Dispersing Content Over Networks in Information-Centric Networking

Noriaki Kamiyama, *Member, IEEE,* Masayuki Murata, *Member, IEEE*

***Abstract*—Information-centric networking (ICN), a new network architecture for efficiently delivering content, has been widely investigated recently. In ICN, cache memory is implemented at each router, and content items are routed in the network by using the content name as the locator determining the destination. The caching strategy that determines the content to be cached at each router strongly affects the cache hit ratio and flow hop length, and it is important to efficiently utilize limited cache resources by avoiding duplicated caching of the same content among routers located closely. However, no caching strategy aiming at dispersing content over networks has been investigated. In this paper, we propose spatially dispersed caching (SDC), which is a caching strategy dispersing content by assigning a binary ID to each router and limiting the cache targets at each router to content with names whose hash value coincides with the router ID. Through computer simulations using backbone networks of actual ISPs in the USA, we show that SDC reduced the average hop length at cache hit by about 50% to 90% compared with the existing caching strategies. Moreover, we show that SDC improves the sustainable ratio of content acquisition in large-scale failures of routers by about 25% to 200% compared with the existing caching strategies.**

***Index Terms*—ICN, autonomous, caching strategy, dispersing**

I. Introduction

Traffic generated by delivering video content including user generated content (UGC), e.g., YouTube, and rich content produced by content providers, e.g., movie and dramas, has dominated a large part of traffic on the Internet. Packets are routed by using IP addresses as locators, so the overhead for resolving the IP addresses of destination hosts from the content names is indispensable. Therefore, as a new network architecture efficiently delivering content without this overhead, information-centric networking (ICN), which caches content at routers and routes packets using the content name, has attracted wide attention [6]. To realize the idea of ICN, various networks, such as TRIAD [10], content-centric networking (CCN) [14], the data-oriented network architecture (DONA) [16], and named data networking (NDN) [31], have been proposed [29].

In many proposals related to the ICN, users who want to acquire content send an Interest, i.e., packets requesting content, destined for an origin server having the original content, and routers transfer the Interest by using the content name as the locator. A content store that caches content[1] is provided at routers, and routers cache content item received [17]. Routers on the route of the Interest packets discard the Interest received without transferring it to the next-hop router and send the content to the requesting users. By using the ICN, we can avoid the overhead of resolving the IP address from the content name, and we can expect to reduce the transmission delay and network load because content can be delivered from a location close to users [6].

Routers need to determine content cached autonomously, and the caching strategy strongly affects the hop length of delivery flows and the link load. To effectively improve the ratio and reduce the hop length by efficiently utilizing limited cache resources, it is important to avoid caching the same content at many routers nearby and distribute the identical content at spatially dispersed locations [24]. Spatially dispersed caching is also important to improve the sustainability of acquiring content in large-scale failures of routers. Although one approach to disperse the locations of caching identical content is obtaining the complete information of cached content at all routers in a network by repeatedly exchanging the information of cached content between adjacent routers [27][28], the processing load at routers will seriously increase. To improve the scalability and reduce the cost of ICN routers, it is desirable to realize the spatially dispersed distribution of content as a result of an autonomous caching decision at each router without exchanging information between routers. However, no autonomous caching strategy with the aim of realizing this has been investigated.

Therefore, in this paper, we propose spatially dispersed caching (SDC), which spatially disperses the locations for caching identical content by autonomous caching judgement at routers[2]. In SDC, each router is assigned a unique ID and caches only content with a hash value obtained from the content name that agrees with the router ID. By assigning router IDs with many different bits for nearby routers, identical content is cached at spatially dispersed locations. Because user requests concentrate on a small number of popular content in general, we can expect to improve the cache hit ratio by storing popular content at many routers. Therefore, in SDC, each router autonomously classifies content items into multiple groups on the basis of the popularity and decreases the number of bits matched for popular content when comparing the router IDs and the hash values of content names in order to increase the number of copies cached at routers for popular content. Using SDC, we can expect the following merits.

- Improves the cache hit ratio due to avoiding the waste of

N. Kamiyama is with the Faculty of Engineering, Fukuoka Univer-sitry, Fukuoka, 814-0180 JAPAN. M. Murata is with the Department of Information Science, Osaka University, Osaka 565-0871, JAPAN. e-mail: kamiyama@fukuoka-u.ac.jp, murata@ist.osaka-u.ac.jp

[1]Although content is divided into multiple chunks and is cached in the unit of chunks, we describe the unit of caching as content for simplicity in this paper.

[2]A shorter version of this manuscript was presented in [15].

cache memory caused by duplicate caching of identical content at multiple routers in nearby areas

- Reduces the hop length when delivering content from routers, i.e., at cache hit, due to decreasing the deviation of distance to each content item from each router
- Reduces the average hop length among all requests as the result of improving the cache hit ratio and reducing the hop length at cache hit
- Improves the sustainability in acquiring content in large-scale failures of routers due to spatially dispersing the cached location of content

In Section II, we briefly summarize the related works. After describing the mechanism of the proposed SDC in Section III, we give details on the methods for assigning IDs to routers and grouping content into popularity groups in Sections IV and V, respectively. In VI, we show numerical results, and finally, we conclude this manuscript in Section VII.

II. Related Works

In ICN, routers store content received at the transmission bit rate of links, so high-speed and high-cost memory, i.e., TCAM and SRAM, is used to implement the content store at routers [21]. Therefore, the storage capacity with ICN is much smaller than that with CDN, so we need to carefully design and manage the content store in ICN. The content items remaining in the content store are determined by both the method determining which content items are stored in the cache among received content items and the method selecting the content items to be removed when the storage capacity of cache memory is full at the insertion of new content items. In this paper, we call the former method a *caching strategy* and the latter method a *cache-replacement policy*.

In many architectures of ICN, the Interest is transmitted on the route destined for the origin server, which is called the *default path*, and content is sent from the router closest to the requesting user on the default path among routers caching a copy of the requested content. If no routers on the default path cache the requested content, the origin server of the content sends the content. The content is delivered to users on the default path where the Interest is transmitted in the reverse direction. The most widely used caching strategy in ICN is transparent en-route caching (TERC), which caches all the content items received at all the routers on the delivery route [17]. As the cache-replacement policy, least recently used (LRU) which removes the content with the longest elapsed time after the final request, is most widely used [27].

Besides TERC, various caching strategies have been proposed for ICN, and we can classify these methods into the two approaches. In the first approach, only routers on the default path are candidates caching content, whereas all the routers in the network are candidates caching content. For example, Psaras et al. proposed the ProbCache, in which content is cached at each router on the default path with a probability based on the relative position between the requesting user and the origin server so that content was cached with a higher probability at routers closer to users [22]. Fayazbakhsh et al. showed that we can obtain a reasonable performance of caches even if content was cached at only edge routers through a computer simulation using tree network topologies [7]. Moreover, Laoutaris et al. proposed leave copy down (LCD), which caches content only at the next-hop router from the sending router of content [18], and Cho et al. also proposed WAVE, which applies a similar policy in the unit of chunks [5]. Cho et al. also proposed UniCache, in which routers on the default path cache content with the probability of one divided by the hop length of the default path, so content is cached at only one router on average [5]. More recently, Ioannedis et al. proposed adaptive caching strategies which minimizes the routing cost of content, i.e., the sum of link costs [12], and they also proposed a method of jointly optimizing the routes and content location minimizing the total routing cost [13].

In general, content popularity is not uniform, and user requests concentrate on a small number of popular content items [30], so popular items are cached at many routers duplicatedly. However, from the viewpoint of reducing the hop length of delivery flows, the effect of caching identical content at multiple routers in nearby locations is small. It is important to cache many content items while sustaining the effect of caches by placing copies of each content item at spatially dispersed locations [27]. However, these existing caching strategies do not explicitly avoid duplicate caching in nearby areas, so it is difficult to avoid wasting cache resources caused by caching the same content items at many caches located closely.

As an approach explicitly avoiding duplicated caching in nearby areas, Rezazad et al. proposed limiting the cache positions on the default path to one router [23]. In other words, parts close to the head of the content are cached at only routers close to the user router, whereas parts close to the tail of the content are cached at only routers close to the source router. Moreover, Saino et al. [25] and Saha et al. [24] proposed assigning the range of hash values of content names without overlap to routers and caching content only at routers whose assigned range includes the hash value of the target content. As a result of limiting the location for caching each content item to just a single router, we can avoid the duplicate caching of identical content and expect to improve the cache hit ratio. However, the default path needs to always traverse the router that was the caching candidate of the target content item, so the hop length of delivery flows is largely increased. Moreover, all these methods did not consider the popularity of content items, so not only unpopular content items but also popular content items are cached at just a single router, and this will degrade the cache hit ratio because content requests concentrate on few popular content items.

The second approach, i.e., enlarging the caching location to the outside of the default path, aggressively isolates the location of caching among content items by exchanging the information of cached content between adjacent routers. For example, Wang et al. proposed optimizing the position for caching each content item by periodically exchanging the information of cached content among routers and solving the greedy algorithm [27]. Moreover, Xie et al. formalized the optimization problem by determining the routes and the cached locations of each content item simultaneously so that the maximum link utilization is minimized [28]. Although we can isolate the positions caching each content item by using these methods, the network needs to solve the optimization problem by exchanging the information of cached content among routers. Although we can reduce the amount of traffic generated in exchanging information among routers by using Bloom Filters [20], routers still need to maintain the information of cached content at other routers, so the processing load and memory cost at routers are increased.

III. Spatially Dispersed Caching (SDC)

A. Overview

In this paper, we propose spatially dispersed caching (SDC), which realizes the spatially dispersed deployment of content at routers as a result of autonomous judgement of caching content at each router without exchanging information between adjacent routers. We assume that a core network operated by a single ISP in which the ICN function is introduced at all the routers[3], and a single authority manages all the routers and executes the same caching strategy at all the routers. We also assume that the origin servers owned by content providers are accommodated into any routers. For N, the number of routers in the network, let K denote the minimum integer satisfying $2^K \geq N$. We assign each router an ID with K bits without duplication. The principal mechanism of the proposed SDC can be summarized as the following three points.

- **Geographically sparse assignment of router ID**: Each router is assigned a binary ID of K bits without duplication so that routers in nearby areas are assigned IDs with different values in the upper digits (see Section IV for the detailed algorithm for assigning router IDs).
- **Content deployment using hash value of content name**: Content having the name A is simply cached only at routers with IDs that agree with the hash value of A, F(A), in some of the upper digits (see remaining part of this section for details). As a result of this simple autonomous judgement on selecting cached content at routers, SDC realizes the spatially dispersed deployment of each content item.
- **Control of copy count based on content popularity**: Each router autonomously classifies content items into $K+2$ groups on the basis of the popularity and checks the consistency between the router ID and F(A) in smaller bits for highly popular content when caching content (see Section V-B for details on the algorithm for grouping content items). As a result, content items with higher popularity are cached at more routers, and we can expect to improve the cache hit ratio as well as the hop length of delivery flows.

Without grouping content on the basis of the popularity and without differentiating the number of bits considered among popularity groups, content is always cached at only a single router in the network similar with the method proposed by Saha et al. [24]. Next, we describe the details of the caching mechanism at routers in the SDC. Table I summarizes the definitions of symbols used in this paper.

TABLE I
Summary of variables

Variable	Semantics
C_n	storage capacity of content store at router n
$d_{i,j}$	hop distance from router i to router j
D	average hop length of delivery flows
D_o	average hop distance to origin servers
D_c	average hop length when delivering content from caches
D_n	average hop length when delivering content to router n
F(A)	hash value of content name A
$\boldsymbol{G}_n(k)$	set of content items classified into group k at router n
H	cache hit ratio
$h_n(m)$	cache hit ratio of content m at router n on which content m is cache target
K	length of binary ID assigned to routers
M	number of content items
$\boldsymbol{M}_n$	set of content items requested one or more times at router n
$m_n(k)$	number of content items classified into $\boldsymbol{G}_n(k)$ at router n
N	number of routers
O_m	origin server of content m
o_n	probability that O_m exists at router n
p_n	ratio of requests generated from router n
Q_n	total ratio of requests for content items that are cache targets at router n
$q_n(m)$	ratio of requests for content m measured at router n
$\hat{q}_n(m)$	effective ratio of requests for content m at router n
$R(x)$	sustainable ratio at level-x failures
$\boldsymbol{S}_k(\boldsymbol{X}_k)$	set of routers assigned ID $\boldsymbol{X} = (x_1, \cdots, x_k)$ at top k bits
$\omega_n(s)$	probability that origin server exists at s hop distance from router n
$y_n(m)$	number of Interest of content m generated at router n
$z_n(m)$	popularity group of content m classified at router n

B. Caching Mechanism

Let us consider when arriving content with the name A that is classified into popularity group k arrives at router n. Router n caches this content only when the most upper k bits of the ID of router N matches those of F(A), the hash value of A[4], if k is in the range $1 \leq k \leq K$. In the case of $k = 0$, router n always caches this content, whereas router n never caches this content if $k = K+1$. In any cases, the router accommodating the origin server of A never caches this content.

As an example of caching decisions, Figure 1 illustrates the case in which content A with a hash value of F(A)=101 is delivered from its origin server to the user terminal traversing through routers e, d, c, b, and a. If the popularity group of this content is $k = 1$, only routers a, c, and d whose highest bit of router ID, that is, “1”, agrees with that of F(A), “1”,

[3]In practice, the ICN function is likely to be implemented in routers step by step, so we could possibly face a situation in which only a part of routers have the function. Although this incremental deployment is an open issue [29], we might be able to cope with it by implementing the ICN function as a network function virtualization (NFV) and operating the ICN as a virtual network [8], for example.

[4]In the case of the hash table, we need to cope with hash collision, i.e., different targets generate the identical hash value, by a linked-list, for example [11]. However, in this case, we do not have to cope with a hash collision because there is no problem even if different content items are cache at the same router.

cache this content. If the popularity group of this content is $k = 2$, only routers a and c, whose first and second highest bits of router ID, "10", agree with those of F(A), "10", cache this content. As shown in this example, different sets of routers on the default path can be candidates according to the popularity group of content even when delivering content with the same hash value. Therefore, if IDs are uniformly assigned to routers without deviation, $\lfloor N/2^k \rfloor$ or $\lceil N/2^k \rceil$ routers among N routers are candidates for caching content belonging to popularity group k, so popular content that is classified into the popularity group with a smaller k is cached at more routers. Moreover, as mentioned in Section IV, by assigning IDs with identical bits in the highest k digits to routers located at geographically separated positions, we disperse the location for caching each content item spatially and avoid duplicate caching of identical content in nearby areas.

It has been reported that LRU achieved a performance close to the optimum in ICN [7][26], so we assume that LRU is used as the cache-replacement policy. In other words, if the available storage capacity of cache is insufficient when inserting content to caches at routers, the content with the longest elapsed time after the final request is removed to make free space in the cache.

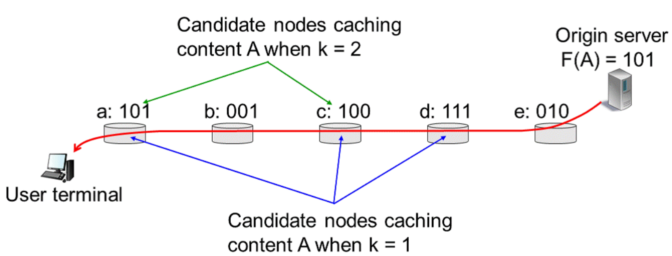


Fig. 1. Example of candidate routers caching content A when popularity group k is 1 or 2

IV. Assignment of Router IDs

In SDC, routers judge whether to cache content on the basis of the router ID, so the method for assigning router IDs strongly affects the effect of spatially dispersing the cached locations of content. In this section, we describe the detail of the algorithm for assigning IDs to routers.

A. Policy of Assigning Router IDs

As mentioned in Section III-B, popular content items are cached at many routers by limiting the number of the digits of IDs checked to fewer highest digits, and the influence of their deployment pattern on the overall performance is strong, so we sequentially assign identical bits IDs to routers located remotely from the highest to lowest digit. Let $\boldsymbol{S}_k(\boldsymbol{X}_k)$ denote the set of routers assigned $\boldsymbol{X}_k = (x_1, x_2, \cdots, x_k)$ as the highest k bits. We can divide $\boldsymbol{S}_k(\boldsymbol{X}_k)$ into the two subsets of routers, $\boldsymbol{S}_{k+1}(\boldsymbol{X}_k, 0)$ and $\boldsymbol{S}_{k+1}(\boldsymbol{X}_k, 1)$, by assigning 0 or 1 to the $(k+1)$-th bit of each router n of $n \in \boldsymbol{S}_k(\boldsymbol{X}_k)$. We repeat this procedure assigning $(k+1)$-th bit for each router n of $\boldsymbol{S}_k(\boldsymbol{X}_k)$ in the order of $k = 0,\ 1,\ \cdots,\ K-1$. We note that $\boldsymbol{N}$, the set of all N routers, is the target of ID assignment when $k = 0$, i.e., $\boldsymbol{S}_0(\phi) = \boldsymbol{N}$.

Figure 2 illustrates the first four steps for assigning IDs to routers when $N = 11$. As shown in Figure 2(a), the first bit is assigned to the IDs to all routers $\boldsymbol{N}$ with the initial state that no bits are assigned to the IDs of all the routers. Each bit of the router IDs takes a value of zero or unity, so $\boldsymbol{N}$ is divided into $\boldsymbol{S}_1(0)$, the set of routers assigned 0 to the first bit of an ID, and $\boldsymbol{S}_1(1)$, the set of routers assigned 1 to the first bit of an ID, with the constraint that the difference of the sizes of $\boldsymbol{S}_1(0)$ and $\boldsymbol{S}_1(1)$ is less than or equal to unity. Next, as shown in Figure 2(b), by assigning the second bit to the ID of each router n of $n \in \boldsymbol{S}_1(0)$, we divide $\boldsymbol{S}_1(0)$ into $\boldsymbol{S}_2(0,0)$, the set of routers assigned 00 at the first two bits, and $\boldsymbol{S}_2(0,1)$, the set of routers assigned 01 at the first two bits.

Next, as shown in Figure 2(c), by assigning the second bit to the ID of each router n of $n \in \boldsymbol{S}_1(1)$, we divide $\boldsymbol{S}_1(1)$ into $\boldsymbol{S}_2(1,0)$, the set of routers assigned 10 at the first two bits, and $\boldsymbol{S}_2(1,1)$, the set of routers assigned 11 at the first two bits. Next, we divide each of the four obtained router sets, $\boldsymbol{S}_2(0,0)$, $\boldsymbol{S}_2(0,1)$, $\boldsymbol{S}_2(1,0)$, and $\boldsymbol{S}_2(1,1)$, into two subsets by assigning the third bit to the IDs of routers in each set. Figure 2(d) shows the procedure for dividing $\boldsymbol{S}_2(0,0)$ into $\boldsymbol{S}_3(0,0,0)$ and $\boldsymbol{S}_3(0,0,1)$ as an example. We repeat this procedure until K bits are assigned to the IDs of all the N routers.

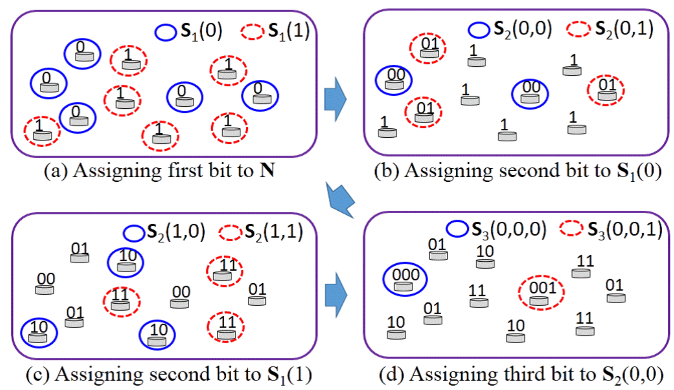


Fig. 2. Example of first four steps of router ID assignment procedure

B. Assignment of $(k+1)$-th bit to IDs of routers of $\boldsymbol{S}_k(\boldsymbol{X}_k)$

Now, we describe how to assign the $(k+1)$-th bit to the IDs of routers of $\boldsymbol{S}_k(\boldsymbol{X}_k)$ at each step of the ID assignment of routers mentioned in Section IV-A. We define T_n as the sum of the minimum hop distance to the router set $\boldsymbol{S}_{k+1}(\boldsymbol{X}_k, 0)$ and that to the router set of $\boldsymbol{S}_{k+1}(\boldsymbol{X}_k, 1)$ from each router n of $\boldsymbol{N}$, and we have

$$T_n = \min_{a \in \boldsymbol{S}_{k+1}(\boldsymbol{X}_k, 0)} d_{n,a} + \min_{b \in \boldsymbol{S}_{k+1}(\boldsymbol{X}_k, 1)} d_{n,b}, \tag{1}$$

where $d_{n,j}$ is the minimum-hop route from router n to router j. Content items of popularity group k are cached at all the routers with the top k bits of ID agreeing with their hash values, so it is desirable to assign the $(k+1)$-th bit to the IDs of each router of $\boldsymbol{S}_k(\boldsymbol{X}_k)$ so that T, the average of T_n, is minimized.

Therefore, we define the following optimization problem dividing $\boldsymbol{S}_k(\boldsymbol{X}_k)$ into the two subsets of routers, $\boldsymbol{S}_{k+1}(\boldsymbol{X}_k, 0)$

and $\boldsymbol{S}_k(\boldsymbol{X}_k, 1)$:

$$\min T = \sum_{n \in \boldsymbol{N}} p_n T_n, \tag{2}$$

$$s.t. \quad -1 \leq \mid \boldsymbol{S}_{k+1}(\boldsymbol{X}_k, 0) \mid - \mid \boldsymbol{S}_{k+1}(\boldsymbol{X}_k, 1) \mid \leq 1, \tag{3}$$

$$\boldsymbol{S}_{k+1}(\boldsymbol{X}_k, 0) \cap \boldsymbol{S}_{k+1}(\boldsymbol{X}_k, 1) = \phi, \tag{4}$$

$$\boldsymbol{S}_{k+1}(\boldsymbol{X}_k, 0) \cup \boldsymbol{S}_{k+1}(\boldsymbol{X}_k, 1) = \boldsymbol{S}_k(\boldsymbol{X}_k), \tag{5}$$

where p_n is the ratio of requests generated from users accommodated at router n. The number of combinations dividing $\mid \boldsymbol{S}_k(\boldsymbol{X}_k) \mid$ routers into two groups exponentially increases as $\mid \boldsymbol{S}_k(\boldsymbol{X}_k) \mid$ grows, so solving this problem strictly is difficult. Therefore, we solve this problem by using the following greedy-based algorithm.

Algorithm 1 Greedy algorithm dividing router set $\boldsymbol{S}_k(\boldsymbol{X}_k)$ into two subsets $\boldsymbol{S}_{k+1}(\boldsymbol{X}_k, 0)$ and $\boldsymbol{S}_{k+1}(\boldsymbol{X}_k, 1)$

1: Initializes $\boldsymbol{S}_{k+1}(\boldsymbol{X}_k, 0)$ and $\boldsymbol{S}_{k+1}(\boldsymbol{X}_k, 1)$ as $\boldsymbol{S}_{k+1}(\boldsymbol{X}_k, 0) = \boldsymbol{S}_k(\boldsymbol{X}_k)$ and $\boldsymbol{S}_{k+1}(\boldsymbol{X}_k, 1) = \phi$
2: Derives T when moving each router a of $a \in \boldsymbol{S}_{k+1}(\boldsymbol{X}_k, 0)$ to $\boldsymbol{S}_{k+1}(\boldsymbol{X}_k, 1)$ by changing the $(k+1)$-th bit of the ID of router a from 0 to 1
3: Moves router a^* whose shift between the two subsets gives the minimum T from $\boldsymbol{S}_{k+1}(\boldsymbol{X}_k, 0)$ to $\boldsymbol{S}_{k+1}(\boldsymbol{X}_k, 1)$
4: Repeats steps 2 and 3 until $\mid \boldsymbol{S}_{k+1}(\boldsymbol{X}_k, 0) \mid = N_a$ is satisfied

We note that N_a is the target size of $\boldsymbol{S}_{k+1}(\boldsymbol{X}_k, 0)$, and we set $N_a = \mid \boldsymbol{S}_k(\boldsymbol{X}_k) \mid /2$ when $\mid \boldsymbol{S}_k(\boldsymbol{X}_k) \mid$ is an even value, and we set $N_a = \lfloor \mid \boldsymbol{S}_k(\boldsymbol{X}_k) \mid /2 \rfloor$ or $N_a = \lceil \mid \boldsymbol{S}_k(\boldsymbol{X}_k) \mid /2 \rceil$ giving a smaller value of T when $\mid \boldsymbol{S}_k(\boldsymbol{X}_k) \mid$ is an odd value.

Figure 3 shows router IDs assigned in the Cable & Wireless network, a commercial backbone ISP network in the USA, whose topology is publicly available at the CAIDA webpage [2]. In this network, $N = 19$ routers exist, and $K = 5$, so the binary IDs of five bits were assigned to each router. We confirmed that the identical value, i.e., zero or unity, at the upper bits is dispersedly assigned to routers because the proposed method assigns the router ID from the first bit to the K-th bit as mentioned in Section IV-A.

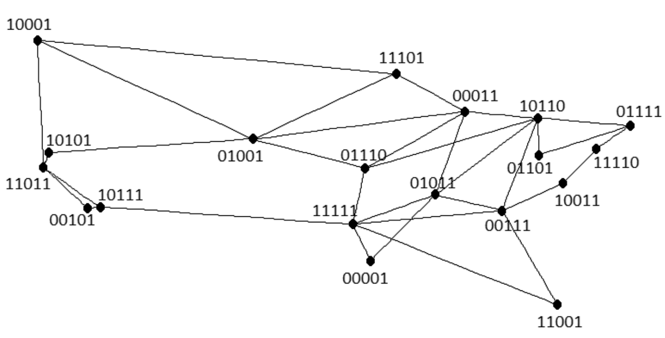


Fig. 3. Router IDs assigned in Cable & Wireless network

C. Discussion on Router ID Assignment

By assigning the IDs to routers using the methods described in Sections IV-A and IV-B, the IDs with identical values in the top k bits are assigned to routers located at dispersed positions for each k of $1 \leq k \leq K$, so we can expect to spatially disperse the caching location of content. However, because of $N \leq 2^K$, there are no routers to which $2^K - N$ IDs are not assigned, and content items having hash value F(A) not assigned to any routers and being grouped into the popularity group K are not cached at any routers. In the case of the Cable & Wireless network shown in Figure 3, 13 IDs including 00000 and 00010 are not assigned. However, the Interest will be transmitted toward the origin servers, so content items with these IDs are still delivered to users from the origin servers.

At the time of failure of any routers, content cannot be cached and delivered at these routers. However, other routers that are normally operated can still cache content on the basis of their assigned IDs without being assigned new IDs. When new routers are added, we can consider two approaches to configuring router IDs: (i) assigning IDs only to routers newly added without modifying IDs for existing routers and (ii) reassigning new IDs for all routers including the existing routers and newly added routers. Although the second approach is desirable to maintain the spatially dispersed assignment of router IDs, IDs will change at existing routers. However, content cached on the basis of old IDs will be removed and replaced by content cached on the basis of new IDs in stages by the cache-replacing policy of LRU.

V. Management of Popularity Groups

To increase the cache hit ratio and reduce the hop length of delivery flows, SDC differentiates the number candidate routers caching content according to the popularity. To realize this function, each router is required to autonomously monitor the popularity of each content item and classify each item to any of the $K + 2$ popularity groups on the basis of the measured popularity[5]. In this section, we describe the details of these functions managing the popularity groups.

A. Measurement of Content Popularity

A popularity group table (PGT) is provided at each router n that manages $y_n(m)$, the counter of measured Interest, and $z_n(m)$, the classified popularity group, for each content item m. Router n increments $y_n(m)$ every time when receiving the Interest for content m from users accommodated in router n[6], and router n calculates $q_n(m) = y_n(m) / \sum_{j \in \boldsymbol{M}_n} y_n(j)$, the ratio of $y_n(m)$ among those of content items of $\boldsymbol{M}_n$, the set of content items from which one or more Interest has been received from local users, in a fixed time interval, e.g., five minutes. In this time interval, router n classifies the content of $\boldsymbol{M}_n$ into $K + 2$ popularity groups by using the

[5]We can also consider the centralized approach in which a controller monitors the content popularity, classifies content items into popularity groups, and informs the popularity group of each content item to routers by adding this information to the header of content chunk. However, the locality of content popularity at each router cannot be reflected to the cache control at routers in this approach.

[6]If routers update the counters when receiving the Interest from other routers, we also consider that the results will be biased depending on the position on the network topology, so we assume that just the Interest is generated locally at each router.

algorithm described in Section V-B while regarding $q_n(m)$ as the estimate of the request ratio of content m and registers the assignments of popularity groups on the PGT.

We assign the group IDs, 0 to $K+1$ in the descending order of popularity, i.e., the most popular content items are grouped into the popularity group 0, and the least popular content items are grouped into the popularity group $K+1$. Moreover, by periodically decrementing $y_n(m)$ for all the content of $\boldsymbol{M}_n$ at all the routers at a fixed time interval, e.g., one minute, SDC copes with the variability of content popularity.

B. Content Grouping

Let $\boldsymbol{G}_n(k)$ denote the set of content items classified into popularity group k at router n, i.e., $\boldsymbol{G}_n(k) = \{m| \ m \in z_n(m) = k\}$. We set the content IDs m in the descending order of $q_n(m)$, and we define $m_n(k)$ as the number of content items classified into popularity group k, i.e., $m_n(k) =| \boldsymbol{G}_n(k) |$. For each k of $0 \leq k \leq K+1$, router n groups content items as

$$\boldsymbol{G}_n(k) = \Big\{m \mid \sum_{i=0}^{k-1} m_n(i) + 1 \leq m \leq \sum_{i=0}^{k} m_n(i)\Big\}. \tag{6}$$

In other words, the most popular $m_n(0)$ content items are classified into $\boldsymbol{G}_n(0)$, the most popular $m_n(1)$ content items except ones grouped into $\boldsymbol{G}_n(0)$ are classified into $\boldsymbol{G}_n(1)$, and so on.

When the Interest for content m with $z_n(m) = k$ arrives at router n[7], the Interest is transmitted toward O_m, the origin server of content m, on the default path from router n, and content m is delivered to router n from router j closest to router n among those caching content m on the default path. Now, let us derive $b_{m,k,s}(d)$, the probability that the hop length from router j to router n is d with the condition that the minimum-hop distance from router n to O_m is s.

As mentioned in Section IV-A, the SDC assigns the IDs to routers so that the identical content is cached at spatially dispersed positions, so we can regard routers that are the candidates for caching the content m of popularity group k as existing in the interval of 2^k on average. Therefore, the probability that the hop distance to router v_0 closest to router n on the default path among routers that are the candidates of caching content m is d is $1/2^k$ when $0 \leq d \leq 2^k - 1$. Moreover, when d is in the range of $r2^k \leq d \leq \min\{(r+1)2^k - 1, \ s-1\}$ for each integer r of $1 \leq r \leq \lfloor (s-1)2^{-k} \rfloor$, we can regard routers $v_1, \ v_2, \ \cdots \ v_r$ located at the interval of 2^k hops on the default path from v_0 to router n as the candidates for caching content m. Figure 4 shows an example when $k = 1$, $r = 3$, and $s = 8$. When only router v_0 among these $r+1$ routers that are the candidates of caching content m caches content m, d is in the range of $r2^k \leq d \leq \min\{(r+1)2^k - 1, \ s-1\}$. Therefore, when d is in this range, $b_{m,k,s}(d)$ is obtained by

$$b_{m,k,s}(d) = \prod_{j=1}^{r} \big\{1 - h_{v_j}(m)\big\} h_{v_0}(m) 2^{-k}, \tag{7}$$

[7]A user accommodated at router n requests content m, or the Interest of content m arrives from an adjacent router.

and it is given by $b_{m,k,s}(s) = 1 - \sum_{j=0}^{s-1} b_{m,k,s}(j)$ when $d = s$. Here, $h_v(m)$ is the probability that router v, which is the candidate for caching content m, actually caches content m. Che′s approximation is widely used as an approximation of the hit ratio of cache under the LRU replacement policy [9]. The approximation was originally proposed by Che et al. [4], and h_m, the cache hit ratio of content m, is approximated by

$$h_m = 1 - e^{-q_m t_C}, \tag{8}$$

where q_m is the ratio of requests for content m, C is the storage capacity of cache, and t_C is the unique root of the equation $\sum_{m=1}^{M}(1 - e^{-q_m t}) = C$.

Among $m_v(k)$ content items classified into popularity group k at any router v, $m_v(k)/2^k$ content items are the caching target at router v on average. Therefore, Q_v, the total ratio of requests for content items that are the caching target at router v, is given by

$$Q_v = \sum_{k=0}^{K} \sum_{m \in \boldsymbol{G}_v(k)} q_v(m) 2^{-k}. \tag{9}$$

Hence, $\hat{q}_v(m)$, the actual request ratio for content m that is the caching target at router v, is given by

$$\hat{q}_v(m) = \frac{q_v(m)}{Q_v}. \tag{10}$$

Moreover, the number of content items that are the caching target at router v is $\sum_{k=0}^{K} m_v(k) 2^{-k}$ on average, so from a given C_v, the storage capacity of cache memory at router v, $h_v(m)$ is given by

$$h_v(m) = 1 - e^{-\hat{q}_v(m) t_{c,v}}, \tag{11}$$

where $t_{c,v}$ is the unique root of the equation,

$$\sum_{k=0}^{K} \sum_{m \in \boldsymbol{G}_v(k)} \Big\{1 - e^{-\hat{q}_v(m)t}\Big\} 2^{-k} = \min\big\{C_v, \ \sum_{k=0}^{K} m_v(k) 2^{-k}\big\}. \tag{12}$$

However, $m_v(k)$ and $q_v(m)$ at other routers v are unknown for router n, so we apply $h_n(m)$ derived by (11) and (12) using $m_n(k)$ and $q_n(m)$ to $h_v(m)$ at all the other routers v.

Next, let us derive D_n, the average hop length when content is delivered to router n. We assume that the probability that O_m exists at router n is given by o_n independently of m and that the set of routers to which the hop distance from router n on the default paths is s is given by $\Omega_{n,s}$. Now, $\omega_n(s)$, the probability that O_m exists at the position with s hop distance from router n, is $\omega_n(s) = \sum_{j \in \boldsymbol{\Omega}_{n,s}} o_j$. Therefore, we obtain D_n by

$$\begin{aligned} D_n \ &= \ \sum_{k=0}^{K} \sum_{m \in \boldsymbol{G}_n(k)} q_n(m) \sum_{s=0}^{S_n} \omega_n(s) \sum_{d=0}^{s} b_{m,k,s}(d) d \\ &+ \sum_{m \in \boldsymbol{G}_n(K+1)} q_n(m) \sum_{j=0}^{S_n} j \omega_n(j), \end{aligned} \tag{13}$$

where S_n is the minimum hop distance to the most distant router from router n. We define the following optimization problem deriving $m_n(k)$ that minimizes D_n:

$$\min \quad D_n, \tag{14}$$

$$s.t. \quad \sum_{k=0}^{K+1} m_n(k) = \mid \boldsymbol{M}_n \mid . \tag{15}$$

Because the number of possible combinations of $m_n(k)$ exponentially increases with the increase of $\mid \boldsymbol{M}_n \mid$, we derive $m_n(k)$ by using the following greedy-based algorithm.

Algorithm 2 Greedy algorithm deriving size of popularity groups at route n

1: Classifies all content items of $\boldsymbol{M}_n$ into $\boldsymbol{G}_n(K+1)$ at the initial state, i.e., $t = 0$
2: Derives D_n when incrementing $m_n(a)$ and decrementing $m_n(b)$ for each integer pair of a and b that satisfies $0 \leq a < K+1$, $a < b \leq K+1$, and $m_b > 0$
3: For a pair of a^* and b^* giving the minimum D_n among all the possible combinations of a and b, increments $m_n(a^*)$ and decrements $m_n(b^*)$
4: Repeats steps 2 and 3 while D_n decreases

Figure 5 shows an example when $b = a + 2$. The most popular content in $\boldsymbol{G}_n(k)$ is shifted to $\boldsymbol{G}_n(k-1)$ for each k in the range of $a < k \leq b$ at step 2.

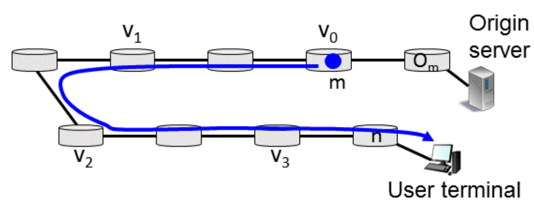


Fig. 4. Example of caching pattern of content when $k = 1$, $r = 3$, and $s = 8$

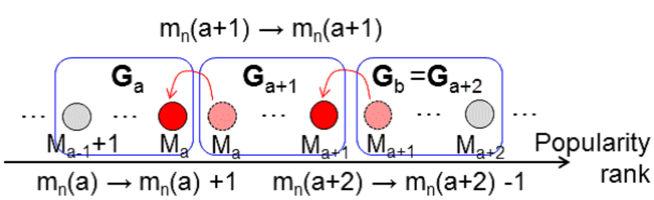


Fig. 5. Example of updating group assignment when $b = a + 2$

VI. Numerical Evaluation

A. Evaluation Conditions

1) Network Topologies: We used the backbone networks of four commercial ISPs in the USA, At Home Network, CAIS Internet, Allegiance Telecom, and Verio, whose PoP-level topologies are publicly available at the CAIDA website [2]. Figure 6 shows the topologies of these four networks, where nodes are PoPs, and we assume that ICN-routers are provided at all the N PoPs. Let r_n denote the population ratio of node n, i.e., the population of node n divided by the total population of all the N nodes. Table II summarizes N, the node count, E, the link count, and D_o, the average hop distance between nodes weighted by their population ratios, i.e., $D_o = \sum_{i,j \in \boldsymbol{N},\, i \neq j} r_i r_j d_{i,j}$. We assume that both o_n, the probability that the origin server of content m exists at node n, and p_n, the ratio of requests generated from node n, agree with r_n. We also assume that the default path of Interests is the shortest-hop route from a node accommodating a requesting user to the origin server. Therefore, D_o corresponds to the average hop length of delivery flows when content is delivered from the origin servers without using caches.

Although all four networks exist in the USA, we can classify these networks into two types with different shapes. We can classify Allegiance Telecom and Verio into a hub and spoke (H&S) type, in which several hub nodes connected with many other nodes exist. In H&S networks, packets can reach destination nodes with a small hop count by traversing through hub nodes, so D_o is small. We can classify At Home Network and CAIS Internet into a ladder type, in which no hub nodes exist, and packets need to visit many intermediate nodes before arriving at destination nodes, so D_o is large. The number of nodes N of all the four networks is in the range between $2^5 = 32$ and $2^6 = 64$, so K, the number of bits of router ID, is six in all the four networks.

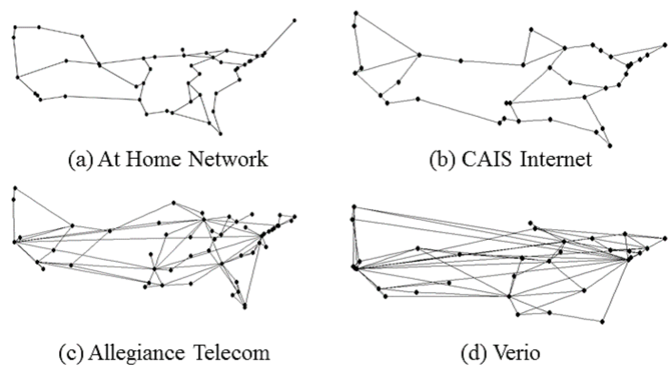


Fig. 6. Topologies of evaluated networks

TABLE II
SUMMARY OF PROPERTIES OF FOUR NETWORKS EVALUATED

Network	N	E	D_o	Type
At Home Network	46	55	6.83	Ladder
CAIS Internet	37	44	5.49	Ladder
Allegiance Telecom	53	88	2.81	Hub & Spokes
Verio	35	72	2.23	Hub & Spokes

2) Content Demand: We set M, the total content count, to 10,000. It has been reported that the request distribution of various types of digital content, e.g., websites and user-generated videos, obey the Zipf distribution [1][3]. For example, the request count of websites obeyed the Zipf distribution with a parameter θ between 0.64 and 0.83 [1] or between 0.74 and 0.84 [19]. The request count of YouTube videos obeyed the Zipf distribution with a parameter θ of about 0.8 [3]. Therefore, we assume that $q_n(m)$, the request ratio of content m measured at router n, obeys the Zipf distribution with a parameter θ in the range between 0.6 and 0.9. Without otherwise stated, we set $\theta = 0.8$ as the default setting. Although each router estimates $q_n(m)$ on the basis of the measured Interest count as mentioned in Section V-A, we use the setting value of $q_n(m)$ at all N routers. We assume that parameter θ as well as the popularity rank of M content items are identical at all N routers. In Section VI-G, we evaluate the case in which the popularity rank of content items is different among routers.

3) Cache and Origin Servers: We assume that the size of all M content items is identical, and C_n, the storage capacity of the content store at router n, is C at all N routers. We set C in the range between 10 and 100 content items, i.e., 0.1 and 1.0 percent of the content-catalogue size. Without otherwise stated, we set $C = 50$ as the default setting. We generated one million requests sequentially from router n randomly selected according to r_n for content m randomly selected according to $q_n(m)$. At the initial state, the cache memory of all N routers was empty, and we started to measure all the statistics after generating 100,000 requests, i.e., a warmup period 10% of the simulation length. At the beginning of each simulation, we placed the origin server of each content item at a router randomly selected with the probability proportional to the population ratio r_n, and we did not change the location of origin servers during the simulation. We repeated ten trials with different random seeds, and we evaluated all the results by the average value over the ten trials with different origin server allocations.

4) Comparison Methods: For each request generated at router u for content m, the Interest was transferred toward router o accommodating O_m on the default path, and content m was delivered from router s closest to router u among routers caching content m on the default path. If content m was not cached at all the routers on the default path, router o was the source router s. To clarify the effectiveness of the proposed SDC, we compared SDC with the following five caching strategies.

AllCache Content was simply cached at all routers on the default path from the source router s to the destination router u [5]. This method is also known as transparent en-route caching (TERC) [17] or universal caching [14].

EdgeCache Content was cached only at the last hop router on the default path, i.e., router u.

UniCache Content was cached at each router on the default path with the probability of $1/d_{s,u}$ [5], so content was cached at only one router randomly selected on the default path between routers s and u on average.

ProbCache Content was cached at each router c on the default path with the probability of $d_{s,c}/d_{s,u}$ [22]. In other words, content was cached at each router on the default path with the probability proportional to the distance from router s, and a router closer to router u was more likely to cache content.

LCD (leave copy down) Content was cached only at the next hop router from router s [18], and WAVE also took a similar approach with the unit of the chunk [5]. Copies of content tended to exist around the origin servers, and they gradually spread over the network.

In all six caching strategies including the proposed SDC, we used LRU as the cache-replacement policy, and content m was never cached at router o accommodating O_m.

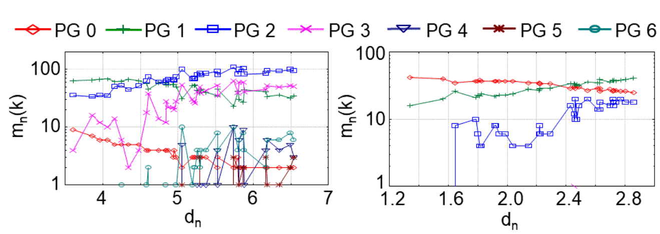
B. Population-Group Size Designed by SDC

As mentioned in Section V-B, each router independently classifies M content items into $K + 2$ popularity groups so that the expected average hop length of delivery flows is minimized. Table III summarizes the average number of content items classified into each popularity group (PG) in each of the four networks when $\theta = 0.8$ and $C = 50$. The distribution of hop length to other routers depends on the position of routers in the network, so the result of content grouping is different among routers. Hence, we show the value averaged over all the N routers in the table. In all four networks, a large part of the content items were grouped into the least popular group, PG7. When deriving D_n, the expected average hop length delivering content to router n, by (13), we assumed that the content m of PG k is cached in the interval of 2^k routers on the default path, so it is desirable to classify more content items into more popular PGs, i.e., smaller k, at routers with a smaller hop length than other routers. In the two H&S networks, i.e., Allegiance Telecom and Verio, the hop distance between routers was much smaller than those in the two ladder networks, i.e., At Home Network and CAIS Internet. So, the average size of the most popular group, i.e., PG0, in the two H&S networks was much larger than those in the two ladder networks Moreover, in the two H&S networks, no content items or just a few content items were classified into the less popular groups, PG3, PG4, PG5, and PG6.

Figure 7 plots $m_n(k)$, the number of content items grouped into PG k at router n, against d_n, the average hop distance to other routers weighted by the request ratio, i.e., $d_n = \sum_{i \in \boldsymbol{N},\, i \neq n} r_i d_{n,i}$, for each PG for CAIS Internet and Verio. In the figure, we show the results for PGs with an average size greater than unity, excluding PG7. We confirmed that more content items were grouped into PG0 in both the networks and PG1 for CAIS Internet at routers with a smaller d_n.

TABLE III
AVERAGE NUMBER OF CONTENT ITEMS GROUPED INTO EACH POPULARITY GROUP

Network	Popularity Group							
	0	1	2	3	4	5	6	7
At Home Network	2.1	31.7	99.2	52.6	7.6	3.5	9.9	9793.5
CAIS Internet	3.5	48.1	73.1	31.9	1.7	0.6	2.9	9838.2
Allegiance Telecom	22.3	42.3	24.7	2.8	0.0	0.0	0.2	9907.7
Verio	32.2	30.1	11.1	0.0	0.0	0.0	0.0	9926.6



(a) CAIS Internet (b) Verio

Fig. 7. Size of popularity groups designed at each router

C. Number of Content Copies Stored at Routers

Now, we compare the tendencies of content deployment among the six caching strategies. Figure 8 plots the number of copies stored at N routers in the descending order of content popularity for each of the six methods at the instance that one million requests were generated for CAIS Internet when $\theta = 0.8$ and $C = 50$. Because the least-recently requested content was removed when the cache memory was full for all six methods, the most popular content items, e.g., the top-10 items, stably remained at many routers for all six methods.

However, the tendencies of the copy counts of lower-ranked content items were clearly different among the six caching strategies. For AllCache and ProbCache, identical content was cached at many routers on the default path, so the copy count of content items was highly variable. Some content items were cached at many routers, i.e., about ten, whereas other content items with a similar popularity rank were cached at no or just a few routers. For EdgeCache, UniCache, and LCD, content was cached at just a single router on the default path[8], so the variability of copy counts was suppressed compared with AllCache and ProbCache. However, the copy count of content items was still variable. For the proposed SDC, the caching location on the default path was explicitly controlled, so the copy count of content items was highly stable, i.e., the copy count gradually changed according to the popularity rank. We also obtained the same tendency with the other three networks as well. We confirmed that SDC can stably control the copy count while reflecting the content popularity.

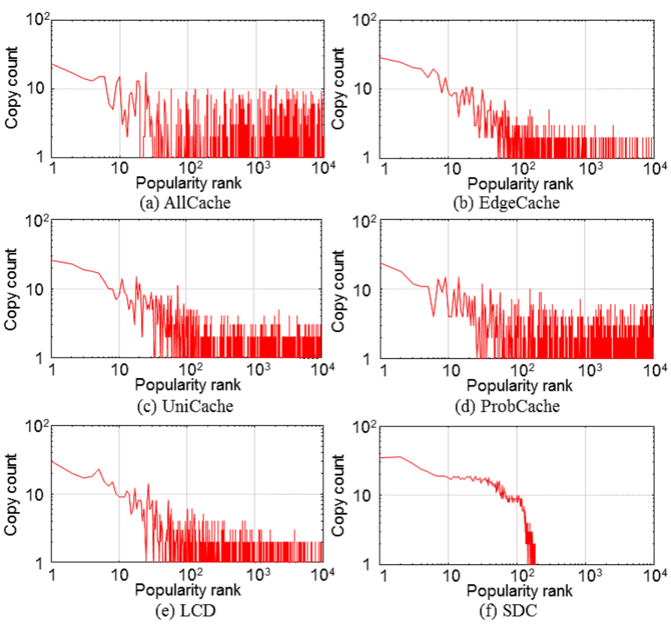


Fig. 8. Number of copies stored at routers at time of simulation completion in CAIS Internet

D. Cache Hit Ratio

We define H, the cache hit ratio, as the probability that content m is cached at any router on the default path between the routers accommodating the requesting user and the origin server O_m, and it is the probability that content is delivered from any router instead of the origin server. Figure 9 plots H against the storage size of caches at routers C in each of the six caching strategies when $\theta = 0.8$ for CAIS Internet and Verio. Figure 10 also plots H against the Zipf parameter θ giving the request distribution when $C = 50$ for each of these two networks. As C increased, more content items could remain at routers, so H increased. Moreover, as θ increased, more requests concentrated on a small number of popular content items, so H increased.

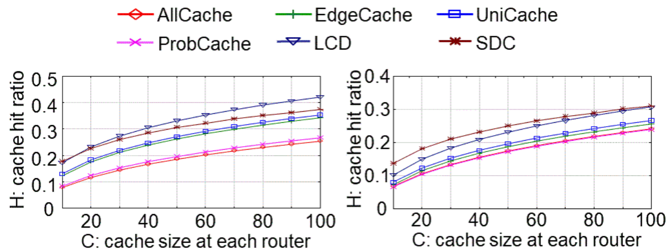
For AllCache and ProbCache, content items were cached at multiple routers on the default path without considering the content popularity, so we could not avoid the wasting of cache resources caused by duplicated caching of identical unpopular content at many routers, and the H of these two methods were lower than that of the other methods. The degradation of H was more remarkable for CAIS Internet with a ladder shape, in which the hop distance between routers was large. For LCD, content items were cached at routers close to the origin servers. Because the Interest for content m was transmitted toward the identical origin server O_m, many default paths of content m from various routers went through the same routers around the router accommodating O_m. Therefore, the Interest will find cached content with a high probability around the origin servers, so the LCD achieved the highest and the second-highest H among the six caching strategies for CAIS Internet and Verio, respectively. However, thanks to the effect of differentiating the cached location of content items, the proposed SDC also achieved a high H, which was close to the result of LCD. For At Home Network and Allegiance Telecom, we also obtained the same tendencies with CAIS Internet and Verio, respectively. We also found the same tendencies on all the following results in the networks with the same type, i.e., the ladder type or H&S type, so we omit the graphs showing the results for At Home Network and Allegiance Telecom due to the space limitation.

Figure 11 plots the H of each popularity group (PG) for when $\theta = 0.8$ and $C = 50$. We calculated the cache hit ratio for each content group derived on the basis of the average size for SDC for all the six caching strategies, and the results are shown for only the PGs with an average size greater than unity. Because LRU was used as the cache-replacement policy in all the methods, H was larger in the groups with a higher popularity for all the six caching strategies. We confirmed that the SDC dramatically improved the H in groups with a high popularity, i.e., PG0, PG1, and PG2, although it degraded the H in groups with a low popularity, compared with the other caching strategies, as a result of explicitly isolating the cached location of content.

E. Average Hop Length at Cache Hit

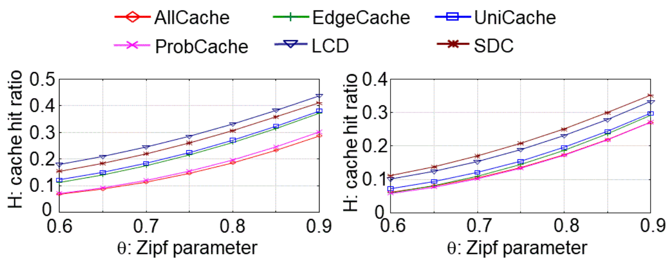
The effect of delivering content from caches instead of origin servers is a reducing the hop length of delivery flows as well as reducing the load of origin servers. Therefore, to investigate the effect of ICN caches, we evaluated D_c, the average hop length of delivery flows, when content was delivered from routers instead of origin servers, i.e., cache hit. Figures 12 and 13 plot D_c against C and θ. For LCD, content items were cached at routers close to the origin servers and

[8]For UniCache, content items were cached probabilistically, and they were cached at one router on average.



(a) CAIS Internet (b) Verio

Fig. 9. Cache hit ratio against cache size at each router

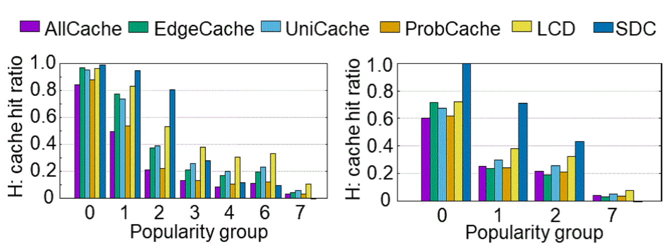


(a) CAIS Internet (b) Verio

Fig. 10. Cache hit ratio against Zipf parameter

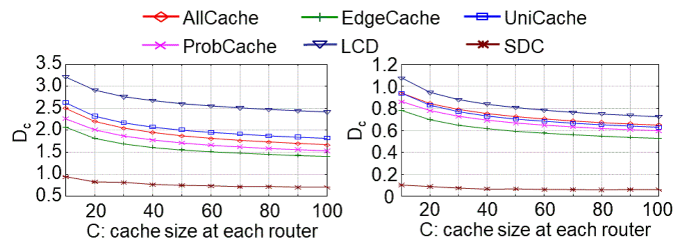
distant from requesting routers, so D_c was largest among the six caching strategies. This tendency was more remarkable in ladder networks, i.e., CAIS Internet, because the hop distance between nodes in ladder networks was larger than that in H&S networks. Among the five existing methods, D_c was the smallest for EdgeCache, in which content was always cached at requesting routers. We confirmed that the proposed SDC achieved a much smaller D_c compared with all the existing methods including EdgeCache. SDC reduced D_c by about 50% to 70% in ladder networks and by about 90% in H&S networks, compared with the existing caching strategies. In all the existing methods, content items were cached without considering content popularity, so even unpopular content items were stored in cache memory once and continued to waste the cache resources until they were removed by LRU. Moreover, all the existing methods did not explicitly isolate the positions for caching the identical content item. In comparison, SDC explicitly isolates the positions for caching each content item based on the content popularity, so copies of content can be found at routers close to users with high probability for many requests.

To illustrate the effect of explicitly isolating the cached location of content, Figure 14 plots σ_c, the standard deviation of hop length at cache hit, for each PG excluding PG7 and PGs with an average size less than unity. As a result of explicitly dispersing the cached locations, SDC dispersed the cached locations of identical content over networks, so the standard deviation of hop length when delivering content from cache memory at routers was reduced for SDC compared with all the other methods. The effect of suppressing the variance of flow hop length was more remarkable for more popular PGs, e.g., PG0.



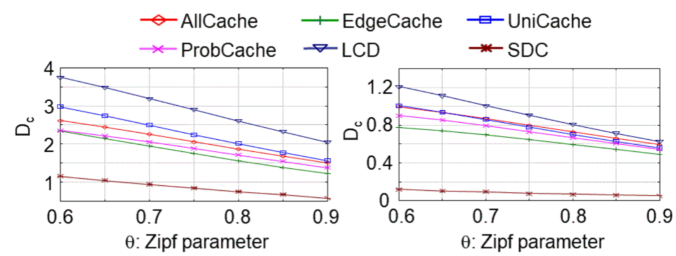
(a) CAIS Internet (b) Verio

Fig. 11. Cache hit ratio of each popularity group



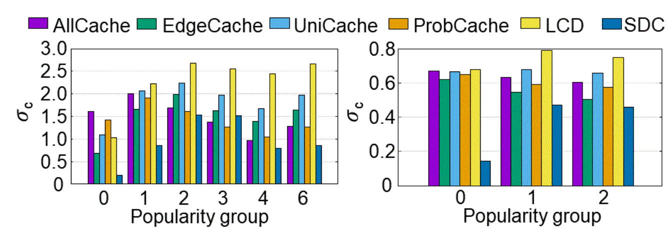
(a) CAIS Internet (b) Verio

Fig. 12. Average hop length at cache hit against cache size at each router



(a) CAIS Internet (b) Verio

Fig. 13. Average hop length at cache hit against Zipf parameter



(a) CAIS Internet (b) Verio

Fig. 14. Standard deviation of hop length at cache hit of each popularity group

F. Average Hop Length

Next, we evaluated D, the average hop length of delivery flows for all requests. When content m was cached at no routers on the default path, i.e., cache miss, content m was delivered from its origin server O_m. Therefore, D is given by $D = D_c H + D_o(1 - H)$. Figures 15 and 16 plot D against C and θ for each of the six caching strategies for CAIS Internet and Verio. Because D_o was identical among all the caching strategies, D was determined by both H, the cache hit ratio, and D_c, the average hop length at cache hit, and D decreased as H increased or D_c decreased. As observed in Figures 9, 10, 12, and 13, SDC achieved the highest H and the smallest D_c among the six caching strategies for Verio. Therefore, the D of SDC was much smaller than that of the other five caching strategies. Although LCD achieved a higher H than SDC for CAIS Internet, the difference was not so remarkable. In addition, the D_c of SDC was much smaller than that of LCD for CAIS Internet, so SDC also achieved the smallest D among the six methods for CAIS Internet as well. We confirmed that SDC can reduce D about 5% to 20% compared with existing caching strategies.

Figure 17 plots the D of each PG with an average size greater than unity. We confirmed that SDC dramatically decreased D for popular content items grouped in PG0, PG1, and PG2, while slightly increasing D for the least popular content items grouped in PG7.

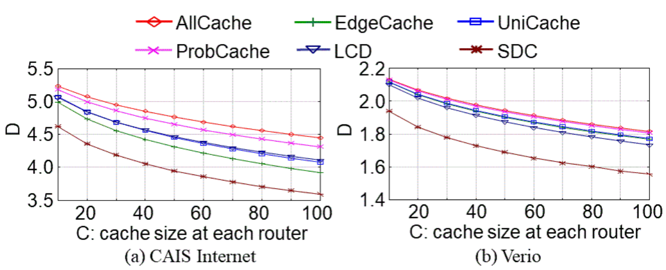


Fig. 15. Average hop length against cache size at each router

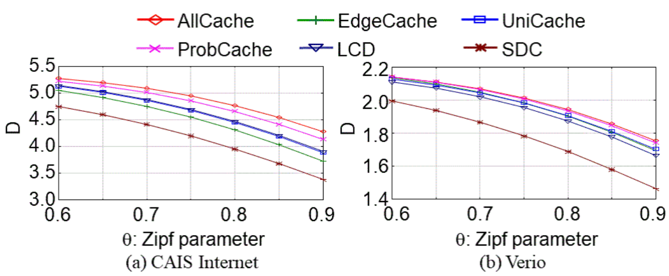


Fig. 16. Average hop length against Zipf parameter

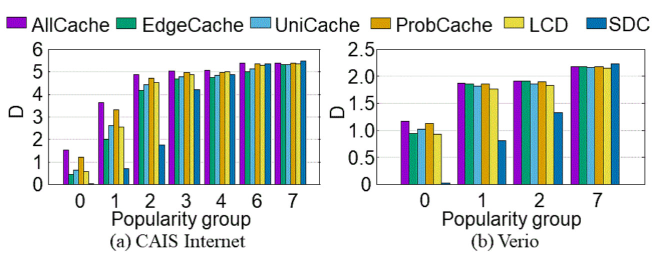


Fig. 17. Average hop length of each popularity group

G. Impact of Spatially Heterogeneous Content Popularity

In the former evaluations, we assumed homogeneous content popularity and used the identical popularity ranks and request distribution at all the N routers. In this section, we relaxed this constraint by assuming heterogeneous ranks of content popularity at routers. At the beginning of each trial of the ten computer simulations with different random seeds, we exchanged the popularity ranks of the two randomly selected content items, and we repeated this procedure ρM times, where ρ is a given parameter taking a real number less than unity. Figure 18 plots D, the average hop length, against ρ for each of the six caching strategies. The case of $\rho = 0$ corresponded to the homogeneous content popularity, and D increased as ρ increased. Although the increase of D when increasing ρ was most remarkable for SDC, we confirmed that SDC was still superior to the other caching strategies in the wide-range of ρ.

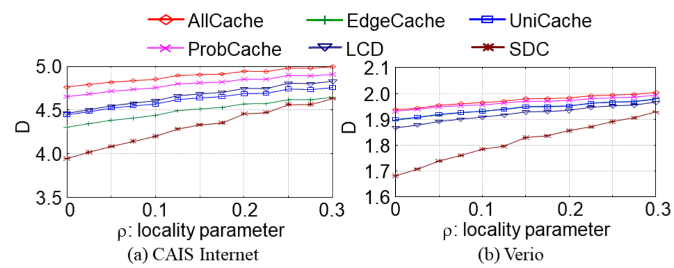


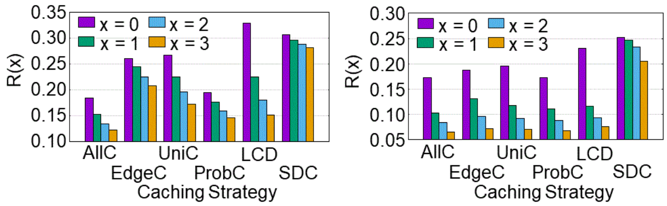
Fig. 18. Average hop length against locality parameter of content-popularity rank

H. Sustainability for Large-Scale Failure of Routers

In ICN, the Interest packet cannot reach the destination routers accommodating the origin servers when the connectivity to the destination routers is lost due to failures of some routers or links in the network. However, content can be delivered from any routers caching the requested content on the default path, so requesting users can still acquire the content items which are cached at any routers on the default path within the range that connectivity is maintained even when the connectivity to the origin servers is lost.

Because identical content items are stored at spatially dispersed locations in SDC, we can expect to improve the sustainability of content acquisition in a large-scale failure of routers, i.e., an outage of multiple routers in nearby areas. To confirm the superiority of the SDC on the sustainability at large-scale failures, we plot $R(x)$, the sustainable ratio in level-x failures, for each x less than or equal to three for each of the six caching strategies in Figure 19, where AllC, EdgeC, UniC, and ProbC stands for AllCache, EdgeCache, UniCache, and ProbCache, respectively. We define $R(x)$ as the probability that content m can be obtained even when all the x routers on the default path closest to the router d accommodating O_m as well as router d are simultaneously in failure. This means that $R(x)$ is the probability that content m is cached at any router on the default path with a hop distance toward router r greater than x. We note that $R(0)$ agreed with H, the cache hit ratio, because $R(0)$ is the probability that content m is cached at any router on the default path. As x increased, i.e., growing the scale of failure, $R(x)$ decreased for all the six caching strategies. We confirmed that the decrease of $R(x)$ with an increasing x in the SDC was the smallest among all the methods, and the SDC can dramatically improve $R(x)$ for x greater than or equal to unity compared with the existing methods. For example, the SDC improved $R(x)$ by about 25% to 125% in ladder networks and about 120%

to 200% in H&S networks compared with existing caching strategies when $1 \leq x \leq 3$.



(a) CAIS Internet (b) Verio

Fig. 19. Sustainable ratio in level-x failures

VII. Conclusion

In this paper, we proposed spatially dispersed caching (SDC), a caching strategy for ICN. In SDC, K bit binary IDs are assigned to routers, and each router caches only content items whose hash values agree with the ID assigned to the router. As a result, SDC spatially disperses the cached location of each content item in a network without exchanging the cached-content information between adjacent routers. SDC classifies content into $K + 2$ groups on the basis of the popularity, and SDC differentiates the number of routers that are the candidates for caching content according to the popularity by changing the number of bits checked at the caching decision. As a result, SDC increases the cache hit ratio and reduces the hop length of delivery flows by highly utilizing the limited cache resources at routers. We also proposed greedy-based algorithms for assigning IDs to routers and classifying content into popularity groups, which minimizes the average hop length at cache hit. Through a numerical evaluation using the topologies of backbone networks of actual commercial ISPs in the USA, we showed that SDC reduced the average hop length at cache hit by 50% to 90% and improved the sustainable ratio at large-scale failures of routers by 25% to 200% compared with the existing caching strategies. In future, we will extend the SDC to dynamically adjust the content groups when the content popularity changes.

References

[1] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, Web Caching and Zipf-like Distributions: Evidence and Implications, IEEE INFOCOM 1999.
[2] CAIDA web page, http://www.caida.org/data
[3] 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 ToN, Vol.17, NO.5, pp.1357-1370, Oct. 2009.
[4] H. Che, Y. Tung, and Z. Wang, Hierarchical Web Caching Systems: Modeling, Design and Experimental Results, IEEE J. Selected Areas of Commun., vol.20, no.7, pp.1305-1314, Sep. 2002.
[5] K. Cho, et al., WAVE: Popularity-based and Collaborative In-network Caching for Content-Oriented Networks, IEEE NOMEN 2012.
[6] J. Choi, J. Han, E. Cho, T. Kwon, and Y. Choi, A Survey on Content-Oriented Networking for Efficient Content Delivery, IEEE Commun. Mag., vol.49, no.3, pp.121-127, Mar. 2011.
[7] S. Fayazbakhsh, et al., Less Pain, Most of the Gain: Incrementally Deployable ICN, ACM SIGCOMM 2013.
[8] A. Fischer, J. F. Botero, M. T. Beck, H. Meer, X. Hesselbach, Virtual Network Embedding: A Survey, IEEE Communications Survey and Tutorials, Vol. 15, No. 4, pp.1888-1906, Apr. 2013
[9] C. Fricker, P. Robert, and J. Roberts, A Versatile and Accurate Approximation for LRU Cache Performance, ITC 24.
[10] M. Gritter and D. R. Cheriton, An architecture for content routing support in the Internet, USENIX USITS 2001.
[11] J. Hasan, S. Cadambi, V. Jakkula, and S. Chakradhar, Chisel: A Storage-efficient, Collision-free Hash-based Network Processing Architecture, IEEE ISCA 2006.
[12] S. Ioannidis and E. Yeh, Adaptive Caching Networks with Optimality Guarantees, ACM SIGMETRICS 2016.
[13] S. Ioannidis and E. Yeh, Jointly Optimal Routing and Caching for Arbitrary Network Topologies, ACM ICN 2017.
[14] V. Jacobson, et al., Networking Named Content, ACM CoNEXT 2009.
[15] N. Kamiyama and M. Murata, Spatially-Dispersed Caching in Information-Centric Networking, IEEE ICC 2018 (to be published).
[16] T. Koponen, M. Chawla, B. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, A data-oriented (and beyond) network architecture, ACM SIGCOMM 2007.
[17] J. Kurose, Information-Centric Networking: The Evolution from Circuits to Packets to Content, Elsevier Computer Networks, Vol. 66, 19, pp.112-120, June 2014.
[18] N. Laoutaris, H. Che, and I. Stavrakakis, The LCD interconnection of LRU caches and its analysis, Elsevier Performance Evaluation, Vol. 63, Issue 7, pp. 609-634, July 2006.
[19] A. Mahanti, C. Williamson, and D. Eager, Traffic Analysis of a Web Proxy Caching Hierarchy, IEEE Network 2000.
[20] T. Mick, R. Tourani, and S. Misra, MuNCC: Multi-hop Neighborhood Collaborative Caching in Information Centric Networks, ACM ICN 2016.
[21] D. Perino and M. Varvello, A Reality Check for Content Centric Networking, ACM ICN 2011.
[22] I. Psaras, W. K. Chai, and G. Pavlou, Probabilistic In-Network Caching for Information-Centric Networks, ACM ICN 2012.
[23] M. Rezazad and Y. C. Tay, CONdnS: A strategy for spreading content and decoupling NDN caches, IFIP Networking 2015.
[24] S. Saha, A. Lukyanenko, and A. Yla-Jaaski, Efficient Cache Availability Management in Information-Centric Networks, Elsevier Computer Networks, Vol.84, No.19, pp.32-45, June 2015.
[25] L. Saino, I. Psaras, and G. Pavlou, Hash-routing Schemes for Information Centric Networking, ACM ICN 2013.
[26] A. Sharma, A. Venkataramani, and R. K. Sitaraman, Distributing Content Simplifies ISP Traffic Engineering, ACM SIGMETRICS 2013.
[27] J. M. Wang, J. Zhang, and B. Bensaou, Intra-AS Cooperative Caching for Content-Centric Networks, ACM ICN 2013.
[28] H. Xie, G. Shi, and P. Wang, TECC: Towards Collaborative In-network Caching Guided by Traffic Engineering, IEEE INFOCOM 2012 Mini-conference.
[29] G. Xylomenos, et al., A Survey of Information-Centric Networking Research, IEEE Communications Survey and Tutorials, Vol. 16, No. 2, pp.1024-1049, 2014.
[30] H. Yu, D. Zheng, B. Zhao, and W. Zheng, Understanding User Behavior in Large-Scale Video-on-Demand Systems, ACM EuroSys 2006.
[31] L. Zhang, et al., Named Data Networking (NDN) Project, Technical Report NDN-0001, Oct. 2010.

PLACE PHOTO HERE

Noriaki Kamiyama received his M.E. and Ph.D. degrees in communications engineering from Osaka University in 1994 and 1996, respectively. From 1996 to 1997, he was with the University of Southern California as a visiting researcher. He joined NTT Multimedia Network Laboratories in 1997, and he has been at NTT Network Technology Laboratories by 2016. He was also with the Osaka University as an invited associate professor from 2013 to 2014 and an invited professor in 2015. From 2017, he is a professor of Fukuoka University. He has been engaged in research concerning content distribution systems, network design, network economics, traffic measurement and analysis, traffic engineering, and optical networks. He received the best paper award at the IFIP/IEEE IM 2013. He is a member of IEEE, the Association for Computing Machinery (ACM), and the Institute of Electronics, Information and Communication Engineers (IEICE).

PLACE PHOTO HERE

Masayuki Murata received M.E. and Ph.D. degrees in information science and technology from Osaka University in 1984 and 1988. In April 1984, he joined the Tokyo Research Laboratory IBM Japan as a researcher. From September 1987 to January 1989, he was an assistant professor with the Computation Center, Osaka University. In February 1989, he moved to the Department of Information and Computer Sciences, Faculty of Engineering Science, Osaka University. From 1992 to 1999, he was an associate professor with the Graduate School of Engineering Science, Osaka University, and since April 1999, he has been a professor. He moved to the Graduate School of Information Science and Technology, Osaka University in April 2004. He has published more than 300 papers in international and domestic journals and conferences. His research interests include computer communication networks, performance modeling, and evaluation. He is a fellow of IEICE and a member of IEEE, the Association for Computing Machinery (ACM), The Internet Society, and IPSJ.