# Constraint-Based Distribution Method of In-Network Guidance Information in Content-Oriented Network

Masayuki Kakida Yosuke Tanigawa Hideki Tode
Dept. of Computer Science and Intelligent Systems
Osaka Prefecture University, Osaka, Japan
Email: {kakida@com., tanigawa@, tode@} cs.osakafu-u.ac.jp

***Abstract*—Lately, network usage is dominated by content distribution, and hence, Content Oriented Network (CON) has attracted attentions. CON has been actively studied at many research organizations, but it is still immature and difficult to establish clean-slate network. Therefore, our research goal is to establish a scalable and feasible architecture, which integrates the conventional IP network and a new content-oriented network framework. We use Breadcrumbs (BC) architecture as the base of integrated network but there still remains large overhead on managing and looking up forwarding table at each router. In this paper, we propose BC-Scoping on Popularity, which distributes the guidance information adaptively based on content popularity. Popularity is one of the criteria for the proposed BC-Scoping framework, which manages several BC's distribution scopes. The proposed method enables the network to use different forwarding policies based on content popularity, and brings the benefits of smaller overhead and better system performance. Finally, we demonstrate the effectiveness of the proposed method by extensive computer simulation, including the comparison with the currently operated Content Delivery Networks (CDN) approach, using state-of-the-art popularity model newly defined.**

***Index Terms*—Breadcrumbs, cache, content oriented network, Breadcrumbs+, BC-Scoping, popularity**

## I. INTRODUCTION

For several years, access loads on servers and network traffic are greatly increasing due to larger content-size and higher request-frequency in content distribution networks. To solve this problem, web caching approaches [1] [2] have been proposed, where network nodes cache copies of contents. On the other hand, the idea of content oriented network (CON) [3] [4] has attracted attention as a framework of new-generation network. This concept is derived from the viewpoint that users are interested not in where the content is but in what the content is; content distribution and retrieval dominate network usage now but network works on host-address oriented architecture designed several decades ago. Therefore, there are some researches that combine the ideas of "caching" with "content oriented network" [5] - [7]. Content-Centric Networking (CCN) proposed by Van Jacobson et al. [8] is also based on the similar concept, where both request and its' request are routed by a name of the content.

In CON, however, we encounter a serious scalability problem on content retrieval especially in a large-scale network like the Internet, where extremely wide variety of contents are distributed. In CON, packets are forwarded based not on address of nodes but on name of contents in network. A much larger number of contents than the one of network nodes increase costs of exchanging routing information and of looking up a routing table at each router.

CON has been actively studied at many research organizations, but it is still immature and difficult to establish clean-slate network. Therefore, our research goal is to establish a scalable and feasible architecture, which integrates the conventional IP network and a new content-oriented network. In terms of scalable routing on CON itself, not an active approach but a passive approach should be adopted for simplicity. Therefore, we established the routing method based on Breadcrumbs [7] [9] among researches on CON routing. Breadcrumbs is an architecture with guidance information (routing information) leading to a node holding a cached content, and log information to follow each content is created or updated only when the content is downloaded at routers on the download path but not at any other routers. BC's passive and simple approach allows us to make scalable and feasible CON autonomously in cooperation with cached contents in network. Although this routing approach has higher scalability than other ones like Forwarding Information Base (FIB) in CCN, BC-based routing still has large overhead on managing and looking up forwarding table at each router. One of the reasons is that most of routing information is not used effectively and searching a required information from the routing table is a costly task.

In the target integration architecture of IP and CON, we tackle to reduce various cost for networking (e.g., exchanging routing information, transferring data, frequency of forwarding operation, etc.) as much as possible. This contributes to enhance the comprehensive network system performance (e.g., content download time, robustness, etc.). Specifically, we propose BC-Scoping on Popularity, which distributes the guidance information of only popular contents. Popularity is one of the criteria for BC-Scoping framework which manages several BC's distribution scopes. As for unpopular contents, IP routing is definitely applied, which is desirable from the viewpoint of routing overhead. Such contents are requested less frequently, hence most of the guidance information for the contents are not used effectively. BC-Scoping on Popularity brings the following benefits.

1) It reduces the amount of BC entries in network, and then diminishes control overhead, including looking up

 ISBN: 978-1-61208-377-3 

a table, creating and deleting BC entries, etc.
2) It prevents the cache replacement of popular content caches by unpopular content ones, and then we can get better performance.

Though we already proposed BC-Scoping on Domain [10], this scoping is based on hierarchical network structure, and its goal is to localize content retrieval within a network domain and to reduce inter-domain communication. On the other hand, the goal of BC-Scoping on Popularity proposed in this paper is to reduce various cost for networking. Thus, it has completely different control policy compared with [10].

Finally, we demonstrate the effectiveness of the proposed method by the extensive computer simulation using state-of-the-art popularity model newly defined.

The main contributions of this paper are the following.

- We extend a specified criterion, which is the network domain in the earlier proposal, for BC-Scoping to general ones. We adopt content popularity as the representative criterion in this paper.
- The proposed method reduces system overhead, and specifically the lookup count of BC table is reduced to 37% of the naive BC system.
- We introduce newly formulated popularity model for generating requests.
- We quantitatively clarify the superiority of the proposed method in comparison to the most popular and widespread Content Delivery Network (CDN) approach.

The rest of the paper is organized as follows. Section II describes qualitative comparison of between our proposed method and other CON researches, and shows our earlier study. In section III, we propose BC-Scoping on Popularity. Section IV describes the performance evaluation of our proposed method, and the section consists of the followings: newly formulated popularity model (section IV-A), the simulation scenario and its results for comparison of our proposed method with other Breadcrumbs systems (seciton IV-B and ), and the simulation scenario and its results for comparison of our proposed method and CDN. Finally, section V shows our conclusion.

## II. Related works and Breadcrumbs

### A. Related Works

One of the main focuses in this paper is routing. Among routing methods in CON [3] - [8], our proposed routing method is based on Breadcrumbs taking a passive and simple approach, just logging a direction in which a content is forwarded, at each router. This simplicity reduces computational cost of processing or maintaining routing information compared with other schemes. Breadcrumbs's passive and simple approach allows us to make scalable and feasible CON autonomously in cooperation with cached contents in network. We use Breadcrumbs as the base of a solution against a scalability problem on searching for wide variety of contents.

CCN [8] adopts one of active approaches: a flooding-based approach. This approach is the most primitive idea of routing and will produce huge amount of traffic for retrieving a content. In contrast, users just issue a request for a content in Breadcrumbs and downloading the content makes a BC trail, which means a series of guidance information to the located cache.

DONA [6] adopts another one of active approaches: advertising routing information and forming routing trees. This approach achieves high efficiency and performance but needs to exploit information of network topology. Maintaining routing trees will require periodic exchange of network topology information and some computation. By contrast, Breadcrumbs does not use any detailed topology information with the exception of basic information about which domain each node belongs to, like subnet mask, especially for BC-Scoping on Domain. And thus, smaller overhead is required over the entire network.

PSIRP [4] adopts a routing scheme based on distributed-hash table (DHT). This approach achieves flat and scalable routing burden among routers but needs that participating nodes have to cooperate closely. On the other hand, in Breadcrumbs architecture, what each node does is just to log a direction and to forward packets. The nodes do not exchange extra information other than about IP routing. This simplicity provides easier management of nodes in network.

In the terms of efficient content distribution, CDN [11] [12] is the most practical and well-known techniques in the current Internet. CDN enables content publishers to make their contents widely available and to distribute the contents efficiently. In order to accomplish the efficient content distribution, Akamai [12], which is a first class example of commercialized CDN operators, monitors the state of service, network and servers through several ways including servers' periodic reporting and end-to-end agents' measuring. CDN architecture takes a centralized approach, which is appropriate for the operators to make decisions for controlling their system. On the other hand, Breadcrumbs architecture takes a decentralized approach, each node in the network works autonomously. In other words, CDN and Breadcrumbs have a fundamental difference in design concept, and they have a complementary relationship. We can then use the both of them simultaneously.

Another one of the important points in our research is that Breadcrumbs is a bridging architecture between the current network and the future network. There are some researches [4] – [6] [8] which adopt a clean-slate approach in order to establish content-oriented network, but it is hard to put the approach into practice in a large scale and in a short period by replacing current network. On the other hand, Breadcrumbs easily forms content-oriented network in cooperation with a network with any routing protocol (e.g., IP network). This characteristic ensures high feasibility of our Breadcrumbs-based approaches. We can use current network equipments supporting only IP if we install middleware for supporting

 ISBN: 978-1-61208-377-3 

TABLE I. BC ENTRY

| Attribute | Description |
|---|---|
| ContentID | Global file ID |
| UpHop | ID of node from which the file was forwarded |
| DownHop | ID of node to which the file was forwarded |
| DownloadTime | Time when the file passed through the node lastly |
| RequestTime | Time when the file was requested at the node lastly |

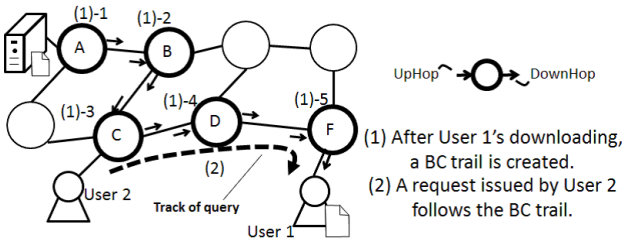



Fig. 1. An example of BC trail

Breadcrumbs on them, without fundamental and drastic replacement or re-installation of the current system [13][14]. In addition, even when Breadcrumbs architecture is partially deployed in the current IP network, the architecture works well by means of forming overlay network with simple IP tunneling technique [15]. Furthermore, we can operate content-oriented system with the support of the current IP system. In other words, we can create, improve and prepare the environment for future content-oriented system with operating the current Internet system.

### B. Breadcrumbs

In this section, we give an outline of Breadcrumbs [7]. Breadcrumbs architecture is stupidly simple from local view but efficient from global view to let application and network cooperate. Through primitive and simple processing such as logging a download path and caching a content, we can place, locate and obtain the content efficiently. We assume that the cache replacement policy follows Least Recently Used (LRU).

When a content is downloaded, each router on the download path makes a *breadcrumb (BC)* entry, which is a minimal information to route requests, and a cache of the content. Note that, because we assume more feasible network environment, core routers do not have any cache space, and only the *user end*: edge router, ONU, STB or user's PC, who downloaded the content, caches a copy of it. Each BC entry is composed of a 5-tuple data shown in Table I. Each of ContentID, UpHop, DownHop, DownloadTime and RequestTime has one value. When a request for a content encounters a BC entry for the content at a node on the way to an original content server by conventional IP forwarding, the request is routed to DownHop in the BC entry until it reaches a node with an intended cache (see Figure1). Thus, through tracing a series of BC entries, a request can follow the content downloaded previously. This series of BC entries is defined as *trail*. If DownHop or UpHop in a BC entry at a node is *null*, the node is at the end of the BC trail.

## III. PROPOSAL

In this paper, we focus on popularity of contents as a criterion for BC-Scoping. We propose BC-Scoping on Popularity, which distributes both of the guidance information and caches adaptively based on content popularity.

### A. Motivation

Name-based forwarding is one of the most important factors of content-oriented network. It provides us some advantages over the conventional location-based forwarding. One is smaller overhead to get the nearest copy of contents, and another one is robustness over dynamic change of content location by cache replacement. However, we should note that we can take these advantages only when we request popular content. "The nearest copy" implicitly means that there are many cached copies of a target content, but there are few ones of unpopular contents. "Dynamic change of locations" also implicitly means that the target content is requested many times and then many copies are frequently created and replaced in a variety of places, but unpopular ones are not so much.

Now, we can get a hint from [16]; taking different approaches based on target contents may provide better performance.

### B. BC-Scoping Framework

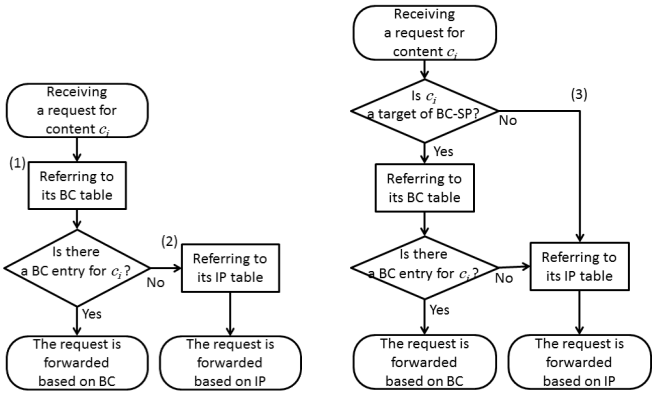
We have proposed BC-Scoping framework, which manages BC's distribution scope according to some criteria. We can reflect constraints on arbitrary criteria, including network domain, access count, content popularity and something. We would rather use simple metrics as criteria of BC-Scoping than complicated ones even if the metrics are too simple to bring us strictly optimal solution. This is because we believe that network system as infrastructure should be as simple as possible.

We already focused on hierarchical network topology and developed BC-Scoping on Domain [10], which limits the distribution scope of guidance information according to network domains. In other words, BC-Scoping on Domain enables or disables intermediate routers to create guidance information based on whether or not the destination node of a downloaded content belongs to the same domain as the router does, respectively. BC-Scoping on Domain promotes intra-domain communication so that it achieves small-hop content retrieval and reduction in traffic volume.

### C. BC-Scoping on Popularity

In this paper, we focus on popularity of contents as a criterion for BC-Scoping. We propose BC-Scoping on Popularity, which distributes both of the guidance information and caches for only popular contents. In the proposed system, when a router forwards a content, the router at first checks the popularity of the content. If the popularity rank is higher than the threshold rank, the router creates a BC entry or updates the

 ISBN: 978-1-61208-377-3




(a) Without BC-Scoping on Popularity (b) With BC-Scoping on Popularity

Fig. 2. Router's request handling flow in BC system

corresponding entry after forwarding the content. Otherwise, the router does nothing after forwarding. In addition, when *user end* receives a content, the user end also checks the popularity of the content. If the popularity rank is higher than the threshold rank, the user end caches a copy of the content. Otherwise, the user end does nothing. The most simplest way to check the above condition is to utilize a packet header. Specifically, in advance, each origin server adds popularity information to contents. Then, according to the popularity information attached to target content, active or inactive flag is changed in the header of transferred packets. It requires the minimal cost.

### D. Benefits of BC-Scoping on Popularity

BC-Scoping on Popularity provides the following benefits.

First, explicit separation of forwarding policy reduces overhead of looking up forwarding table at each router. In the naive BC system, when a router receives a request, the router always refers to its BC forwarding table at first for name-based forwarding regardless of content popularity ((1) in Figure 2(a)). In most cases, there is no cached copy of unpopular contents in network, and there is no corresponding BC entry, either. The router eventually refer to IP table ((2) in Figure 2(a)). In short, routers have to refer to two different forwarding tables for unpopular contents in most cases. With BC-Scoping on Popularity, when a router receives a request for unpopular contents, the router skips reference to BC table, and then refers to IP tables at first ((3) in Figure 2(b)). Though the router may refer to two different forwarding tables for popular contents, the size of BC table is smaller and we have a lower possibility of no cached copy in network compared with the case without BC-Scoping on Popularity. From these assumptions, BC-Scoping on Popularity can reduce system overhead.

Second, it prevents frequent replacement of a cache of popular content by the one of unpopular content. As a result, more caches of popular contents are distributed in network. Since cached copies of contents with high popularity are clearly more valuable in terms of traffic reduction and small-hop content retrieval, we will get better performance on those metrics.

## IV. Evaluation

We conducted computer simulations on 2 scenarios with newly developed popularity distribution of request for contents. The distribution used in the simulations is developed, by combining 2 characteristics of content popularity [18][20]. 2 scenarios are the following:

1) the proposed method vs BC+
2) the proposed method vs CDN.

In the former evaluation, we compare the proposed method with the conventional Breadcrumbs methods. In the latter evaluation, we also compare the proposed method with the simplified model of CDN, which play the important roles in the current content distribution.

How to get the popularity information of each content is out of scope in this paper, and we assume the information is attached in each content beforehand. From the aspect of implementation, one of promising ways is that, according to request frequency, content servers attach the flag indicating whether in-network guidance information should be logged via the traversing route to each content or not.

### A. State-of-the-art popularity distribution of request for contents

There are some observations on users' request pattern [18] –[20]. In [18], YouTube's web pages were observed, and it was showed that video popularity almost follows a Zipf distribution with an exponential cutoff and also has fetch-at-most-once like behavior. Fetch-at-most-once [19] behavior makes a gap between actually traced data and a logical Zipf distribution in high popularity rank. [20] reported this gap can be expressed using Zipf-Mandelbrot's law instead of Zipf's law. Under this distribution, the probability of an occurrence of a request for the content with the $k$-th popularity is defined by the following equation:

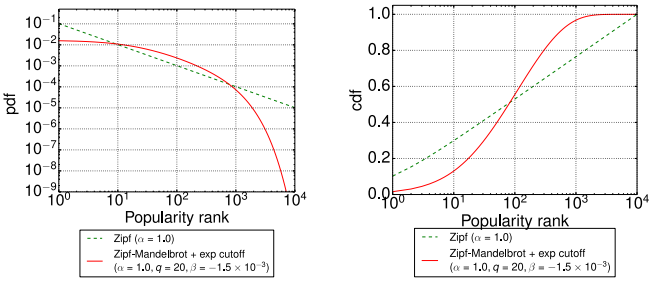
$$f(k; N, q, \alpha) = \frac{1}{(k+q)^{\alpha}} \times \frac{1}{\Sigma_{i=1}^{N} \frac{1}{(i+q)^{\alpha}}}. \quad (1)$$

$N$ is the total number of ranks, $q$, $\alpha$ are the parameters which determine the shape of distribution. The above two were independently discussed so far. Therefore, we developed a new distribution model by combining two factors, Zipf-Mandelbrot with exponential cutoff distribution. Under this distribution, the probability of an occurrence of a request for the content with the $k$-th popularity is defined by the following equation:

$$g(k; N, q, \alpha, \beta) = f(k; N, q, \alpha) \times e^{\beta k} \times a. \quad (2)$$

$\beta$ is the parameter to determine the shape of distribution. $a$ is the parameter for normalization.

 ISBN: 978-1-61208-377-3 




(a) pdf (b) cdf

Fig. 3. Dist. of requests for contents

We show the probability density function (pdf) and the cumulative distribution function (cdf) of this distribution in Figures 3(a) and 3(b), respectively.

### B. Simulation scenario (vs BC+)

We set each parameter as shown in Table II and other settings are as follows. Note that we assume that contents are cached on not the core routers but the edge of the network, *user end* which includes edge router, users' storage, ONU, or STB, unlike original Breadcrumbs [7], for higher feasibility.

- Network topology

We generate a flat router-network based on the Waxman model [17]. The parameters for generating each network are as follows: A edge length of a square is 1000, $\alpha_{waxman} = 0.3$ and $\beta_{waxman} = 0.05$. Every router is connected with five users and the location of each server is chosen in a uniformly random manner. In the generated topology, a packet can be transferred between any two routers in 18 hops at most.

- Requests for contents

Each user determines which content to request at random following the distribution described in Section IV-A, where $N = 10000, \alpha = 1.0, q = 20, \beta = -15/N$. Each user requests a content at the independent, identical, and exponentially-distributed random interval.

- Packet size and delay

According to the paper of the original Breadcrumbs [7], we assume that the network is not congested, and each node has an enough routing capability to traffic load so that some parameters in the system are constant for simplicity. Size of packets is consistently 1,500 Byte. A request or a control packet consists of 1 packet. On the other hand, a content consists of 768,000 packets. This content size is nearly equal to the size of a content with a bit rate of 5 Mbps and length of 30 min. A delay for a packet to travel to the adjacent node is always 2.3 ms, which includes delays for processing, queuing, propagation and transferring. This delay means that a throughput of any connection is consistently 5 Mbps.

- Compared methods

TABLE II. PARAMETERS

| parameter | value |
|---|---|
| # Routers | 1000 |
| # Users | 5000 |
| # Servers | 50 |
| # Contents | 10000 |
| Cache capacity per user | 2 |
| Interval of request generation per user | 2000 |
| $T_f$ for purge of BC | 90000 |

We compared the following 3 methods.

1) BC+ : Naive BC+ system without BC-Scoping
2) BC+ (rate) : BC trails and caches are created at a constant rate.
3) BC-S (pop) : Proposed method.

In this simulation, "popular" contents mean "top 1%" popular contents among all, and we call those contents $target$ contents. BC trails and caches for only the target contents are created in network.

We prepare a method $BC+(rate)$ as an object for comparison in terms of reduction in overhead. In BC+ (rate) system, BC trails are created at a constant rate, where each server and cache node decide whether or not to create BC trails based on access count. This method is the simplest way to reduce overhead. We can consider this frequency as a criterion for BC-Scoping. We set the rate to 55.5% in this simulation because requests for top 1% popular contents occupy 55.5% of all generated requests. This means that we have the same number of opportunities for BC trail creation in BC+ (rate) and BC-S (pop) (see Figure 3(b)).

We should note that we use Breadcrumbs+ [9] instead of Breadcrumbs for the following evaluation, because the original BC [7] has a routing loop problem due to a broken BC trail, where requests are transferred forever within a series of routers with forming a loop of route and cannot reach the intended contents. The original Breadcrumbs is too simple to solve the problem, and hence we revised attributes in a BC entry and improved the invalidation operation of BC. We discussed the detail of this problem in [9].

### C. Simulation results

We show the results in Table III, and in Figures 4 - 10. We used the following evaluation metrics: BC operation count, lookup count of routing table, request hop count, content hop count and server access ratio.

With these results, we can obtain the following considerations. The first two considerations are on overhead, the latter two are on performance trade-off.

*1) BC operation count:* BC operation count is the total number of the events operating BC entries at each router. Smaller BC operation count means that each router spends smaller resources to manage Breadcrumbs system.

Figure 4 shows that BC+ (rate) reduces total BC operation count to approximately 56% compared with BC+. This reduction in operation count is proportional to that in creation

 ISBN: 978-1-61208-377-3

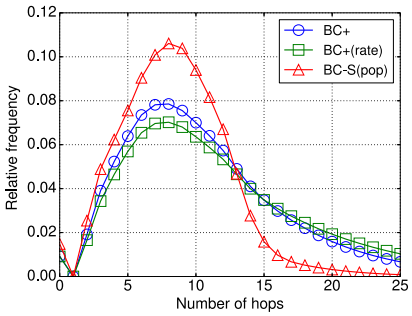



Fig. 6. Dist. of request hop count

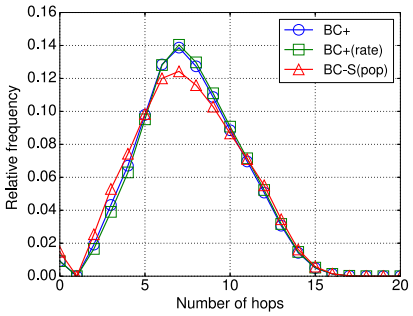



Fig. 8. Dist. of content hop count

TABLE III. MEAN VALUES

| Method | Request hop count | Content hop count | ServerAccess Ratio |
|---|---|---|---|
| BC+ | 10.93 | 7.62 | 0.132 |
| BC+(rate) | 12.48 | 7.71 | 0.109 |
| BC-S (pop) | 8.47 | 7.54 | 0.453 |

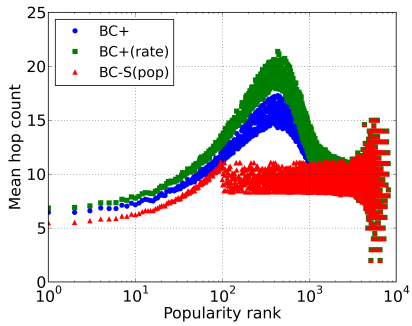



Fig. 7. Dist. of mean request hop count by popularity rank

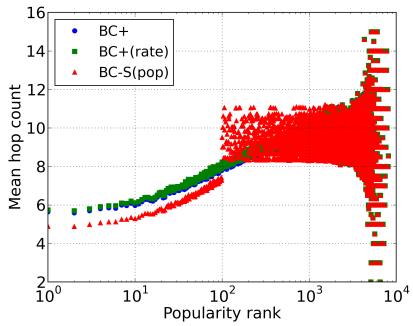



Fig. 9. Dist. of mean content hop count by popularity rank

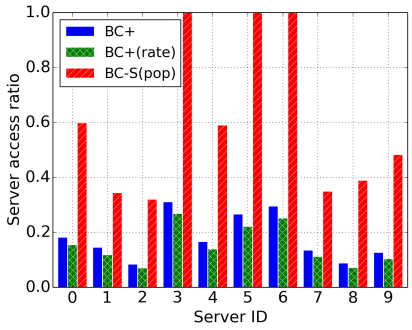



Fig. 10. Server access ratio

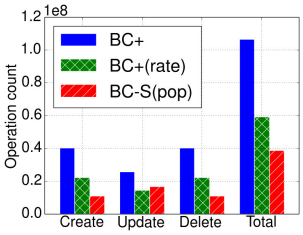



Fig. 4. BC operation

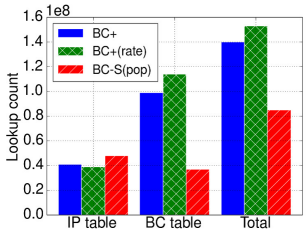



Fig. 5. Lookup count of routing table

opportunities of BC trail and cache. On the other hand, our proposed method of BC-S (pop) reduces total BC operation count to approximately 37% compared with that of BC+ even though requests for popular contents ranked in top 1% account for 55.5% of all generated requests. The reason is the following. In all the three methods, cache replacement causes to delete a part of BC entries for the evicted content, and then to create a new BC entries for the newly cached content. In BC-S (pop), however, an opportunity of cache replacement is reduced, and then just updating BC entries, which means refreshing expiration times of the BC entries, is enough to maintain the system.

*2) lookup count of routing tables:* Lookup count is the total number of events referring to routing table at each router.

Figure 5 shows that BC-S (pop) has the lowest total lookup cost of routing table which is approximately 61% compared with BC+. Focusing on lookup count in IP table, BC-S (pop) increases lookup cost to approximately 117% of BC+ whereas BC+ (rate) decreases to approximately 95%. Focusing on that in BC table, BC-S (pop) decreases lookup cost to approximately 37% of BC+ whereas BC+ (rate) increases to approximately 115%. We should note that BC-S (pop) increases the lookup count in IP table but decreases the total lookup count, whereas BC+ (rate) increases the total lookup cost. These results mean that we cannot reduce lookup count through just reducing an opportunity of BC entry creation without ingenuity.

*3) Hop count:* Request hop count is the number of links that a request traverses before it reaches the target content from the requesting user. Content hop count is the number of links that a content traverses before it reaches a requesting user from a node providing the target content.

Table III shows that BC-S (pop) reduces hop count of both request and content. Figure 6 shows that BC-S (pop) increases the amount of request with smaller hop count (13 or less) and decreases with larger hop count (14 or more), and Figure 8 shows that BC-S (pop) also increases the amount of content with extremely smaller hop counts (5 or less). These results are due to the two reasons shown in Figures 7 and 9. One is that popular contents ranked in top $10^2$th are retrieved more frequently from near caches with smaller hop count in BC-S (pop) compared with the other methods, because BC-S (pop) increases the amount of cached copies of popular contents. The other is that unpopular contents ranked under $10^2$th are retrieved from servers mainly with between 8 and 11 hop counts in BC-S (pop). These hop counts are smaller than ones

 ISBN: 978-1-61208-377-3 

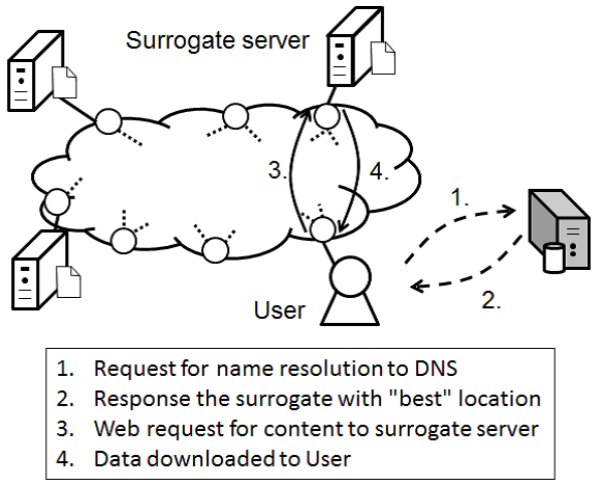



Fig. 11. CDN system applying DNS redirection

from servers and caches in the other methods, because BC-S (pop) does not create caches and BC trails of unpopular contents, which guides requests to a far cache.

*4) Server access ratio:* Server access ratio is defined as

$$\frac{\text{The number of requests which reached a server}}{\text{The number of all the generated requests}}.$$

As server access ratio becomes lower, more requests reach cache-nodes, and this results in higher cache utilization.

Figure 10 shows that BC-S (pop) increases server access because requests for unpopular contents are always forwarded to server in IP routing. On the other hand, Table III says that almost all requests for popular contents reached caches of the corresponding contents since requests of popular contents occupied 55.5% of all generated requests in this simulation setting shown in Figure 3(b). We can improve this metric through just loosing the limitation on scoping by expanding the target range, but it may cause to worsen other metrics including hop count described above. In short, there is performance trade-off relationship between server access ratio and other metrics.

### *D. Simulation scenario (vs CDN)*

In this section, we illustrate the CDN model, which we developed and used in the simulation to evaluate the proposed method by comparing with the current major content distribution technique.

The core idea of CDN [11] is to replicate contents into some *surrogate servers*, which provide the replicated contents instead of the *origin server*, and to redirect web requests for the contents to one of the surrogate servers. One of the typical implementations is shown in Figure 11.

In order to make a CDN model, we should consider the following topics: surrogate placement, surrogate selection on request redirection, object selection on replication, surrogate selection on replication, and decision on amount of replication.

- surrogate placement

We applied *greedy algorithm* [21] to determine where to place surrogate servers. The idea of this algorithm is as follows. At first, among all $N$ routers, the first surrogate server is placed on a router to which the summation of the numbers of hops from all users is minimum. Then, the second surrogate is placed on another router so that the summation of the numbers of hops from all users to the nearest surrogate of all becomes minimum. We iterate this process until $M$ surrogate servers are placed. Note that in this simulation scenario $N$ is 1000 as described in Table II and $M$ is set to 1 and 5, as described below.

In the real world, the number of potential sites depends on where we can place data center facilities and what ISP provides the Internet connection for the data center.

- surrogate selection on request redirection

We simplified *DNS redirection* technique in our simulation. The core idea of this technique is illustrated in Figure 11. When the DNS receives a name resolution request for a content, it resolves the request so that a web request is forwarded to the “best” surrogate server. In order to realize this best selection, an actual CDN operator monitors the state of service, network and servers [12]. In our simulation, we did not consider the monitoring overhead in order to simplify the scenario; we just redirect web requests to the nearest surrogate server according to pre-computed topology information. In other words, we assume that all surrogate servers and network links have the unlimited capacity.

Three topics: object selection on replication, surrogate selection on replication and decision on amount of replication, mean what content should be replicated, where a replica of the content should be stored, and how many replicas should be made, respectively. We simplified these 3 topics in our simulation; all surrogate servers statically store each replica of all kinds of contents under the CDN’s control. Although there are some strategies [11] about these topics, our assumption enables us to reduce simulation parameters.

In the earlier part of this sub-section, we described the simplified CDN model, which we developed in this paper. We then show the other parameters in the rest part.

- Compared methods

We compared the following 3 methods.

1) IP CDN (1) : the simplified CDN model which has 1 surrogate in network.
2) IP CDN (5) : the simplified CDN model which has 5 surrogates in network.
3) BC-S (pop) : Proposed method.

In this simulation, target contents are also “top 1%” popular contents among all like in Section IV-B. We replicated only the target contents in the surrogates in *IP CDN(1)* and *IP CDN(5)*.

We set the other parameters same as setting in Section IV-B.

 ISBN: 978-1-61208-377-3 

TABLE IV. MEAN VALUES

| Method | Request | Data |
|---|---|---|
| IP CDN(1) | 8.45 | 8.45 |
| IP CDN(5) | 7.30 | 7.30 |
| BC-S(pop) | 8.47 | 7.54 |

TABLE V. FAIRNESS INDEX OF CONNECTION COUNT ON LINKS

| Method | Router NW |
|---|---|
| IP CDN(1) | 0.1209 |
| IP CDN(5) | 0.3341 |
| BC-S(pop) | 0.4969 |

### E. Simulation results (vs CDN)

We show the results in Tables IV and V, and in Figures 12 - 17. We used the following evaluation metrics: request hop count, content hop count, connection count on link in the router network and cumulative distribution function (CDF) of upload connection count on each surrogate.

*1) Hop count:* With respect to request, BC-S(pop) requires more hop count for requests than IP CDN (n). This is because BC trail sometimes causes a long travel of request described as a long tail in Figure 12. In IP CDN (n), location of the content is fixed at each surrogate, and the request redirection provides the smaller hop content travel for the target contents (see Figures 13 and 15). On the other hand, BC-S (pop) and IP CDN (5) have similar content hop count. In BC-S (pop), distributed large amount of user-cache with higher popularity occurs small hop travel of contents.

*2) Connection count on links:* Table V shows Fairness Index of connection count on each link which interconnects routers. In Table V, BC-S(pop) achieves higher fairness of connection count on links than each IP CDN (n) does. This is because Breadcrumbs architecture makes a distributed system whereas CDN makes a centralized system, even though the both system are on a client-server model. Figure 16 shows complementary cumulative distribution function (CCDF) of connection count on links. In Figure 16, each IP CDN (n) forms a straighter shape than BC-S(pop) like the Pareto distribution. This means that a few links near surrogate servers are extremely crowded but most of links are not in IP CDN (n). In BC-S(pop), caches of target contents are distributed everywhere and this promotes better load-balancing compared with IP CDN (n). This difference on fairness arises from the difference of fundamental policy on architecture design; CDN takes a centralized approach whereas Breadcrumbs does a decentralized approach.

*3) Upload connection count on each surrogate server:* Figure 17 shows that the distribution of connection count on each surrogate server follows normal distribution and the connection count is stable. In our simulation, each connection consumes 5 Mbps on the link bandwidth and each access link for surrogate servers may be able to accept these connections at the same time, but some surrogate may not be because it requires expensive equipment in terms of workloads. Comparing IP CDN (1) with IP CDN (5), multiple surrogates placed at distributed locations achieve load-balancing but increasing a new surrogate at another place is not easy. On the other hand, our proposed method based on Breadcrumbs uses distributed and totally large user-cache space, and upload connection count on each user in BC-S (pop) is approximately 1.4. This is extremely smaller than that on each surrogate, and this difference on workloads also arises from the difference of the fundamental policy on architecture design.

*4) IP CDN (5) vs IP CDN (1):* IP CDN (5) outperforms IP CDN (1) in the all metrics because of more surrogates without any cost. More surrogates require more management cost but we ignore the cost in this simulation for simplicity. In addition, we should note that more surrogates at different locations in this simulation means more data center facilities in the real world. It is therefore easier and more feasible to enhance the existing surrogates than to increase new surrogates.

## V. Conclusion

Our research goal is to establish a scalable and feasible architecture, which integrates the conventional IP network and a new content-oriented network. In terms of scalable routing on CON itself, not an active approach but a passive approach should be adopted for simplicity. Therefore, we established the routing method based on Breadcrumbs among researches on CON routing. Breadcrumbs is an architecture with guidance information (routing information) to a node holding a cached content. Breadcrumbs's passive and simple approach allows us to make scalable and feasible CON autonomously in cooperation with cached contents in network. In this paper, we focused on popularity as a criterion of BC-Scoping framework, and we proposed BC-Scoping on Popularity, which distributes the guidance information of the only popular contents. As for unpopular contents, IP routing is definitely applied, which is desirable from the viewpoint of routing overhead. Such contents are requested less frequently, hence most of the guidance information for the contents are not used effectively. The proposed method clearly separates the forwarding policy based on popularity of content so that it promotes cache utilization of popular contents and server utilization of unpopular contents. This method brings the following benefits: reduction in system overhead including BC operation cost and lookup cost of BC table, small-hop content retrieval. We should note that the latter benefit is in return for increase of server access because there is trade-off relationship between hop count and server access. Finally, we conducted simulations and demonstrated the effectiveness of our proposal. We can highlight that we introduced the state-of-the-art requesting model based on content popularity for evaluation, and that we quantitatively clarified the superiority of the proposed method in comparison to most popular and widespread CDN approach. We have the following two tasks as our future works. First, we will consider a way to estimate popularity of contents at each servers or cache node. One idea is that each server and cache node can count access frequency at itself autonomously, and another is that we prepare dedicated servers for managing content popularity. Second, we will implement Breadcrumbs architecture [13] and evaluate the effectiveness of our proposal in the real world.

 ISBN: 978-1-61208-377-3 

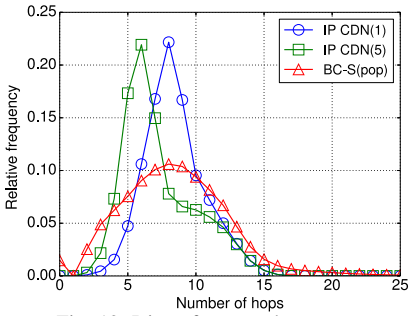



Fig. 12. Dist. of request hop count

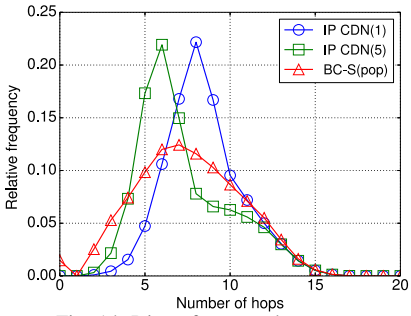



Fig. 14. Dist. of content hop count

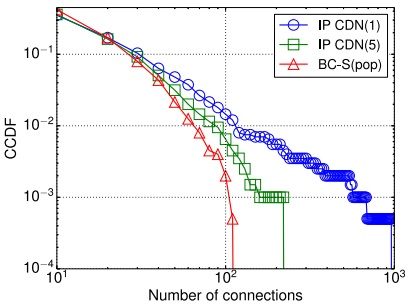



Fig. 16. CCDF of connection count on links

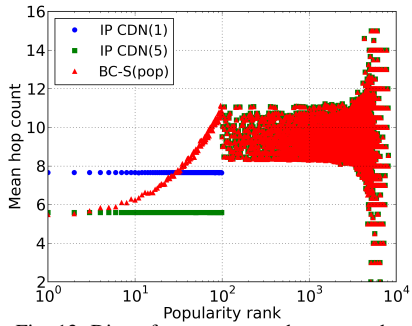



Fig. 13. Dist. of mean request hop count by popularity rank

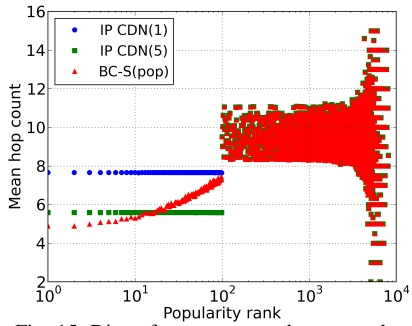



Fig. 15. Dist. of mean content hop count by popularity rank

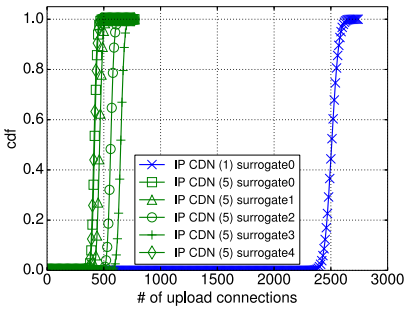



Fig. 17. CDF of upload connection count on surrogates

## ACKNOWLEDGEMENT

This research was supported in part by National Institute of Information and Communication Technology (NICT), Japan. We also appreciate insightful comments of Prof. J. Kurose and Dr. E. J. Rosensweig, University of Massachusetts, and the other project members.

## REFERENCES

[1] G. Barish and K. Obraczka, "World wide web caching: Trends and techniques," IEEE Communications Magazine, vol. 38, no. 5, May 2000, pp. 178–184.

[2] J. Wang, "A Survey of Web Caching Schemes for the Internet," ACM SIGCOMM Computer Communication Review, Vol. 29, Issue 5, Oct. 1999, pp. 36–46.

[3] J. Choi, J. Han, E. Cho, K. Kwon, and Y. Choi, "A Survey on Content-Oriented Networking for Efficient Content Delivery," IEEE Communications Magazine, vol. 49, no. 3, Mar. 2011, pp. 121–127.

[4] D. Lagutin, K. Visala, and S. Tarkoma, "Publish/Subscribe for Internet: PSIRP Perspective," Towards the Future Internet - A European Research Perspective, 2010, pp. 75-85.

[5] I. Stoica, D. Adkins, S. Zhuang, and S. Shenker, "Internet Indirection Infrastructure," IEEE/ACM Transactions on Networking, vol. 12, no. 2, Apr. 2004, pp. 205–218.

[6] T. Koponen et al. "A Data-Oriented (and Beyond) Network Architecture," in Proc. ACM SIGCOMM 2007, Oct. 2007, pp. 181–192.

[7] E. J. Rosensweig and J. Kurose, "Breadcrumbs: efficient, best-effort content location in cache networks," in Proc. IEEE INFOCOM 2009, Apr. 2009, pp. 2631–2635.

[8] V. Jacobson et al, "Networking named content," in Proc. ACM CoNEXT 2009, Dec. 2009, pp. 1–12.

[9] M. Kakida, Y. Tanigawa, and H. Tode, "Breadcrumbs+: Some Extensions of Breadcrumbs for In-network Guidance in Content Delivery Networks" in Proc. SAINT 2011, July 2011, pp. 376–381.

[10] M. Kakida, Y. Tanigawa, and H. Tode, "Distribution Method of In-network Guidance Information for Inter-AS Content-Oriented Network Topology," Proc. WTC 2012 Poster Session, Mar. 2012.

[11] A. Passarella, "Review: A survey on content-centric technologies for the current Internet: CDN and P2P solutions," Computer Communications, vol. 35, no. 1, Jan. 2012, pp. 1–32.

[12] J. Dilley et al, "Globally Distributed Content Delivery," IEEE Internet Computing, vol.6, no.5, 2002, pp. 50–58.

[13] T. Yagyu, M. Kakida, Y. Tanigawa, and H. Tode, "Prototype Development of Active Distribution of In-network Guide Information for Contents Delivery," in IEICE Technical Report, vol. 111, no. 468, NS2011-211, Mar. 2012 (in Japanese), pp. 179–184.

[14] T. Yagyu, "Synergic Effects among Plural Extensions of Breadcrumbs for Contents Oriented Networks," Proc. AFIN 2013, Aug. 2013, pp. 35–42.

[15] T. Tsutsui, H. Urabayashi, M. Yamamoto, E. Rosensweig, and J. Kurose, "Performance Evaluation of Partial Deployment of In-Network Query Direction Method in Content Oriented Networks," in Proc. Future Net V, IEEE ICC 2012, Canada, June 2012, pp. 7386–7390.

[16] R. Chiocchetti, D. Rossi, G. Rossini, G. Carofiglio, and D. Perino "Exploit the Known or Explore the Unknown? Hamlet-Like Doubts in ICN," Proc. ICN'12, ACM SIGCOMM 2012, Aug. 2012, pp. 7–12.

[17] B. M. Waxman, "Routing of multipoint connections," IEEE J. Selected Areas in Communications (Special Issue on Broadband Packet Communication), vol. 6, no. 9, Dec. 1998, pp. 1617–1622.

[18] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, and S. Moon, "Analyzing the video popularity characteristics of large-scale user generated content systems," IEEE/ACM Transactions on Networking, vol.17, no.5, Oct. 2009, pp. 1357–1370.

[19] K. P. Gummadi et al, "Measurement, modeling, and analysis of a peer-to-peer file-sharing workload," Proc. ACM SOSP 2003, Dec. 2003, pp. 314–329.

[20] O. Saleh and M. Hefeeda, "Modeling and Caching of Peer-to-Peer Traffic," Proc. ICNP 2006, 2006, pp. 249–258.

[21] L. Qiu, V. Padmanabhan, and G. Voelker, "On the placement of web server replicas," Proc. IEEE INFOCOM 2001, 2001, pp. 1587–1596.

 ISBN: 978-1-61208-377-3