Routing Information Management for Content Oriented Networks Using Bloom Filters

Kenji Komatsu
Tokyo Metropolitan University
Graduate School of System Design
6-6 Asahigaoka, Hino, Tokyo 191-0065, JAPAN
Email: komatsu-kenji1@sd.tmu.ac.jp

Takuya Asaka
Tokyo Metropolitan University
Faculty of System Design
6-6 Asahigaoka, Hino, Tokyo 191-0065, JAPAN
Email: asaka@sd.tmu.ac.jp

***Abstract*—Content oriented networks are proposed as a new network architecture that does not rely on IP addresses. In these networks, content queries are routed based on the content name itself instead of a destination address. However, one problem is the increased memory usage that occurs with content oriented networks. This is because a huge amount of content exists in the network, and an entry is created in the routing table for each content item. One technique to address this problem has attracted attention because it employs a routing scheme using Bloom filters, which are data structures capable of compressing data. However, this technique can-not cope with content that moves or is deleted. To solve this problem, we propose a method that can cope with moving or deleted content by using two bloom filters. Simulations we conducted to evaluate the proposed scheme indicated the method was more effective than two other schemes.**

Keywords*—*Content delivery, Content oriented, Bloom filter

I. INTRODUCTION

Internet traffic for distributing content such as text, music and pictures is increasing. Each node performs routing using IP address and lead to detect content in the Internet. The architecture of the Internet has been referred to as location oriented architecture because IP addresses include information about the location. However, the application that clients use to obtain content has changed from a client-server model to a content delivery network (CDN) [1] and peer-to-peer (P2P) [2] model to achieve load balancing.

In CDNs, some servers that have the same content. A content query is sent to the appropriate server, for example the nearest server or a lightly loaded server, by the DNS (Domain Name System). A P2P overlay network is a network system composed of a terminal peer that can behave as both a server and a client. A content search is performed in the overlay network, and the requested content is sent from an other client that has it. In particular, content distribution is performed using chunks, which are the pieces that content is split into on BitTorrent [3] protocol. That is to say, clients obtain content by gathering chunks of content from multiple content sources.

CDN and P2P are a content oriented model as opposed to a client-server model, which is a location oriented model. It is not important where the clients obtain content from in terms of content distribution. The above indicates that the Internet architecture is a different model from one of application for contents distribution. This difference leads to inefficiencies and overhead in content distribution. To solve these problems, content oriented network is proposed [4][5]. A content oriented network is a network architecture that is based on a content ID instead of IP address. A routing table entry is the content ID in a content oriented network. However, the significant increase in the number of routing table entries is one of the problems with content oriented networks. Thus, routing schemes using Bloom filters (BFs) [6], which are data structures capable of compressing data, have attracted attention [7][8]. One problem, however, is that such schemes can-not cope with content that is moving or being deleted. In this paper, to solve this problem, we propose a scheme that can cope with moving or deleted content by using two BFs.

The rest of this paper is as follows. Section 2 reviews related research on content oriented networks and routing schemes that use BFs. In Section 3, we describe the details of our proposed scheme, and in Section 4 we report on simulations we carried out to evaluate effectiveness. Finally, we conclude the paper in Section 5.

II. RELATED WORKS

A. *Content oriented networks*

A content oriented network is a network architecture in which network control is based on a content ID instead of an IP address. Content sources broadcast their content information to all nodes in the network through flooding, and each node sets a routing table based on the content information. In this paper, process of broadcasting content information to all nodes in the network through flooding is referred to as ”advertising”. A routing table entry is the content ID in a content oriented network. Each node checks its routing table and forwards a query to detect the content (Fig. 1).

A content oriented network is classified as an architecture that focuses only on content detection or one that focuses on both content detection and provision. i3 (Internet Indirection Infrastructure) [9], DONA (Data-Oriented Network Architecture) [10], and TRIAD [11] are architectures that focus on content detection. In particular, TRIAD is said to be the subject of a pioneering study. CCN (Content Centric Network) [12][13] is an architecture focusing on both content detection and provision. In CCN, each node has a table called a pending interest table (PIT) that stores information indicating where the content query has passed and content provision is performed by using the PIT. To realize a content oriented network, it is

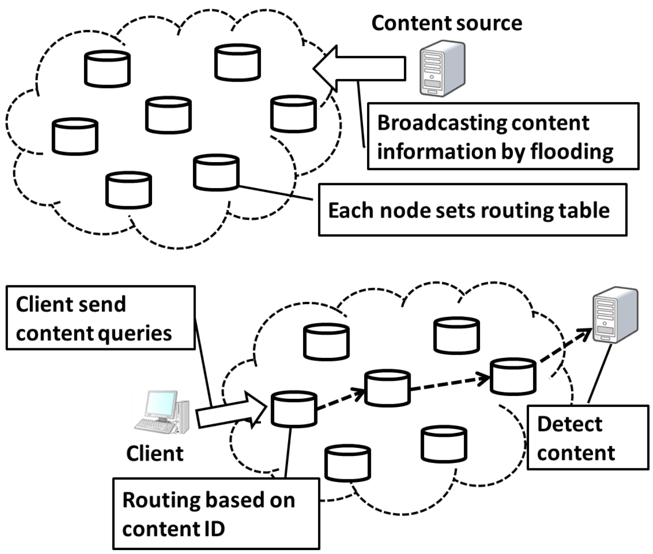

Fig. 1. Overview of content oriented network.

necessary to research and develop a method of naming the content ID [14], routing scheme [15][16], use of the cache [17][18], network control protocol [19][20] and other tasks.

B. Routing scheme using Bloom filters (BFs)

A large amount of content exists in a network but the memory of each node is limited. The significant increase in the number of routing table entries is one of the problems that occurs in content oriented networks. Routing schemes using BFs, i.e., data structures capable of compressing data have therefore attracted a lot of attention [6].

A BF is a space-efficient data structure that is used to test whether an element is a member of a BF. An empty BF is a bit array of any number of bits, all set to 0. The process to add an element is as follows, first, the elements is fed to each of the hash functions to get the array positions. The bits at these positions are set to one. To test whether an element is in the BF, it is fed to each of the hash functions to check the position in the BF. Some problems with BFs are that the positions in a BF may have by chance been set to 1 during the insertion of other elements. This result in a false positive and makes it impossible to remove an element from the BF because false negatives are not permitted. In an existing study, the amount of memory usage is about seventy five percent reduction by using BFs[8].

In an existing routing scheme using BFs in a content oriented network, the links of all nodes have a BF that store content information that content sources broadcast by flooding in their BF. When a content query arrives at the node, the node determines the next hop by checking all the BFs it has. For example, in Fig. 2, the query including hash:01 arrives at the node and the node determines that the BF of the link on the right is storing content information that has hash:01 by checking all the BFs. That node then forwards the query to the right side.

However, routing schemes that use BFs in a content oriented network are problematic in that they can-not cope with

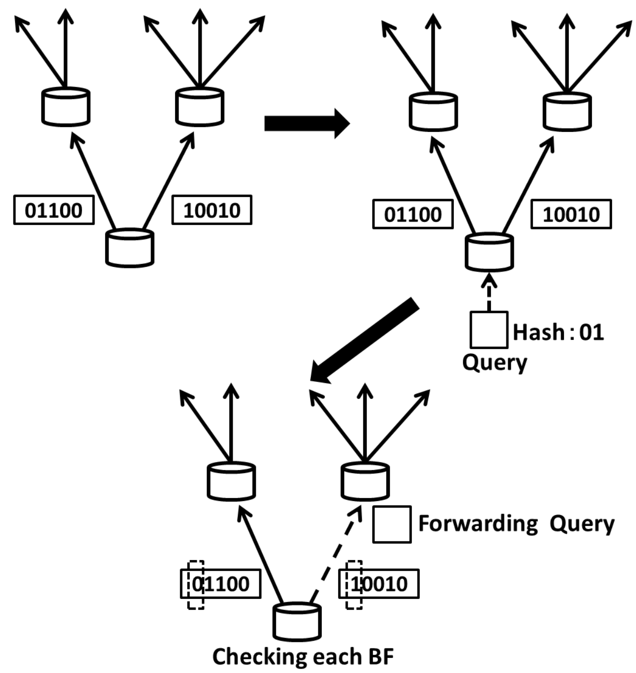

Fig. 2. Process of determining next hop in routing scheme using Bloom filters.

content that is moving or being deleted, because removing an element from a BF is impossible. When content sources are mobile terminals, it is possible that some content will be moving content [21][22].

III. Proposed Method

This section describes a routing information management scheme that can cope with moving content by using two BFs in a content oriented network. The links of all nodes have two BFs (Fig. 3), and content information is stored in both BFs. The proposed scheme clears the two BFs alternately and advertises content information regularly. The scheme repeats the advertising and clearing of the BFs, which prevents unnecessary information from accumulating, which might cause queries to be forwarded on the wrong route, and it also updates the states of all BFs.

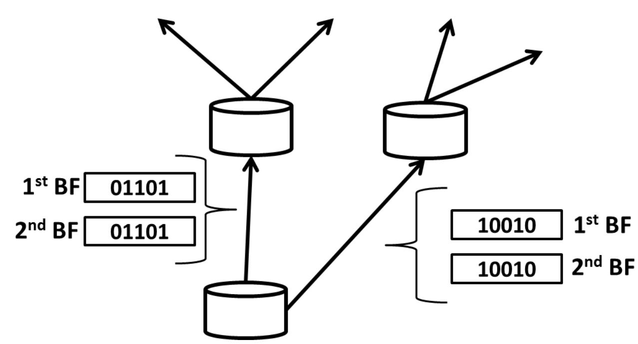

Fig. 3. Schematic diagram of nodes having two bloom filters.

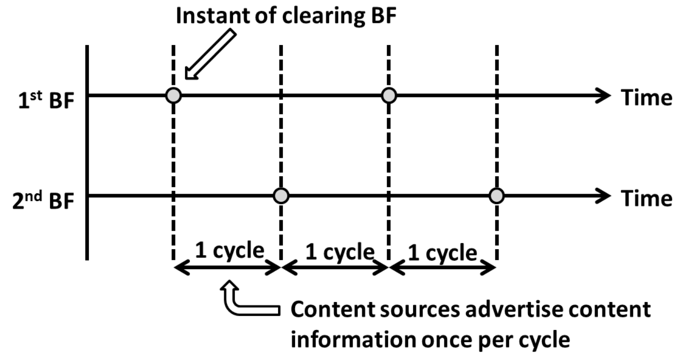

Fig. 4. Cycle of clearing BFs and advertising content information.

We show the instant of clearing BFs and advertising content information in Fig. 4. The second BF is cleared at a fixed period after clearing first BF. The first BF is cleared at a fixed period after clearing second BF. Content sources advertise their content information once in the period between clearing first BF and clearing second BF by broadcasting. Furthermore, content sources advertise their content information when they move to different place. As a result, unnecessary information is deleted from the network, and necessary information is stored in either of second BFs.

Next, we describe the process that the node determine the next-hop node. When a content query arrives at the node, the node checks all the BFs it has. Two BFs that a specific link has store requested content information, the node determine the link as the next-hop preferentially. For example, in Fig. 3, when the query including hash:01 arrives at the node, the node checks their BFs and infer that first and second BFs of the right link store the requested content information(that has hash:01). Therefore that node forwards the query to the right side. If it's not possible to determine the next hop in the above procedure, the node checks either of first BF or second BF and determine the next-hop node. Each nodes perform the above process repeatedly, and lead to detect the requested content.

In the proposed scheme, if content information is advertised twice in one cycle from different locations, the plural BFs of the nodes would have the same state (Fig. 5). Therefore, the nodes cannot determine the correct next-hop node. The states of each node continue until the BFs have been cleared once and the content information has been advertised once. In this case, each node must determine the next hop randomly.

In the proposed scheme, routing information is compressed by using BFs. Moreover, each node can cope with moving or deleted content. However, there are problems that number of times of advertising content information from content sources increase, and the plural BFs of the nodes would have the same state.

IV. Simulation

In this section, we describe the simulation model and the results of evaluating the proposed scheme. Table I lists the parameter values used in the simulation. A lot of simulations have been carried out in the existing research, but simulations in a large scale network is not performed much. Therefore, in this time, we performed simulations in a small scale network. The content ID was a flat ID, for example, /content-title.png/, in the simulation. Each content item had one hash, and each node stored hashes of content in their BF. Content sources once moved to random locations in simulation time. The content query was sent at the random location every unit time. Next, about a network topology construction, each node is placed to a random location and establishes links with other nodes in the order of a near distance. Each node has at least two links, and has a maximum of five links. Content sources are placed to random location and establish one link with other node.

TABLE I. PARAMETER VALUES IN SIMULATION

Heading	Values
Simulation time [unit time]	500
Number of nodes	100 · 1000
Network topology	random allocation
Number of content items	50 · 100 · 300 · 500
BF size [bit]	10 · 30 · 50 · 70 · 100
Length of advertising cycles [unit time]	10 · 20 · 30 · 40 · 50

We compare the proposed scheme with schemes 1 and 2 in this paper, because we confirm that more efficient searching is possible with the proposed scheme than an existing routing scheme using BFs. even when the content moved in the network. In scheme 1, each link in a nodes have one BF, all nodes clear BFs and all content sources broadcast their content information at regular cycles. In scheme 2, each link in a node have one BF, but the nodes do not clear the BFs. Scheme 2 is a simple scheme using BFs in a content oriented network. In this study, we do not compare

We plot the results of the simulation in Figs. 6 and 7. The results in Fig. 1 are for the case when there were 100 content items, the BF size was 50, the length of advertising cycle was 10 and there were 100 nodes. The details are the same for the results in Fig. 7, except there were 1000 nodes. In the proposed scheme, each node deletes unnecessary information and retains the latest content information in either of the two BFs by regularly advertising content information and clearing BFs. Accordingly, we confirmed that a more efficient search than in schemes 1 and 2 was possible in spite of content sources moving from Fig. 6 and Fig. 7. In scheme 1, the

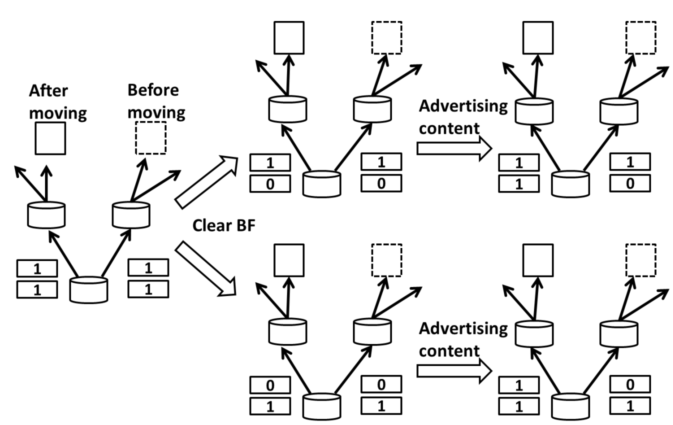

Fig. 5. The state in which nodes cannot determine the next-hop node in the proposed scheme.

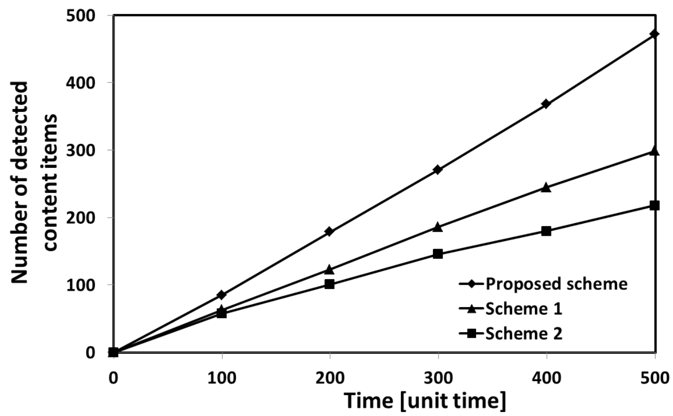

Fig. 6. Number of detected content items when number of nodes was 100.

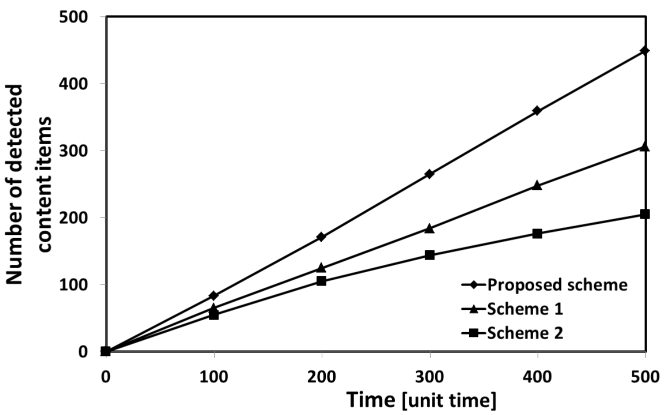

Fig. 7. Number of detected content items when number of nodes was 1000.

amount of unnecessary information continued to increase in the network until all nodes cleared their BFs and all content sources broadcast their content information. In scheme 2, since each node did not clear their BFs, the amount of unnecessary information continued to increase. Therefore, the number of detected content items was less than in the proposed scheme.

A. Simulation1

We show the results of the simulation in Fig. 8. There were 100 nodes, 100 content items, the length of advertising cycle was 10, and the BF sizes were 10, 20, 30, 40 and 50. We confirmed that the content detection rate decreased as the BF size decreased. The amount of data to be stored in the BF was small, when the BF size is small. In this case, multiple content items have the same hash and are stored in the same location within the BF. It is not possible to retrieve the original content information from the BF. Therefore, each node transfers the content query to the location where the other content item exists. We must set an adequate BF size. However, we must not set the BF size too large because only setting the BF consumes the node memory.

B. Simulation2

We show the results of the simulation in Fig. 9. There were 100 nodes, the BF sizes was 50, the length of advertising cycle was 10 and there were 50, 100, 300 and 500 content items.

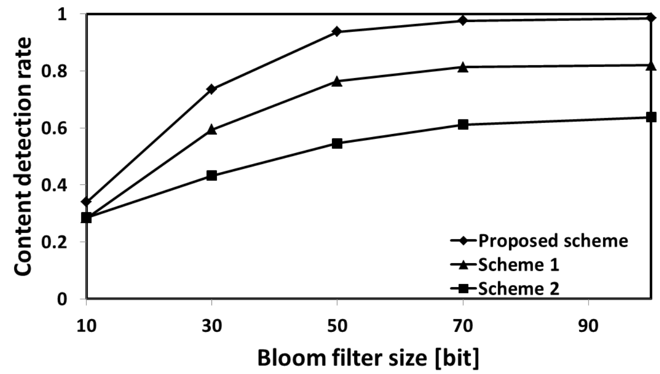

Fig. 8. Content detection rate when the BF size was changed.

We confirmed that the content detection rate decreased as the number of content items increasing. In this case, multiple content items have the same hash and are stored in the same location within the BF, so it is not possible to retrieve the original content information from the BF. Therefore, we must set an adequate BF size.

C. Simulation3

The results of this simulation are plotted in Fig. 10. There were 100 nodes, 100 content items, the BF sizes was 50, and the length of advertising cycle were 10, 20, 30, 40 and 50.

We confirmed that the content detection rate decreased as the advertising cycles becames longer. Lengthening the advertising cycles leads to a reduction in the number of cleared BFs. Consequently, the amount of unnecessary information that was forwarded to places where content existed before it moved increased in the network. The links each node has were in the same state because of the increased amount of unnecessary information. Therefore, the nodes could not determine the correct next-hop node. With shorter advertising cycles, the BFs are cleared more often, and content information from content sources is advertised more frequently. Thus, all nodes almost always have the latest information. A shorter advertising cycle is better in terms of content detection, but

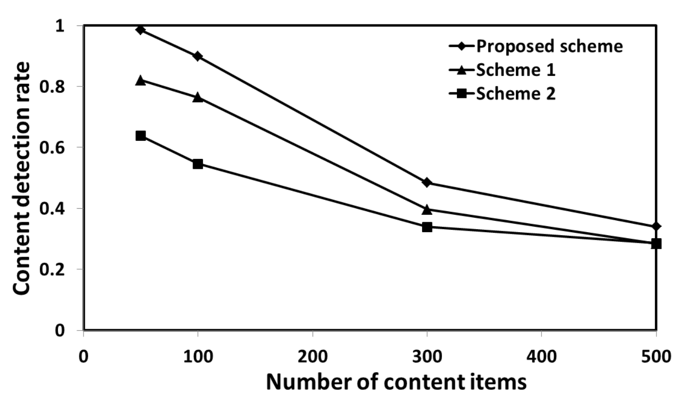

Fig. 9. Content detection rate when number of content items varied.

advertising content information more frequently increases the amount of network traffic and the network load. Accordingly, we must set an appropriate advertising cycle.

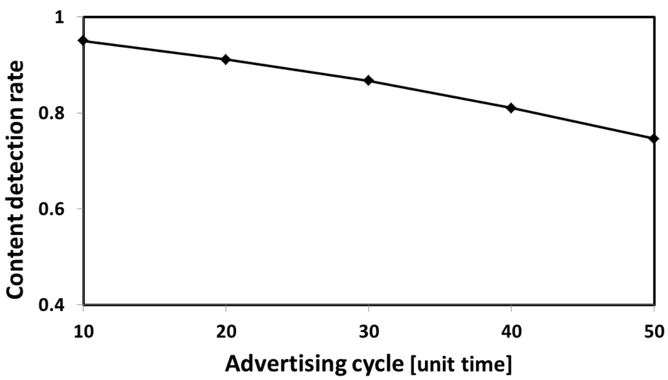

Fig. 10. Content detection rate when the length of advertising cycle was varied.

V. Conclusion

In this study, we proposed a routing information management method that can cope with content that moves by using two BFs in a content oriented network. We confirmed through simulation that efficient searching was possible with the proposed scheme even when the content moved in the network. We also found that in the proposed scheme, it was necessary to appropriately set each parameters in order to realize target figures (e.g., network load, content detection rate).

Nevertheless, our proposed scheme has some problems that need to be solved. These include coping with the state where the nodes can-not determine the correct next-hop node when the content source broadcasts content information twice from different locations in one cycle, evaluating the influence of network topology, and establishing a scheme that can cope with an increasing amount of content. In addition, for confirming that the amount of memory usage is reduced by using BFs, it is needed to compare memory usage between the proposed scheme with adequate BFs size and the other scheme that not using Bloom Filters.

References

[1] F. Douglis and M. Kaashoek. Scalable Internet Services. IEEE Intenet Computing, pp. 36-37, Vol. 5, No. 4, July 2001.

[2] E. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim. A Survey and Comparison of Peer-to-Peer Overlay Network Schemes. IEEE Communications Survey and Tutorials, pp. 72-93, Vol. 7, No. 2, 2005.

[3] BitTorrent. Inc. http://www.bittirrent.com

[4] B. Ahlgren, C. Dannewitz and C. Imbrenda. A Survey of Information-Centric Networking. IEEE Communications, pp. 26-36, Vol. 50, 2012.

[5] M. F. Bari, S. Chowdhury, R. Ahmed, R. Boutaba and B. Mathieu. A Survey of Naming and Routing in Information-Centric Networks. IEEE Communications. pp. 44-53. 2012.

[6] A. Broder and M. Mitzenmacher. Network Applications of Bloom Filters:A Survey. Internet Mathematics, pp. 485-509, Vol. 1, No. 4, 2004.

[7] M. Yu, A. Fabrikant, and J. Rexford. BUFFALO: Bloom Filter Forwarding Architecture for Large Organizations. CoNEXT ' 09, pp. 313-324, 2009.

[8] C. Esteve, F. L. Verdi, and M. F. Magalhaes. Towards a New Generation of Information-Oriented Interworking Architecture. ACM CoNEXT '08, No. 65, 2008.

[9] I. Stocia, D. Adkins, S. Zhuang, S. Shenker, and S. Surana. Internet Indirection Infrastructure. ACM SIGCOMM '02, pp. 73-86, 2002.

[10] T. Koponen, M. Chawla, B. Chun, A. Ermolinskiy, K. Kim, S. Shenker, and I. Stoica. A Data-Oriented Network Architecture. ACN SIGCOMM '07, pp. 181-192, 2007.

[11] M. Gritter and D. Chariton. TRIAD: A New Next-Generation Internet Architecture. http://www-dsg.stanford.edu/triad, 2000.

[12] V. Jacobson, D. K. Smetters, and J. D. Thornton. Network Named Content. CoNEXT '09, pp. 1-12, 2009.

[13] G. Carofiglio, M. Gallo, L. Muscariello and D. Perino. Modeling Data Transfer in Content Centric Networking. Teletraffic Congress (ITC), pp. 111-118, 2011.

[14] J. Choi, H. Jinyoung, C. Eunsang and T. Kwon. A Survey on Content-Oriented Networking for Efficient Content Delivery. IEEE Communications. pp. 123-127, Vol. 50. 2011

[15] V. Dimitrov and V. Koptchev. PRISP Project: Publish-Subscribe Intenet Routing Paradigm. Int'l Conf Comp and Tech, pp. 167-171, 2010.

[16] J. Lee, D. Y. Kim, M. Jang and B. Lee. Mobility Management for Mobile Consumer Devices in Content Centric Networking (CCN). IEEE International Conference Consumer Electronics (ICCE), pp. 502-503, 2012.

[17] E. Rosenweig and J. Kurose. Breadcrumbs: Efficient, Best-Effort Content Location in Cache Networks. IEEE INFOCOM '09, pp. 2631-2635, 2009.

[18] Y. Zhu, M. Chen and A. Nakano. Conic: Content-Centric Network with Indexed Caching. IEEE INFOCOM, pp. 1-6, 2010.

[19] D. Corujo, R. Aguiar, I. Videl and J. Reinoso. A Named Data Networking Flexible Framework for Management Communications. IEEE Communications. pp. 36-43. 2012.

[20] P. K. Agyapong, M. Sirbu. Economic Incentives in Information-Centric Networking: Implications for Protocol Design and Public Policy. IEEE Communications. pp. 18-26. 2012.

[21] D. Kim, J. Kim, Y. Kim, H. Yoon and I. Yeon. Mobility Support in Content Centric Networks. ICN '12, pp. 13-18, 2012.

[22] J. Lee, S. Cho and D. Kim. Device Mobility Management in Content-Centric Networking. IEEE Communications. pp. 28-34, 2012.

[23] H. Xie, Y. Yang, A. Krishnamurthy, Y. Liu, and A. Silbershatz. P4P: Provider Portal for Applications. ACM SIGCOMM '08, pp. 113-120, 2008.