Proceedings of the International MultiConference of Engineers and Computer Scientists 2017 Vol II,
IMECS 2017, March 15 - 17, 2017, Hong Kong

Content Download Method with Distributed Cache Management

Masamitsu Iio, Kouji Hirata, and Miki Yamamoto

***Abstract*—This paper proposes a content download method with distributed cache management. Recently, in-network caching has attracted much attention, which will be used in the future network architecture such as content-centric networking. In networks with in-network caching, routers store contents in their cache and users can download them from the routers. By utilizing in-network caching, the loads for content servers and networks can be reduced. The proposed method aims at enhancing the effect of in-network caching. In the proposed method, management nodes belonging to routers manage location of cached contents in a distributed manner, which uses a distributed hash table (DHT). When a user downloads a content, the user sends a request for the content to a management node that has information on location of the content based on DHT. The request then goes to a router having the content according to the information on location. As a result, the proposed method can efficiently utilize cache of routers. Through simulation experiments, we show that the proposed method reduces the loads of content servers and networks.**

***Index Terms*—In-network caching, content download, distributed hash table, content-centric networking.**

I. Introduction

In recent years, the amount of traffic has rapidly increased because of the development of Internet applications. Multimedia content services such as video distribution have especially developed and they generate lots of traffic, which puts loads on both content servers and networks. In order to reduce the loads, several network architectures have been considered in the past. Content delivery networking (CDN) [1], [4] is currently used for content distribution, which distributes loads by storing copies of contents on multiple content servers. Furthermore, content-centric networking (CCN) [3] attracts much attention as a future network architecture. While CDN operates on IP networks based on host-to-host communications with IP addresses, CCN operates on content-oriented networks based on host-to-content communications. Specifically, CCN controls packet transmission with not IP addresses but content name.

One of the most important concepts in CCN is in-network caching [15]. In general, users send content requests to content servers and download the contents from them. In-network caching allows routers in networks to store contents in cache of them. When a content is downloaded from a content server, the content is cached in routers on its path according to replacement policy such as least-recently-used (LRU). Users can download the content from the routers in addition to content servers. Therefore, in-network caching alleviates access concentration in content servers and loads for network links.

In order to efficiently use cached contents, we should consider how users download them. A simple download method is passive download, in which users download only contents cached in routers along their paths to content servers, i.e., default paths. However, this method cannot utilize cache of routers that are not located on the paths, so that many packets arrive at content servers. It does not distribute loads efficiently. Furthermore, contents located around content servers are frequently replaced due to access concentration in the content servers, and thus cache utilization for popular contents decreases. We need an information sharing mechanism regarding location of cached contents in order to efficiently download the cached contents. Although it is ideal that all users know all the information on the cached contents and they directly download them, this case has large overhead to perform information retention and sharing. It is difficult to accommodate increasing traffic due to the growth of multimedia services.

This paper proposes a content download method with distributed cache management to overcome this difficulty. The proposed method aims at enhancing the effect of in-network caching. In the proposed method, management nodes connecting to routers manage location of cached contents in a distributed manner, which uses a distributed hash table (DHT) [7], [12]. When a user downloads a content, the user sends a request for the content to a management node that has information on location of the content based on DHT. The request then goes to a router having the content according to the information on location. As a result, the proposed method is expected to efficiently utilize cache of routers.

The rest of this paper is organized as follows. In Section II, we describe in-network caching. Section III explains our proposed method. In Section IV, the performance of the proposed method is discussed with the results of simulation experiments. Finally, we conclude the paper in Section V.

II. In-network caching

In-network caching enables intermediate routers to store contents. In CCN, in-network caching is assumed to be used. CCN uses a receiver-driven communication model. In this model, users send content request packets named Interest packets to content servers in order to obtain contents. The content servers send packets of the corresponding contents named Data packets back to the users. An Interest packet has a name of a content, and a Data packet has the corresponding name of the Interest packet. These packets are relayed by routers with a name-based forwarding mechanism. In CCN, a router consists of three components: forwarding information base (FIB), pending interest table (PIT), and content store

Manuscript received December 22, 2016; revised January 9, 2017. This work was partially supported by Grant-in Aid for Young Scientists (B) of the Japan Society for the Promotion of Science under Grant No. 15K15988.

M. Iio, K. Hirata, and M. Yamamoto are with Department of Electrical and Electronic Engineering, Faculty of Engineering Science, Kansai University, 564-8680 Japan (e-mail: {k775273,hirata,yama-m}@kansai-u.ac.jp).

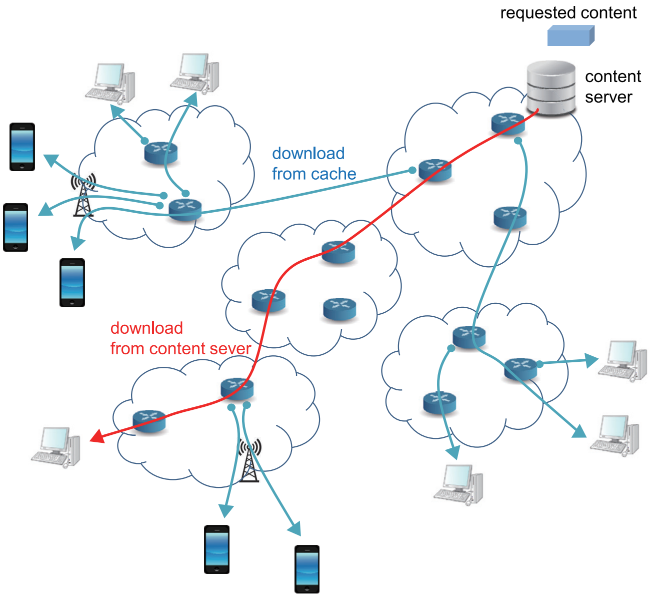

Fig. 1. In-network caching.

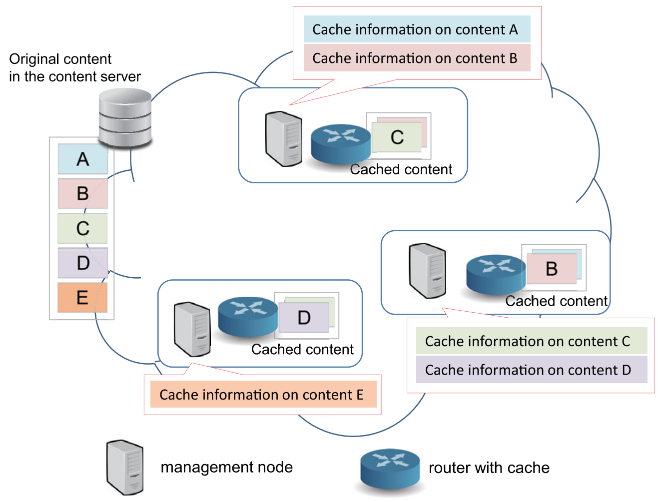

Fig. 2. Management nodes.

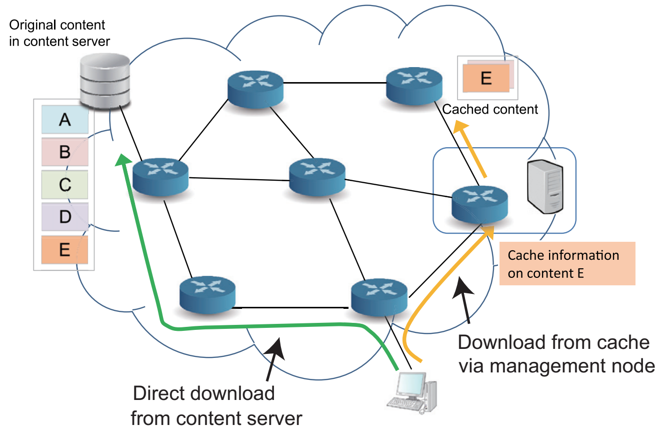

Fig. 3. Decision of packet destination.

(CS). The router forwards Interest packets based on name entries in FIB, which is a name prefix-based routing table for Interest packets. Data packets are transmitted based on PIT in the router, which represents reverse paths of the Interest packets.

The router can cache Data packets into CS as in-network caching in CCN. When the router receives an incoming Interest packet, it checks its CS. If a corresponding Data packet is stored there, the router replies it to the user; otherwise the router forwards the Interest packet to a content server. Because the length of the downloading path decreases, using CS reduces not only loads of content servers but also loads of network links as shown in Fig. 1.

In-network caching with CS is one of most important characteristics of CCN. In general, contents transmitted from content servers are temporarily and passively cached in intermediate routers. These cached contents are frequently replaced by other downloaded contents according to cache decision policies and replacement policies [9]. Note that the cache decision policies decide whether intermediate routers along downloading paths store contents or not. The cache replacement policies decide which cached contents are replaced with a new downloaded content. In order to efficiently use these cached contents, this paper focuses on content download methods [2], [5], [6], [8], [10], [11], [13]. When we use a simple download method that downloads contents cached in routers along paths from users to content servers, cache of routers that are not located on the paths cannot be utilized. On the other hand, when all users know all the information on the cached contents and they directly download them, the cached contents are efficiently utilized. However, this case has large overhead to perform information retention and sharing. To resolve these problems, the proposed method provides content downloading with distributed cache management.

III. Proposed Method

A. Overview

In the proposed method, we prepare management nodes, each of which belongs to a router and manages location of cached contents in a distributed manner, in a network. To do so, the proposed method uses DHT. As shown in Fig. 2, each management node manages a certain range of hash values. It keeps information on only contents whose hash values is in the range. When a user downloads a content, the user first checks the hash value of the content. The user decides whether it sends a request packet for the content such as Interest packets in CCN to a management node that manages the hash value or a content server that has the content according to the distances to them as shown in Fig. 3. Note that we assume that users or intermediate routers know which management nodes manage the hash value in advance. If the content server is located relatively near the user, the request packet is transmitted to the content server; otherwise the user sends the request packet to the management node. The management node receiving the request packet checks which routers cache the content. If there is a router having the content near the management node, the request packet is transmitted to the router and downloads the content from there. On the other hand, if there are no routers having the content near the management node, the request packet is transmitted to a content server having the content. As a result, the proposed method efficiently utilizes contents cached in intermediate routers.

The advantage of the proposed method is low control overhead. In the proposed method, when a content is cached or deleted in any of routers, each management node that manages the content has to update location of the content. In this case, a router that caches or deletes the content advertises the information to the management node. However, the router does not need to advertises the information to users and other management nodes. The users do not need to know location of contents, which means that control overhead is low.

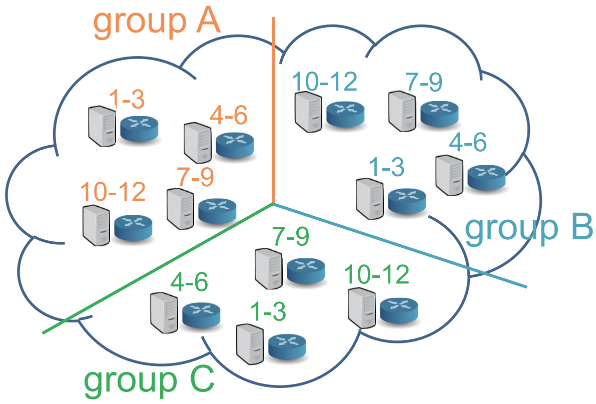

Fig. 4. Management group.

B. Procedure of the proposed method

1) Content download and information update: In the following, we explain the procedure of the proposed method, which consists of content download of users and information update of management nodes. Although the proposed method is assumed to be used for CCN, the idea of the proposed method can be applied to conventional networking architectures such as IP networking. Therefore, this paper introduces a general procedure of content download in the proposed method and does not focus on detailed operation for CCN.

We assume that there are some management nodes connecting to routers in a network and each management node has information on location of cached contents, hash values of which are in the range of the hash values managed by the management node. We also assume that each user or router knows location of content servers, management nodes, and hash values managed by the management nodes. Note that each user or router does not know location of contents cached in routers. When a user download a content, the proposed method performs the following procedure.

Step (1) The user checks the hash value of the content and the number h_n of hops between the user and the nearest management node that manages the hash value.

Step (2) If $h_c < h_n + k$, the user sends a request packet for the content to the nearest content server that has the content and downloads the content from the content server, where h_c denotes the number of hops between the user and the content server and k is a parameter. In this case, the procedure finishes.

Step (3) If $h_c \geq h_n + k$, the user sends the request packet to the management node.

Step (4) After the management node receives the request packet, it checks which routers cache the content now. It also checks the number of hops to the nearest router having the content.

Step (5) If the number of hops to the router is larger than the number of hops to the nearest content server or there are no routers that have the content, the management node sends the request packet to the content server. Then the content is downloaded from the content server.

Step (6) Otherwise, the management node sends the request packet to the nearest router having the content, and then the content is downloaded from the router.

Note that in Steps (2) and (3), when the content is cached in an intermediate router along the path to the content server or the management node, the content is downloaded from there, i.e., performing passive download. In Step (2), the user sends the request packet to the content server because the content server is relatively near compared with the management node. We can adjust the decision with parameter k.

As mentioned earlier, this paper does not focus on detailed operation of the proposed method for CCN. If the proposed method is used for CCN, users and management nodes need assist by intermediate routers because CCN assumes that users do not know location of content servers and packets are forwarded by routers based on content names. However, the operation of the proposed method follows the above general procedure.

In the proposed method, each management node forwards request packets transmitted by users based on information on current location of contents. Therefore, when a content, the hash value of which is in the range of the hash values managed by the management node, is cached or deleted in a router, the information should be updated. Therefore, whenever a content in a router is cached or deleted, the router advertises update information to management nodes that manage the updated content. By doing so, the management nodes know current location of the content and the above procedure of the proposed method works well.

2) Grouping of management nodes: We should consider the range of hash values managed by each management node in order to efficiently use the proposed method. If a content is managed by only one management node in a network, arrival of request packets concentrates in the management node. Therefore, the load of the management node becomes heavy. Moreover, the number of hops between each user and the management node tends to be large. On the other hand, if a content is managed by many management nodes, each management node needs to have much information and control overhead becomes high.

To balance loads of management nodes and control overhead, the proposed method divides management nodes into m groups. Management nodes belonging to the same group manage different contents. On the other hand, management nodes belonging to different groups can manage common contents. We show an example with Fig. 4. In this figure,

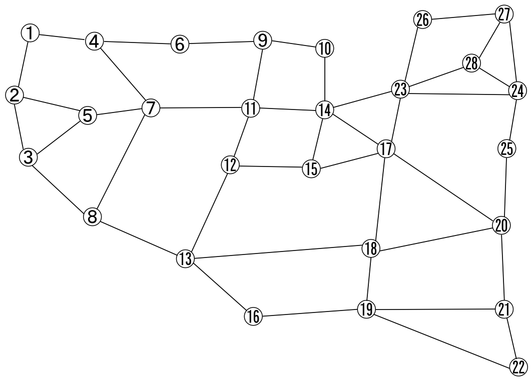

Fig. 5. USA topology.

we divide management nodes into three groups, where each router has a management node and there are 12 types of contents. While management nodes belonging to the same group manage three different contents, some management nodes belonging to different groups manage common contents. By doing so, we can reduce loads for management nodes and the number of hops between users and management nodes. This paper uses a simple dividing strategy that divides management nodes according to node indices. Specifically, management node i ($i = 1, 2, \ldots, N$) belongs to group $\lceil im/N \rceil$.

IV. Performance evaluation

A. Model

To evaluate the performance of the proposed method, we conduct simulation experiments. We use two network topologies. The first one is a USA topology shown in Fig. 5, which consists of 28 routers and 45 bi-directional links. Each router connects to a user, a content server, and a management node. There exist 112 types of contents in the network. Each content server has 4 original contents. Different content servers have different contents and each content is stored in one of content servers. Each router can cache 10 different contents. The second one is a topology made by the Watts-Strogatz (WS) model [14], which consists of 100 routers and 400 bi-directional links. As the parameters in the WS model, the mean degree a of nodes is set to be 4 and the probability p that each link is replaced is set to be 0.1. In this network, there exists 1000 types of contents. Each content server has 10 original contents and each router can cache 10 different contents. The system parameters are listed in Table I.

In each simulation experiment, we randomly select a combination of a user and a content. Then the content is downloaded by the procedure of the proposed method. The router to which the user belongs replaces one of its cached contents with the downloaded content based on LRU while intermediate routers do not replace their cached contents. In the proposed method, management nodes manage the same number of contents. The simulation experiment finishes after it repeats this process one hundred thousand times. For the sake of comparison, we use a passive download method that does not use management nodes and always sends request packets to content servers. Although this method uses in-network caching as well as the proposed method, it downloads only contents cached in routers along paths between users and content servers.

TABLE I
System parameters.

Parameters	USA	WS
Number of nodes	28	100
Number of links	45	400
Number of types of contents	112	1000
Number of original contents in each content server	4	10
Number of contents that each router can cache	10	10

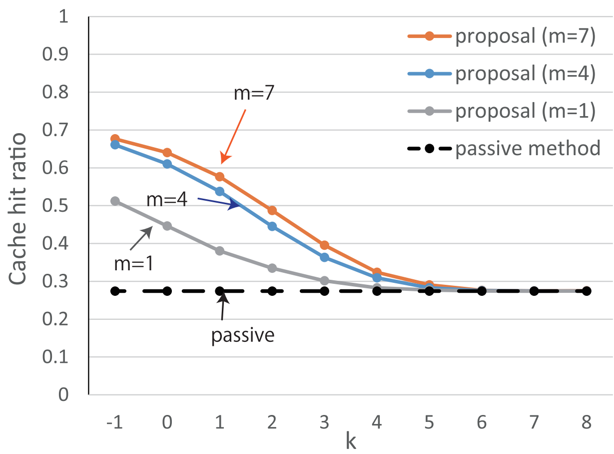

Fig. 6. Cache hit ratio (USA topology model).

In the simulation experiments, we assume that the length of every link is the same because we focus on the number of hops. Furthermore, since we focus on cache utilization, we do not consider propagation delay of links and processing time of content servers and management nodes. Also, we do not consider bandwidth of links and size of contents.

B. Results

Figure 6 shows the cache hit ratio as a function of the parameter k of the proposed method in the USA topology model. We define the cache hit ratio as the ratio of the number of contents downloaded from intermediate routers to the total number of downloaded contents. Note that the performance of the passive method does not depend on the value of parameter k, so that the cache hit ratio of the passive method has the identical value for any k. As we can see from this figure, the cache hit ratio of the proposed method decreases with the increase in the value of parameter k. In the proposed method, large k means that users tend to send request packets to content servers directly because the users send request packets to content servers if $h_c < h_n + k$. Therefore, when k is large, the proposed method does not efficiently utilize management nodes and thus the cache hit ratio decreases. For $k \geq 6$, the cache hit ratio of the proposed method is almost the same as that of the passive download method. This result indicates that the proposed method always directly sends request packets to content servers as well as the passive download method when $k \geq 6$.

Moreover, we observe that the cache hit ratio of the proposed method increases with the number m of management groups. This is because there exist many management nodes that manage location of common contents. Specifically, the

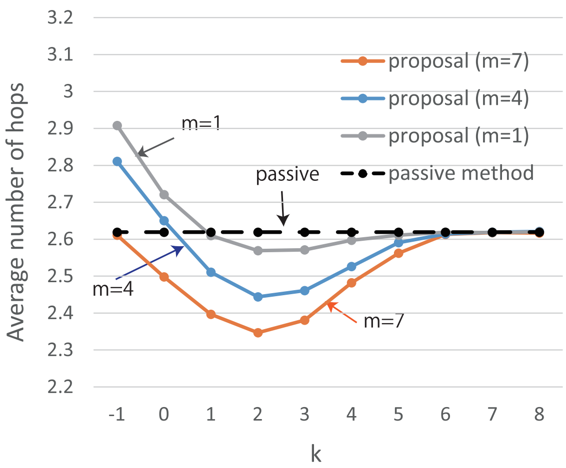

Fig. 7. Average number of hops (USA topology model).

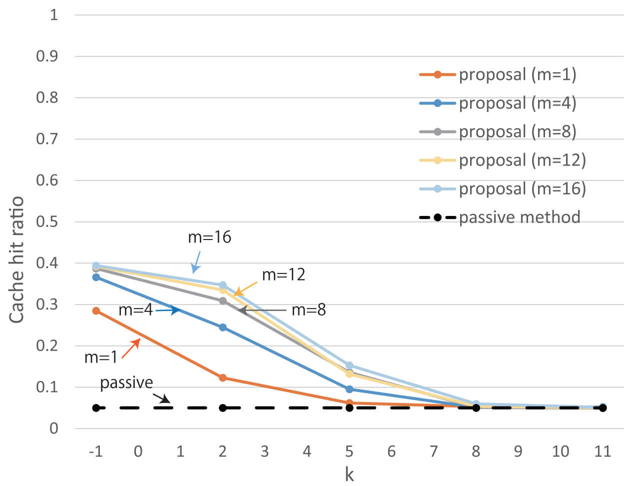

Fig. 8. Cache hit ratio (WS model).

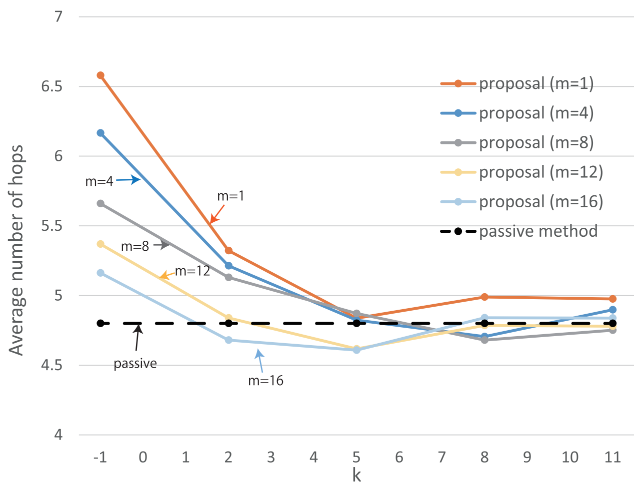

Fig. 9. Average number of hops (WS model).

average number of hops from users to management nodes tends to be small when m is large, and thus the condition $h_c \geq h_n + k$ in Step (3) is easily satisfied. We also observe that while the cache hit ratio for $m = 4$ is significantly larger than that for $m = 1$, it is slightly smaller than that for $m = 7$. This result indicates that $m = 4$ is better than $m = 7$ in this case to suppress the control overhead of the proposed method.

Figure 7 shows the average number of hops of content download as a function of the parameter k of the proposed method in the USA topology model. From this figure, we observe that the average number of hops decreases with the value of the parameter k until k reaches 2. Then the average number of hops increases with the value of the parameter k. When k is small, i.e., $k \leq 0$, the proposed method sends request packets to management nodes even if the number of hops to a content server is smaller than that to a management node. Therefore, the average number of hops increases in this case. On the other hand, when k is large, i.e., $k \geq 3$, the proposed method tends to directly send request packets to content servers even if the number of hops to a content server is larger than a management node. In this case, the proposed method cannot utilize management nodes and cached contents. As a result, the average number of hops of content download increases. When $k \geq 6$, the performance of the proposed method is the same as the passive download method. When $k = 2$, the proposed method shows the best performance for any value of m, which is smaller than the average number of hops of the passive download method.

We observe that the average number of hops of the proposed method decreases with the increase in the number m of management groups. The reason is that the average number of hops from users to management nodes is small for large m and users efficiently utilizes cached contents. However, as mentioned earlier, the control overhead is high for large m and the cache hit ratio is not proportional to the value of m as shown in Fig. 6. Therefore, we need to appropriately select the value of m according to requirements of network design.

We then examine the performance of the proposed method in the WS model. In this model, we collect 10 independent samples from simulation experiments and obtain the average of them. Note that for each experiment, the topology is randomly constructed by the WS model. Figure 8 shows the cache hit ratio as a function of the parameter k of the proposed method in the WS model. As shown in this figure, the cache hit ratio decreases with the increase in the value of parameter k and it increases with the number m of management groups, similar to the result in the USA topology model. We also observe that the cache hit ratio is almost the same for $m \geq 8$.

Figure 9 shows the average number of hops of content download as a function of the parameter k of the proposed method in the WS model. As we can see from this figure, the average number of hops is minimized when k is about 8 for any value of m. We observe that the average number of hops decreases with the increase in the number m of management groups. This result is also similar to the result in the USA topology model.

V. Conclusion

This paper proposed a content download method with distributed cache management, which aims at enhancing the

ISBN: 978-988-14047-7-0
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

effect of in-network caching. Through simulation experiments, we showed the that proposed method reduces the average number of hops of content download and enhance the cache hit ratio. As the future work, we will consider the impact of system parameters such as propagation delay and bandwidth of links.

REFERENCES

[1] F. Douglis and M. Kaashoek, "Scalable internet services," *IEEE Internet Computing*, vol. 5, no. 4, pp. 36–37, 2001.

[2] A. Ghodsi, S. Shenker, T. Koponen, A. Singla, B. Raghavan, and J. Wilcox, "Information-centric networking: seeing the forest for the trees," in *Proc. the 10th ACM Workshop on Hot Topics in Networks*, Cambridge, MA, Nov. 2011.

[3] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard, "Networking named content," in *Proc. the 5th International Conference on Emerging Networking Experiments and Technologies (CoNEXT 2009)*, Rome, Italy, Dec. 2009, pp. 1–12.

[4] N. Laoutaris, V. Zissimopoulos, and I. Stavrakakis, "On the optimization of storage capacity allocation for content distribution," *Computer Networks*, vol. 47, no. 3, pp. 409–428, 2005.

[5] M. Lee, K. Cho, K. Park, T. Kwon, and Y. Choi, "SCAN: scalable content routing for content-aware networking," in *Proc. IEEE International Conference on Communications (ICC 2011)*, Kyoto, Japan, Jun. 2011.

[6] S. Lee, D. Kim, Y. Ko, J. Kim, and M. Jang, "Cache capacity-aware CCN: selective caching and cache-aware routing," in *Proc. IEEE Global Communications Conference (GLOBECOM 2013)*, Atlanta, GA, Dec. 2013.

[7] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker, "A scalable content-addressable network," in *Proc. ACM SIGCOMM*, San Diego, CA, Aug. 2001, pp. 161–172.

[8] G. Rossini and D. Rossi, "Evaluating CCN multi-path interest forwarding strategies," *Computer Communications*, vol. 36, no. 7, pp. 771-778, 2013.

[9] G. Rossini and D. Rossi, "Coupling caching and forwarding: benefits, analysis, and implementation," in *Proc. the 1st ACM Conference on Information-Centric Networking*, Paris, France, Sep. 2014, pp. 127–136.

[10] L. Saino, I. Psaras, and G. Pavlou, "Hash-routing schemes for information centric networking," in *Proc. the 3rd ACM SIGCOMM Workshop on Information-Centric Networking*, New York, NY, Aug. 2013, pp. 27–32.

[11] S. Saha, A. Lukyanenko, and A. Jaaski, "Cooperative caching through routing control in information-centric networks," in *Proc. INFOCOM 2013*, Turin, Italy, Apr. 2013, pp. 100–104.

[12] I. Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan, "Chord: a scalable peer-to-peer lookup service for internet applications," in *Proc. ACM SIGCOMM*, San Diego, CA, Aug. 2001, pp. 149–160.

[13] Y. Wang, K. Lee, B. Venkataraman, R. Shamanna, I. Rhee, and S. Yang, "Advertising cached contents in the control plane: Necessity and feasibility," in *Proc. IEEE INFOCOM 2012 Workshop*, Orlando, FL, Mar. 2012.

[14] D. Watts and S. Strogatz, "Collective dynamics of 'small world' networks," *Nature*, vol. 393, pp. 440–442, 1998.

[15] M. Yamamoto, "A survey of caching networks in content oriented networks," *IEICE Transactions on Communications*, vol. E99-B, no. 5, pp. 961–972, 2016.