A Study on Cache Suppression Control According to Content Popularity for In-network Guidance

Hiroki Kawabata, Yumi Takaki, Chikara Ohta, and Hisashi Tamaki
Graduate School of System Informatics, Kobe University
{kawabata, yumi, ohta, tamaki}@al.cs.kobe-u.ac.jp

ABSTRACT
Content servers have become burdened by the transfer of large content sets such as high-definition video and music. To mitigate this problem, we propose a Mapping Server with Cache-location Resolution with Cache Suppression (MSCR/CS), which suppresses the unnecessary duplication of popular contents in network caches. Simulations show that MSCR/CS decreases content-server loads.

Categories and Subject Descriptors
C.2.1 [Computer-Communication Networks]: Network Architecture and Design—Network communications

Keywords
Mapping server, In-network cache, Cache suppression, Zipf-Mandelbrot

1. INTRODUCTION
The Internet is now being used to transfer large contents (or files) such as video and music more often. This has stimulated research on content-oriented networks (CON) [1, 2, 3, 4]. In CONs, how to identify content, i.e., content naming, how to locate (or find) content, and where and how to store contents are major issues.

It was shown in [5, 6] that content popularity on the Internet follows a Zipf-Mandelbrot distribution. Accordingly, the network-caches are occupied by the most popular contents to the exclusion of other contents; this lowers the cache hit ratio and thus network performance. Increasing the variety of cached contents raises the hit ratio which effectively reduces content server load and core network traffic.

To retrieve a content, the contents ID must be known after which the nearest server’s location must be resolved. In [7], we proposed “Mapping Server with Cache-location Resolution (MSCR)”; it utilizes distributed caches efficiently by resolving prospective content caches (PCL: Prospective Cache Location) close to the user as well as the original content server from content ID.

This paper introduces an improved version of MSCR, called “MSCR with Cache Suppression (MSCR/CS),” which suppresses content duplication in order to increase the variety of cached content. It describes simulations conducted to verify the effectiveness of MSCR/CS.

2. MSCR/CS: MAPPING SERVER WITH CACHE-LOCATION RESOLUTION WITH CACHE SUPPRESSION

2.1 Procedure of content fetching
We use Figure 1 to explain content fetch in MSCR/CS. Here, we assume that the user already knows content ID.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

CoNEXT Student Workshop'13, December 9, 2013, Santa Barbara, CA, USA.
Copyright © 2013 ACM 978-1-4503-2575-2/13/12 ...$15.00.
http://dx.doi.org/10.1145/2537148.2537784.
((3) in the figure). If the query finds a copy of the content on the way to the PCL, the content is transmitted from there to the user ((4) in the figure) and the query is terminated at that server. If the top PCL cannot serve the query, it deletes its name from the PCL-list, and forwards the query to the (new) top PCL in the PCL-list ((5) in the figure). If the PCL-list becomes empty, the query is forwarded to the content server ((6) in the figure), and the content is fetched from there ((7) in the figure). If the MSCR/CS can not generate the PCL-list, it simply replies {content server location, null, suppression flag} toward the user, who then sends a query to the content server.

As the content is forwarded to the user, copies are placed in the transit servers if-and-only-if the suppression flag in the query is OFF (see below). Note that MSCR/CS is designed to maximize the variety of content cached, the actual determination of which server caches the content can be controlled by any of the existing cache replace policies, like LRU (Least Recently Used).

### 2.2 Cache suppression flag

In this paper, we assume that content popularity follows a Zipf-Mandelbrot distribution (equation (1)) since it can model the popularity distribution of video contents [6]. MSCR/CS can estimate the distribution accurately according to the history of resolution requests.

$$p(i) = \alpha(i + 5)^{-0.6} \left( \alpha = 1/\sum_{i=1}^{N}(i + 5)^{-0.6} \right), \quad (1)$$

where $p(i)$ denotes the probability that the content requested is the $i$th most popular content, and $N$ denotes the total number of unique contents.

In order to increase the variety of cached contents, MSCR/CS restricts cache creation more frequently for more popular contents. MSCR/CS allows a query to create a content cache (suppression flag is set “OFF”) with probability $q(i)$:

$$q(i) = \begin{cases} \min(\beta(i + 5)^s, 1), & \text{if } i \leq \hat{C}, \\ 0, & \text{otherwise}, \end{cases} \quad (2)$$

where $\beta$, $s$, and $\hat{C}$ are control parameters whose values are positive. Parameter $\beta$ determines entire content cache generation frequency. The smaller the value of $\beta$ is, the less frequent is cache creation. Parameter $s$ adjusts the cache generation frequency according to the popularity rank of contents. Increasing the value of $s$ selectively suppresses the more popular content. Parameter $\hat{C}$ is the popularity rank threshold; any content whose popularity is less than $\hat{C}$ is always suppressed because such a content is rarely accessed and cached intrinsically.

Note that the value of $q(i)$ lies in the range of 1 to 0 because it is a probability.

### 3. PERFORMANCE EVALUATION

This section evaluates how much MSCR/CS can decrease the content server load and core network traffic. To do so, we developed an event-driven simulator. In our simulation, we used 3-tier network topologies, which were generated by gt-itm [8]. In the topologies, the tier-1 network is regarded as the core network. We set the number of routers, content servers, and users to 1, 924, 50, and 913, respectively. The number of unique contents was set to 10,000, and content popularity followed a Zipf-Mandelbrot distribution as shown in equation (1). We compared four scenarios as shown below from the viewpoints of server download ratio (ratio of the number of contents obtained from content servers to that of accessed contents) and the relative ratio of the amount of tier-1 traffic in traditional server/client (IP) model. 1) IP: a user fetches a content from its content server as per the server/client model of basic IP networks. 2) Cache: access routers have content caches. 3) MSCR: a query can be sent toward a PCL as well as the original server location. 4) MSCR/CS: excess cache duplication is controlled as well as basic MSCR function. In the above, cache storage can store up to five contents, which corresponds to 0.05% of content number. In our simulation, a single centralized MSCR/CS (MSCR) manages all content information, and we set $\beta = 0.1$, $s = 0.4$, and $\hat{C} = 4,656$. Development of distributed MSCR/CS(s) for load balancing is left as future work.

### 3.1 Simulation results

As shown in Table 1, MSCR decreases server download ratio and the amount of tier-1 traffic compared with IP and Cache schemes. This is because MSCR enables users to more effectively utilize the content caches of access routers other than their own. MSCR/CS reduces the server download ratio by 13.7 compared to the original MSCR. On the other hand, the amount of tier-1 traffic slightly increases. This is because the distance from users to prospective cache locations of popular contents tends to be longer than that in MSCR, which allows more caches to be created overall. Consequently, we confirm that MSCR/CS is effective in reducing the server download ratio.

### 4. CONCLUSION AND FUTURE WORK

In this paper, we proposed MSCR/CS; it suppresses unnecessary cache duplication of highly popular contents in order to increase the variety of cached contents in the network. Simulations showed that MSCR/CS is effective in reducing the content server loads. As future works, we will extend MSCR/CS to include a function that can estimate content popularity, and investigate how to realize the cooperation of multiple MSCR/CS(s) for load balancing.

### 5. ACKNOWLEDGMENTS

We would like to express our deepest gratitude to Prof. Jim Kurose and Dr. Elisha Rosenweig who provided valuable comments and suggestions. We gratefully acknowledge the financial support of Information and Communications Technology (NICT), Japan.
6. REFERENCES


