular content chunks resident on the many caches; however, Rossi [12] proposed a cache replacement strategy, called BIAS, to make diverse contents cached. When a cache is full, the BIAS selects two chunks in cache queue at random and the more popular one is evicted.

Compared to general caching operations, the distinctive feature of on-path caching is that cached data can have different original storages. It implies, when cache misses occur, the distance required to fetch corresponding data from its origin server can be different each other. Distance travelled on the network is strongly related to the network cost of the traffic. For Web environment, Peter [13] argued that a cache replacement algorithm should minimize the cost of cache misses and thus proposed a delay-conscious cache replacement algorithm, LNC-R-W3. Recently, Sen Wang [14] also proposed a cache replacement strategy named Least Benefit with taking into account the distance factor in on-path caching.

IV. PERFORMANCE EVALUATION

We evaluate representative caching strategies that are combination of cache placement and cache replacement strategies with simulation.

A. Simulation Parameters

In the simulation, we used a perfect binary tree to generate a simple network topology that consists of 127 internal nodes and 128 leaf nodes. The internal nodes play a role of routing

nodes and the leaf nodes function as both content servers and users. Thirty thousand content files are evenly distributed among the content servers. For the simplicity, each content file has a fixed size of 10,000 Kbytes and it is divided into 1 Kbytes chunks that are tolerated by a default MTU (i.e., 1500 bytes) in Ethernet interface. Cache size on each routing node is 1,000 Kbytes. Often, the popularity of contents on the Internet is said to follow a Zipf-like distribution [15]. In the simulation, we also used the Zipf distribution with various values of α parameter in order to model a skewed popularity of contents.

B. Performance Metrics

We evaluate the performance of on-path caching in terms of the average hop counts and cache hit ratio. The average hop count estimates the average number of hops that content requests from users travel on the network to obtain requesting content chunks. This metric can be used to quantify the expected download time minimized by using on-path caching. Cache hit ratio estimates the average probability to find chunks on the routing nodes throughout the network. This metric indicates the efficiency and load balancing effect of used on-path caching strategies.

C. Simulation Results

Figure 1 shows the average hop counts of two networks: one utilizes on-path caching and the other does not. No on-path caching network lets users obtain contents only from corresponding content servers; on the other hand, on-path caching network enables users to get the requested contents from caches as well as content servers. In case of Zipf parameter of 0.7, there are shown just 0.1 decreases of average hop count by on-path caching. However, as Zipf parameter increases, the on-path caching network explicitly outperforms the network without on-path caching. At Zipf parameter of 2.5, drops in average hop counts of on-path caching network is 6.24. These results suggest that the competitive advantage of on-path caching comes from large value of Zipf parameter. Here the on-path caching uses ALWAYS and LRU strategies.

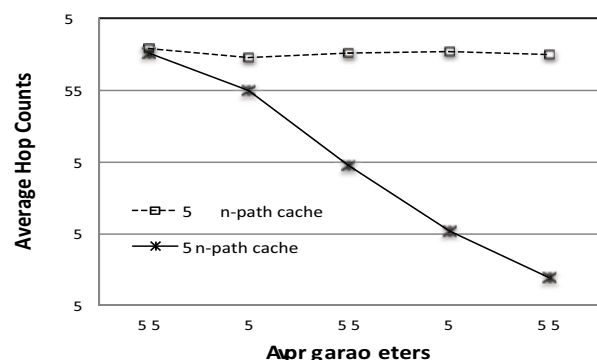


Figure 1. Average hop counts comparison between on-path caching and no on-path caching network

Large value of Zipf parameter means a given content gets much higher popularity than others. Nowadays, such an extremely skewed popularity temporally happens due to social

networks such as Facebook and Tweeter. Social networks enable contents to become highly popular for very short time. Also, some famous sites bring about slashdot effects and flash crowd in very short time [16]. As a result, on-path caching is a necessary concept to lessen the expected download time of contents in ICN.

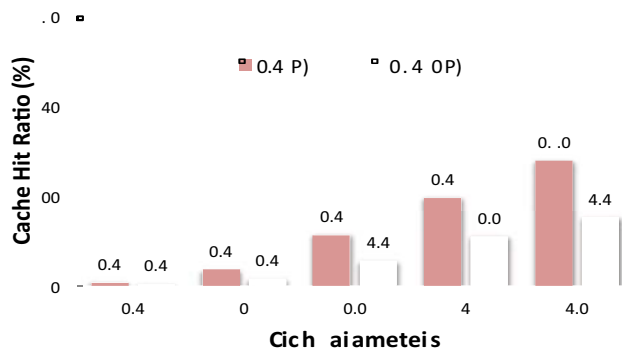


Figure 2. Cache hit ratio comparison between cache placement strategies

Figure 2 plots the cache hit ratio as a function of Zipf parameter for different cache placement strategies: FIX (P) and ALWAYS. In our simulation, FIX (P) is set to a probability resulting in best performance. In case of Zipf parameter of 0.7, FIX (0.1) shows its best performance. FIX (0.2) performs the best at 1 and 1.5 Zipf parameters. In Zipf parameter of 2 and 2.5, the probability of 0.3 causes the best performance. As can be seen, FIX (P) shows better cache hit ratio that is nearly two times higher than ALWAYS (i.e., FIX presents 6.3% higher cache hit ratios at 2.5 Zipf parameter, respectively). This implies that workload of content servers in FIX (P) is better distributed than ALWAYS. This result is not surprising because ALWAYS strategy caches content chunks aggressively. The cache hit ratio of FIX (P) increases as Zipf parameter increases. Here LRU is used as cache replacement strategy in common.

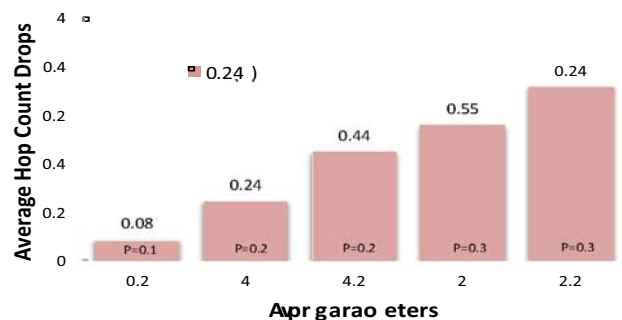


Figure 3. Drops in average hop counts of FIX (P) compared to ALWAYS

In Figure 3, we describe drops in average hop counts by adopting FIX (P), compared to ALWAYS, while LRU is used as cache replacement strategy in common. As expected, the FIX (P) brings benefit to the average hop counts and such a benefit increases as Zipf parameter increases. However, FIX (P) does not show a notable difference from ALWAYS in

comparison with the cache hit ratio (i.e., FIX (P) cannot even decrease 1 hop in the average hop count).

In Figures 4, 5, 6, and 7, we explore the impact of four different cache replacement strategies, LRU, LFU, LRU + network cost (NC) and LFU+NC. The network cost refers to the distance travelled to content server to get content chunks in case of cache miss. The LRU+NC considers the network cost in addition to recency and the LFU+NC utilizes the network cost multiplied with frequency. Hence, the network cost factor is more emphasized in LFU+NC rather than in LRU+NC.

From Figures 4, 5, 6, and 7, we observe that the cache replacement strategies exhibit no obvious difference in their performance. In a static popularity distribution, LFU generally outperforms LRU by a considerable margin. However, in our simulation results, the performance of LRU is quite similar to that of LFU. We think this is because on-path caching provides a different reference pattern (e.g., not static popularity) rather than traditional caching. Recently, Web 2.0 products lots of short media contents and social networks generate dynamic content popularity, which do not show strong locality of reference. Also LRU+NC does not show explicit difference with LRU and LFU. It seems that the network cost in LRU+NC has a little impact. Only LFU+NC makes a slightly negative impact on FIX (P) while providing a little positive impact on ALWAYS. This behaviour is related to effect of considering network cost.

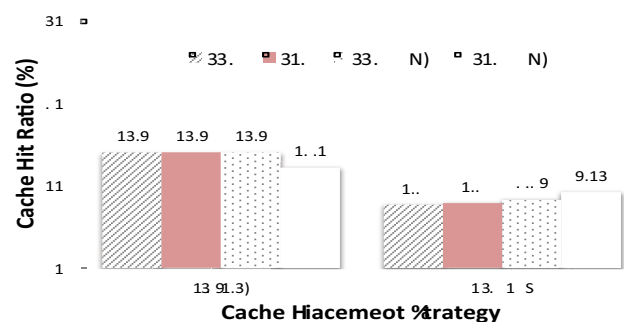


Figure 4. Cache hit ratio comparisons between caching strategies in 2.5 Zipf parameter

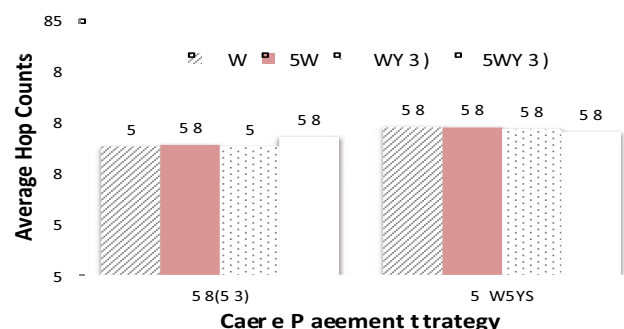


Figure 5. Average hop counts comparison between caching strategies in 2.5 Zipf parameter

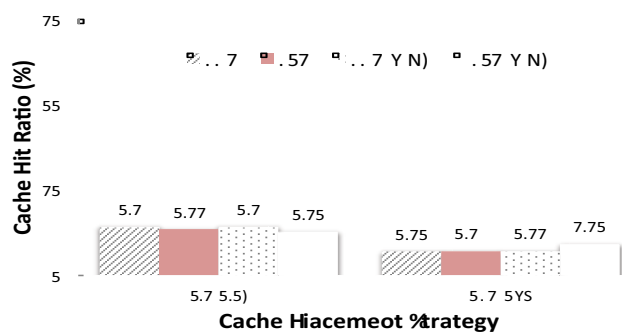


Figure 6. Cache hit ratio comparison between caching strategies in 1.5 Zipf parameter

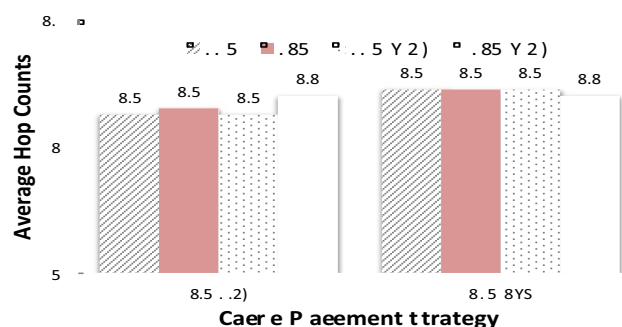


Figure 7. Average hop counts comparison between caching strategies in 1.5 Zipf parameter

V. CONCLUSIONS

This paper focuses on the impact of representative caching strategies on the performance of the on-path caching. We have performed simulation experiments to compare the two cache placement strategies and four cache replacement strategies. The results show that the cache placement strategy is a key factor in on-path caching performance; on the contrary, cache replacement strategies do not show obvious performance improvement. Also, the results point out that high cache hit ratio does not imply low average hop counts. To lessen the average hop count, i.e., content download time, some interworking between routing nodes might be required. We plan to further investigate such an interworking method to decrease average hop counts remarkably as future work.

ACKNOWLEDGMENT

This research was supported by the KCC (Korea Communications Commission) / KCA (Korea Communications Agency) of Korea, under the 'R&D on Smart Node Technology for Cloud Networking and Contents Centric Networking' support program supervised by the KCA (Korea Communications Agency) (KCA-2011-11913-05003)

REFERENCES

- [1] S. Bhattacharjee, K. L. Calvert, and E. W. Zegura. 1998. Self-organizing wide-area network caches. Proc. IEEE INFOCOM 1998 (March. 1998)
- [2] X. Tang and S. T. Chanson. 2002. Coordinated En-Route Web Caching. Journal. IEEE Transaction on Computer. (June 2002), 595-607.
- [3] T. Koponen, et al. 2007. A data-oriented (and beyond_ network architecture. Proc. ACM SIGCOM 2007 (Aug. 2007)
- [4] V. Jacobson, et al. 2009. Networking named content. Proc. ACM CoNEXT 2009 (Dec. 2009)
- [5] A. Zahemszky, et al. 2009. Exploring the Pub/Sub Routing & Forwarding Space. Proc. IEEE ICC 2009 Workshop Future-Net (June. 2009)
- [6] L. Dong, et al. 2009. On the Cache-and-Forward Network Architecture. Proc. IEEE ICC 2009 (June. 2009)
- [7] N. Laoutaris, S. Syntila, and I. Stavrakakis. 2004. Meta algorithms for hierarchical web caches. Proc. IEEE PCCC 2004 (2004)
- [8] K. Cho, et al. 2012. WAVE: Popularity-based and Collaborative In-network Caching for Content-Oriented Networks. Proc. IEEE INFOCOM 2012 Workshop NOMEN (March. 2012)
- [9] L. Dong, et al. 2011. Optimal Caching with Content Broadcast in Cache-and-Forward Networks. Proc. IEEE ICC 2011 (June. 2011)
- [10] S. Podlipnig and L. Boszormenyi. 2003. A Survey of Web Cache Replacement Strategies. Journal. ACM Computing Survey (CSUR) (Dec. 2003)
- [11] J. Famaey, T. Wauters, and F. D. TURCK. 2011. On the Merits of Popularity Prediction in Multimedia Content Caching. Proc. 12th IFIP/IEEE INM 2011 (2011)
- [12] D. Rossi and G. Rossini. 2011. Caching performance of content centric networks under multi-path routing (and more). Technical report, Telecom ParisTech (2011)
- [13] P. Scheuermann, J. Shim, and R. Vingralek. 1997. A case for delay-conscious caching of Web documents. Journal. ELSEVIER Computer Networks and ISND Systems (September. 1997), 997-1005
- [14] S. Wang, et al. 2011. Could In-Network Caching Benefit Information-Centric Networking?. Proc. ACM AINTEC 2011 (2011)
- [15] G. Carofiglio, et al. 2011. Modeling Data Transfer in Content-Centric Networking. Proc. International Teletraffic Congress (ITC). (2011)
- [16] C. Canali, M. Colajanni, and R. Lancellotti., 2010. Characteristics and evolution of content popularity and user relations in social networks. Proc. IEEE Symposium on Computers and Communications (ISCC'10) (June. 2010)