A Collaborative Cache Off-Loading Algorithm for Content Centric Networks

Dabin Kim, Sung-Won Lee, Young-Bae Ko
Graduate School of Information and Communication
Ajou University
Suwon, Republic of Korea
{dabin912, sungwon}@uns.ajou.ac.kr, youngko@ ajou.ac.kr

***Abstract*—Content centric networking (CCN) is suggested as one of the key troubleshooter technologies for addressing increasing Internet traffics efficiently. However, physical capacity of in-network cache is restricted despite of a major component in CCN due to the limited hardware technology. Therefore the caching performance of each CCN router can be very limited to deal with huge amount of contents. In this paper, we present a new cache management algorithm to cooperate with neighbor routers' caches. Our preliminary simulation study verifies that the proposed algorithm effectively deals with huge amount of contents with reduced network load.**

***Index Terms*— Content Centric Network, Cache Management**

I. Introduction

Content centric network (CCN) [1] is one of the state-of-the-art network paradigms to efficiently disseminate large amounts of increasing Internet contents and provide high level of services to users. The main distinctive differences between CCN and current Internet are request-driven communication model and in-network caching capability. Most important of all, in-network caching is a key feature of CCN. It helps alleviate server burden by placing contents replicas to local caches in the vicinity of the requestor and reduce traffic redundancy. To take advantage of those characteristics and realize an intelligent in-network caching performance in CCN, many literatures have studied how to efficiently cache the content in a local cache. Some researchers proposed new cache replication algorithms for CCN, such as collaborative cache mechanisms which utilize adjacent cached contents that are exactly matched to data requests.

However, caching technologies in CCN still do not press home their advantages when the amount of popular contents such as multimedia data increases. This is because current reasonable cache memory of CCN is relatively small compared to the overall Internet traffic volume [2]. Therefore, when lots of various requests including much popular contents are concentrate to a specific router, it suffers from heavy caching overhead of replacements. In that case, it cannot support proper services for upcoming request because contents are evicted too quickly. For this reason, even though each CCN router tries to cache the contents based on the well-managed policies, it is difficult to avoid chronic cache problems such as cache pollution and cache thrashing [3]. Consequently, caching performance significantly decrease while those kind of problems remain.

To resolve these problems we propose a novel collaborative caching algorithm which expands a cache memory logically by sharing underutilized neighbor's cache space according to caching burden of a single CCN router. Our main goal is to cooperatively utilize more than one neighbor's cache to mitigate chronic cache problems as well as increase cache utilization. To the best of our knowledge, this is the first-attempt to share some part of unloaded cache space for reducing caching burden of overloaded CCN router. Furthermore, we consider our proposed scheme as a starting point for constructing the cache cloud concept in CCN.

II. Collaborative Cache Off-LoadingAlgorithm

A. Off-Loader and Collaborator Decision

In the proposed scheme, we define two terms for the algorithm: 1) *off-loader* and 2) *collaborator*. The *off-loader* is defined as a temporarily overloaded router which is mainly on the communication paths and suffers from heavy cache processing. Off-loader hands its cached contents to the underutilized neighbor router, the *collaborator*. When routers have responsibility for caching contents according to each caching policy, routers in the network determine whether they act as an off-loader or a collaborator dynamically. To make this decision, we utilize the term of cache survival time. The cache survival time represents the average residence time for each evicted contents in a cache, which is time between contents insertion and eviction. The low value indicates that most of contents are frequently replaced due to lots of data flows. Therefore we decide the off-loader as a node that has lower cache survival time and a collaborator as a node that has higher caching survival time. For a convenience of cache survival time calculation, we consider LRU replacement policy and calculations are exchanged through additional control packets.

B. Cache Off-Loading Procedure

When a new chunk arrives at a router and there is a necessity for caching, router compares its cache survival time

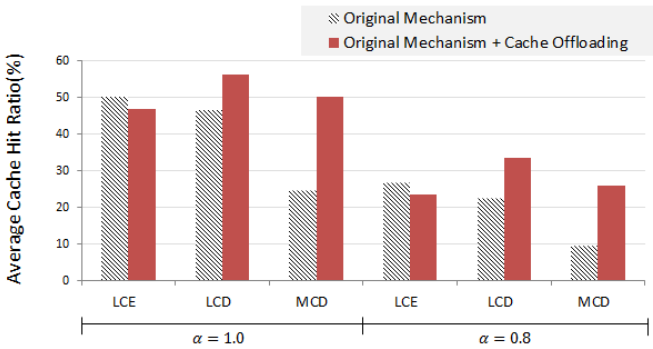

Figure 1 Cache hit ratio comparison

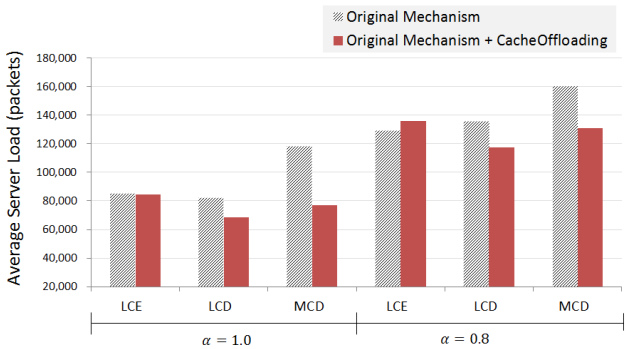

Figure 2 Server load comparison

with its 1-hop neighbor nodes for making role decision. If it has the lowest cache survival time of overloaded contents requests, it is determined as off-loader, and it selects *collaborator* as the node that highest cache survival time value. After selecting collaborator, off-loader caches a newly received chunk in its local cache and transfers the evicted chunk, which has been highly reused since being stored, to the collaborator. In this process, to prevent potential oscillatory effect from sharing cache space between collaborator and off-loader, we give restriction for that offloaded contents cannot be offloaded again until it is evicted at the collaborator's cache. After then, off-loader inserts a path for the chunk toward a collaborator into the Forwarding Information Base (FIB) with an expiration timer set by its *cache survival time*. As cache survival time of *off-loader* is obviously lower than collaborators, it prevents interests from being misled after certain periods. To separate the off-loading transfer from normal data forwarding, we define a special field in the data header, named by *enforce flag*. The collaborator which receives a chunk with the set *enforce flag* inserts the chunk into the cache without checking Pending Information Table (PIT).

III. Performance Evaluation

Our preliminary performance analysis is evaluated by using OPNET [4]. To gather reliable results, we have simulated under the realistic Internet-like topology generated with GT-ITM [5]. In this network, one of 100 CCN routers is set as a content server and the other nodes have ability to request contents and cache chunks. We use 10,000 contents following the Zipf's distribution popularity whose alpha value is set to 0.8 and 1.0 considering the large popular content portion [6]. Each node can store at most 1% of the whole contents. The proposed scheme can cooperate with the conventional cache management algorithms such as Leave Copy Everywhere (LCE), Leave Copy Down (LCD), and Move Copy Down (MCD) [7]. Therefore we evaluate the caching performance under the cooperation with these algorithms as shown in Figure 1 and Figure 2. In case of LCD and MCD, the caching performance with our scheme presents better cache hit ratio and lower server load than the original LCD and MCD. This is because those algorithms make more popular contents disseminated into wider region. Hence by adopting our proposed scheme those contents are maintained in nearby cache with higher cache survivability, affording more opportunity to offer the content without simply eviction. However, in case of LCE, our proposed scheme shows slightly reverse effect. The reason of this phenomenon is that LCE equally distributes popular and unpopular contents around the whole network so that there are huge amount of cache replacements due to unpopular contents. Due to an absence of popularity-based decision in the proposed scheme yet, the combination of LCE and the proposed scheme leads to performance decreases with more cache replacements which have negative influence to cache popular contents.

IV. Conclusion and Future Work

We proposed the off-loading based cache management algorithm to deal with large amount of popular contents and improve cache utilization as well as reducing server load. Future work we will include supporting the decision method of off-loader and collaborator with a consideration of more complex environments. We will also consider various contents popularity to determine which contents should be cached for improving the caching performance.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (2012-003573).

References

[1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, R. L. Braynard, Networking Named Content, In ACM CoNEXT, 2009.

[2] D. Perino and M. Varvello. A reality check for content centric networking. In ACM Sigcomm workshop on ICN, 2011.

[3] V. Seshadri, O. Mutlu, M. AKozuch, T. C Mowry, The Evicted-Addreess Filter: A Unified Mechanism to Address Both Cache Polluion and Thrashing, PACT, 2012.

[4] OPNET Simulator, http://www.opnet.com/

[5] E. Zegura, K. Calvert, and S. Bhattacharjee, How to model an Internetwork, In IEEE INFOCOM, 1996.

[6] C.Fricker, P.Robert, J.Roberts, and N.Sbihi, Impact of traffic mix on caching performance in a content-centric network, In IEEE NOMEN Workshop, 2012.

[7] N. Laoutaris, S. Syntila, and I. Stavrakakis, Meta algorithms for hierarchical web caches, In IEEE IPCCC, 2004.