Content Retrieval Method in Cooperation with CDN and ICN-based In-network Guidance **Over IP Network**

Yutaro Inaba

Yosuke Tanigawa

Hideki Tode

Dept. of Computer Science and Intelligent Systems Osaka Prefecture University, Osaka, Japan Email:{inaba@com.,tanigawa@,tode@}cs.osakafu-u.ac.jp

Abstract-These days, in addition to host-to-host communication, Information-Centric Network (ICN) has emerged to reflect current content-centric network usage that many users are now interested not in where the content is but in acquired contents themselves. However, the authors believe that current IP network still remains, at least from deployment perspective, as one of near future network architectures. This is because ICN has various scalability and feasibility challenges, and hostto-host communication is also diffused like remote login, VoIP, and so on. Therefore, the authors' research aims to establish the feature of information-centric network on conventional IP network to achieve feasible architecture. In this paper, we propose to operate Breadcrumbs (BC) and Content Delivery Network (CDN) frameworks coordinately on IP network to improve the performance on content retrieval and acquisition. Both BC and CDN are important as a content-centric technique. Finally, we compare the proposed method with CDN that we carefully modeled through simulation. Simulation results show that our approach can reduce server load.

I. INTRODUCTION

Information-Centric Network (ICN) [1] has emerged as a future network architecture to fit current content-centric network usage. ICN names contents in network layer and routing is processed by not location information but contents' name. However, ICN has scalability problem because naming every content in the network takes much overhead. Hence, it is less feasible to operate ICN in very large scale network like the Internet. On the other hand, the idea of ICN can be utilized to establish effective system on current IP network. There are several content-centric techniques which can be implemented on IP network. Among them, we focus on Breadcrumbs (BC) [2] and Content Delivery Network (CDN) [3] [4].

In CDN, replica servers called surrogate servers are placed dispersedly in network. Content providers replicate their contents to surrogate servers and contents are delivered to users from there. Using CDN, users' requests can be dispersed, and thus fast and stable content delivery becomes possible. However, CDN has some disadvantages. Surrogate servers need to be capable enough to handle large amount of accesses, because CDN is supposed to deal with flash crowds. This results in high management cost. As a result, using CDN also costs to some degree.

On the other hand, BC method has been proposed, that easily implement content-centric feature on not only the ICN but also the conventional IP network. BC entry is created at each router on the download path when a content is downloaded to a user, and the user makes a cache of the content. Here, we assume the cache capability is placed in only user-side such as edge router, ONU, STB and user's PC in terms of higher feasibility. Each BC entry is utilized to route a request to the target content located at the corresponding user's cache. Then, the content is delivered from the user's cache. We can implement the BC method on the conventional IP network with only small change of current system [5] [6] because of it's passive and simple approach.

In this paper, we operate BC and CDN frameworks coordinately on the IP network. Our main goal is to decrease server load and improve the performance on content retrieval and acquisition.

II. RELATED WORKS

Although there are some ICN approaches [7] [8], ICN is still immature and has scalability problems. In Named Data Networking (NDN) approach [7], each Content Router (CR), which corresponds to conventional router, needs to keep contents' names and their directions to forward them for routing. Thus, each CR must have huge amount of routing information because there are enormous contents in the Internet. Data Oriented Network Architecture (DONA) approach [8] forms hierarchical routing structure. Content providers register the contents' names and locations at Resolution Handler (RH), which corresponds to conventional router. These registrations are sent up to high level RHs, and hence, top level RH needs to have tremendous routing information. ICN has serious scalability problem as its routing is based on contents' names. To avoid this problem, we focus on Breadcrumbs [2] and CDN [3] [4] as content-centric methods; the former can be established easily on conventional IP network because of its simple mechanism, and the latter is actually working on the IP network.

In the proposed method, we utilize only user's cache. There are some researches in which router's cache is also utilized. In [9], each router has a cache of contents. Users' requests are sent through not shortest paths but bypassed paths where many caches exit. As a result, some users can obtain contents from nearer caches than the server. However, replicating cache in a router requires much memory capacity and very fast read/write operation of the memory in routers. In [10] [11], users can obtain the contents from various proxy servers. Each proxy server coordinately communicates where the contents are, and users' requests are guided to one of the proxy servers.

In the proposed method, by contrast, only users have caches of contents, and guide the users' requests to the caches by using BC. We make the requests encounter the BC entries with higher probability by using CDN, and bypass the requests to the user cache.

III. PROPOSAL

In this section, we propose cooperative control between BC and CDN frameworks. In the proposed method, BC guides as many requests as possible.

A. Calculation of BC hit rate

In most of CDN services, DNS selects the best surrogate server for each requesting user based on the distances and the server loads. In addition, the proposed method introduces BC hit rate (R_{BChit}) to select surrogate servers. BC hit rate is calculated per surrogate server and content, based on the following formula,

$$R_{BChit} = C_{BC}/C_{all},\tag{1}$$

where C_{BC} is the number of the requests handled by BC and C_{all} is the number of all the generated requests. To calculate BC hit rate, we need to count these two numbers. DNS counts C_{all} when it sends an IP address to a user. Users need to report C_{BC} if we try to obtain it directly. However, to suppress users' loads, surrogate servers periodically report the number of requests handled by them (C_s) to DNS. Then, DNS calculates C_{BC} as follows,

$$C_{BC} = C_{all} - C_s. \tag{2}$$

B. Request redirection

DNS selects the surrogate server with the highest BC hit rate among the N_S designated surrogate choices. Top N_S surrogate servers as for the smaller distances to a user are selected as the surrogate choices for the user. We assume that the distances between each user and surrogate servers are computed in advance. Here, we eliminate high load surrogate servers, which offer many contents simultaneously, from surrogate choices. When all the surrogate servers in surrogate choices are high load, we choose nearest surrogate server which is not high load outside of surrogate choices. If all surrogate servers are high load, we choose the surrogate server with the least load.

In Fig. 1, we demonstrate the behavior of the proposed method. First, a user sends a name resolution request to DNS (1). Different from common CDN, DNS selects a high BC hit rate surrogate server as described above and sends it's IP address to the user (2). The user sends a request to the selected

surrogate server. If the request reaches a router with requesting BC, it is guided by BC and transferred to the user-side with cache instead of the surrogate server (a). Then, the requesting content is sent to the user (b). Meanwhile, if the request does not encounter a BC, it is sent to the selected surrogate server, and the content is delivered from there (3, 4). By delivering contents from users' cache, the workload on surrogate server can be decreased.

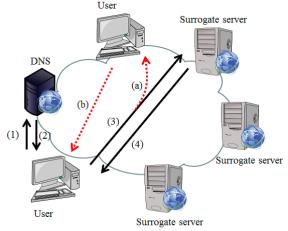


Fig. 1. Surrogate server selection in the proposed method. IV. EVALUATION

In this section, we evaluate the performance of the proposed method through computer simulation.

A. CDN model

1) Where to place surrogate servers: Although some methods challenge to determine an optimal surrogate server placement, these methods are NP-hard. Alternatively, heuristic methods are proposed to approximate an optimal placement like [12]. In [12], Greedy and HotSpot algorithms are proposed, and Greedy outperforms HotSpot. Hence, we determine the positions of surrogate servers according to Greedy algorithm. The first surrogate server is placed on a router where sum of the number of hops from all users is minimum among all routers. The second and the subsequent surrogate servers are sequentially placed on other routers so that sum of the number of hops from all users to their nearest surrogate servers is minimum.

2) Which surrogate server to redirect a user's request to: We redirect requests to a surrogate server by DNS redirection. In this method, as the response to a user's name resolution request, DNS selects the best surrogate server for the user and sends the server's IP address to it. In the proposed method, we select surrogate server according to above-mentioned selection policy. On the other hand, as for the compared existing method explained in the following subsection, each user sends requests always to its nearest surrogate server.

3) How to make caches on surrogate servers: We adopt non-cooperative pull based outsourcing as many commercial CDN administrators use it because of it's simplicity. In this method, when a request reaches a surrogate server, if the

requested content is not on the surrogate server, it requests the content to an original server. When the surrogate server receives the content from the original server, it makes a cache of the content and sends it to the user.

B. Simulation scenario

We set each parameter as shown in TABLE I. Unlike original BC method, we assume that contents are cached only in user-sides except for surrogate servers, and routers have only BC entries to save their computational cost. In addition, we use BC+ method [13] instead of original BC method because naive BC method has a routing loop problem that requests are transferred within specific routers forever and cannot reach a server or a cache.

In this simulation, we do not consider the overhead to calculate BC hit rate. In other words, we assume that BC hit rate can be immediately calculated when requests are generated at users. As described in Section III, the proposed method takes into account surrogate server load to choose a surrogate server. This time, however, we do not consider the surrogate server load for simplicity.

Parameter	Value
# Routers	1000
# Users	5000
# Original servers	50
# Surrogate servers	5
# Contents	10000
User cache space	2 contents
Surrogate server cache space	1000 contents
User upload limit	5 contents
Interval of request generation per user	2000 sec
Cache replacing policy	LRU

TABLE I. Parameter settings.

1) Network topology: We generate a random network based on Waxman model ($\alpha = 0.3$, $\beta = 0.05$) [14]. Each original server is connected with a router at random. All routers are connected with five different users.

2) Requests for contents: Each user requests a content at the independent, identical and exponentially distributed random interval, and a content to be requested is selected according to Zipf-Mandelbrot with exponential cutoff distribution ($\alpha = 1.0, q = 20, \beta = -1.5^{-3}$) [15].

3) Packets and delay: We assume that each packet size is constantly 1,500 Byte. A request and a control packet consist of one packet, and a content consists of 768,000 packets, which corresponds to 30 min video file with transfer rate of 5 Mbps. In this simulation, we suppose each router has an enough processing capability, and delays are constant. It consistently takes 2.3 ms for one packet to travel from a router to it's next router.

4) Compared methods: We evaluate the proposed method by comparing following systems:

- Proposed CDN+BC(N_S) Users select a surrogate server with as high BC hit rate as possible according to the policy in Section III. We vary surrogate choices (N_S) from one to five.
- Simple coexisting approach of BC and CDN This simply operates BC and CDN frameworks simultaneously and independently. Each user selects the nearest surrogate server and does not consider BC hit rate. This system corresponds to CDN+BC(1).
- Legacy CDN+IP
 - CDN with conventional IP routing.

We use the following evaluation metrics:

- Surrogate utilization ratio The ratio of requests handled by a surrogate server to all the generated requests.
- Average hop counts Average number of hops at which requests and contents are sent.
- Surrogate selection counts

Number of requests each user sends to each selected surrogate server, ranked by the distance between the user and the surrogate server.

C. Results

1) Surrogate utilization rate: Figure 2 shows the ratio of requests handled by a surrogate server to all generated requests. In the proposed CDN+BC, each number on the horizontal axis represents the setting of N_S . In legacy CDN+IP, all requests are transferred to a surrogate server. Surrogate utilization ratio, therefore, equals one. In the proposed CDN+BC, some requests are guided by BC; then contents are delivered from users' caches, and thus, surrogate utilization ratio decreases. In case of $(N_S = 1)$, note that BC and CDN are not operated coordinately but just coexisting together. In other words, each user constantly selects the nearest surrogate server and does not consider BC hit rate. We can see that surrogate utilization ratio decreases as surrogate choices increase. This is because requests are sent to high BC hit rate surrogate server, and the contents are delivered from the user-side cache guided by BC trail. Compared with legacy CDN+IP, the proposed method can greatly reduce the server load; 80 % reduction in case of CDN+BC(5).

2) Hop count: TABLE II represents average hop counts of request and content. In the proposed CDN+BC, both request and content hop counts become large compared with legacy CDN+IP, and the hop counts increase as the number of surrogate choices is enlarged. This is because a surrogate server, which is relatively far from a user compared with the nearest surrogate server for the user, is also selected if its BC hit rate is larger, when there are many surrogate choices. In addition, some requests are guided by BC to a far user-side cache. As a result, average hop counts become large. In particular, request hop count shows larger increasing rate to the

number of surrogate choices because requests are not always transferred based on minimum hop route by BC-based routing. However, size of request is relatively small, and therefore, large request hop count does not have serious impact on network load. In the proposed method, we can keep content hop count low. Thus, our approach can keep negative effect on network load minimum.

3) Surrogate server selection counts: Figure 3 shows the number of requests that users send to each selected surrogate server. It is counted by every distance rank from user to surrogate server. Generally, when a user sends a request to far surrogate server, there is more possibility to encounter BC as the request goes through many routers. Hence, the most simple way to raise BC hit rate is only sending a request to far surrogate server. However, it is not effective because hop count becomes large. In Fig. 3, we can confirm that the proposed method sends requests to not only far surrogate server but also near one. As to surrogate choices $(N_S = 3)$, users send a request to only top three surrogate servers in the nearest. Therefore, users do not send a request to surrogate servers farther than 4th distance rank.

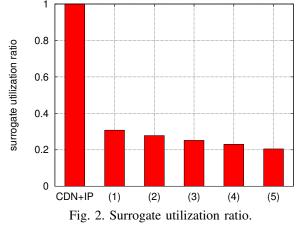


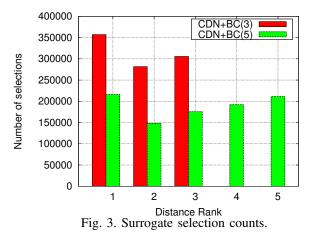
TABLE II. Average hop counts.

Method	Request hop	Content hop
CDN+IP	5.87	5.87
CDN+BC(1)	7.68	6.34
CDN+BC(2)	8.26	6.58
CDN+BC(3)	8.78	6.75
CDN+BC(4)	9.27	6.90
CDN+BC(5)	9.84	7.08

V. CONCLUSION

In this paper, our goal is to decrease server workload and improve the performance on content retrieval and acquisition. To achieve this, we presented operating BC and CDN methods coordinately.

Simulation results show that combining CDN with BC method results in reduction of CDN utilization although average hops increase to some degree. Also, as to the increase of average hop, the increase of the number of hops of content, which has great impact on the network load, can be kept small.



We are planing to consider better request guidance method as a future work.

ACKNOWLEDGMENT

This research was supported in part by National Institute of Information and Communication Technology (NICT), Japan.

References

- G. Xylomenos, "A Survey of Information-Centric Networking Research," IEEE Communications Survey and Tutorials, vol. 16, pp. 1024-1049, Sept. 2013.
- [2] E. J. Rosensweig and J. Kurose, "Breadcrumbs: efficient, best-effort content location in cache networks," in Proc. IEEE INFOCOM 2009, pp. 2631-2635, Apr. 2009.
- [3] A. Passarella, "Review: A Survey on content-centric technologies for the current Internet: CDN and P2P solutions," Computer Communications, vol. 35, no. 1, pp. 1-32, Jan. 2012.
- [4] J. Dilley, et al., "Globally Distributed Content Delivery," IEEE Internet Computing, vol. 6, no. 5, pp. 50-58, 2002.
- [5] T. Yagyu, M. Kakida, Y. Tanigawa, and H. Tode, "Prototype Development of Active Distribution of In-network Guide Information for Contents Delivery," in Proc. IEICE Technical Report, vol. 111, no. 468, NS2011-211, pp. 179-184, Mar. 2012 (in Japanese).
- [6] T. Yagyu, "Synergic Effects among Plural Extensions of Breadcrumbs for Contents Oriented Networks," in Proc. AFIN 2013, pp. 35-42, Aug. 2013.
- [7] NSF Named Data Networking project. [Online]. Available: http://www.named-data.net/
- [8] T. Koponen, et al., "A data-oriented (and beyond) network architecture," in Proc. ACM SIGCOMM 2007, pp. 181-192, Aug. 2007.
- [9] V. Sourlas, et al., "Cache-aware routing in Information-Centric Networks," in Proc. IFIP/IEEE IM 2013, pp. 582-588, May 2013.
- [10] D. Wessels and K. Kaffy, "Internet Cache Protocol (ICP), version 2," RFC1286, Sept. 1997.
- [11] T. Asaka, et al., "Hash-Based Query Caching Method for Distributed Web Caching in Wide Area Networks," IEICE Transactions on Communications, vol. E82-B, pp. 907-914, June 1999.
- [12] L. Qiu, V. Padmanabhan, and G. Voelker, "On the placement of web server replicas," in Proc. IEEE INFOCOM 2001, pp. 1587-1596, 2001.
- [13] M. Kakida, Y. Tanigawa, and H. Tode, "Breadcrumbs+: Some Extensions of Breadcrumbs for In-network Guidance in Content Delivery Networks," in Proc. SAINT 2011, pp. 376-381, July 2011.
- [14] B. M. Waxman, "Routing of multipoint connections," IEEE J. Selected Areas in Communications, vol. 6, no. 9, pp. 1617-1622, Dec. 1998.
- [15] M. Kakida, Y. Tanigawa, and H. Tode, "Constraint-Based Distribution Method of In-Network Guidance Information in Content-Oriented Network," in Proc. IARIA AFIN 2014, pp. 39-47, Nov. 2014.