ISSN 2047-3338

Mobility Support Using Bloom Filter in Content Oriented Networks

Kenji Komatsu[1] and Takuya Asaka[2]
[1,2]Tokyo Metropolitan University, Japan
Email: {komatsu-kenji, asala}@sd.tmu.ac.jp

***Abstract*— Content oriented networks have been proposed as a new network architecture that does not rely on IP addresses, and these networks, content queries are routed based on the content name itself instead of a destination address. However, one problem that occurs with content oriented networks is the increased memory usage. 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 is a routing scheme using Bloom filters, which are data structures capable of compressing data. However, this technique cannot cope with content that is moved or deleted. To solve this problem, we propose a method that can cope with moved or deleted content by using two Bloom filters. Simulations we conducted to evaluate the proposed scheme indicated the method was more effective than conventional schemes.**

***Index Terms*—Content Delivery, Content Oriented Networks, Bloom Filter and Mobility**

I. Introduction

INTERNET traffic for distributing content such as text, music, and pictures is increasing. Each node performs routing using IP addresses, which makes it possible to detect content on 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 may 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 another client that has it. In particular, content distribution is performed using chunks, which are the pieces that content is split into on the BitTorrent protocol [3]. That is to say, clients obtain content by gathering chunks of content from multiple content sources.

Both CDN and P2P are content oriented models 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. This indicates that the Internet architecture is a different model from the application model for content distribution. This difference leads to inefficiencies and overhead in content distribution. Content oriented networking has been proposed in order to solve these problems [4], [5]. The architecture of a content oriented network is based on content IDs instead of IP addresses. 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 these schemes cannot cope with content that is moved or deleted. In this paper, we propose a scheme to solve this problem. Our scheme uses two BFs in order to cope with moved or deleted content.

Our proposed scheme has the following three features:

1. The links of all nodes have two BFs, and content information is stored in both of them.
2. Each node clears the two BFs alternately.
3. When content is moved, the content source broadcasts content information such as the content ID.

In existing routing schemes using one BF, content information remains in the BF after the content has been moved because it is not possible to erase only the information of the moved content. This unnecessary information can cause queries to be forwarded on the wrong route. This phenomenon is increasing because of the increase in the amount of moved content. However, our proposed scheme can prevent this increase in unnecessary information and can update the states of the BFs. As a result, in our proposed scheme, all nodes can cope with moved or deleted content. A large number of control packets may be necessary in order to control the two BFs. It is a trade-off relationship, but our method is nevertheless effective

Thsis paper is an extended version of the paper that appeared in the Asia-Pacific Network Operations and Management Symposium 2013.

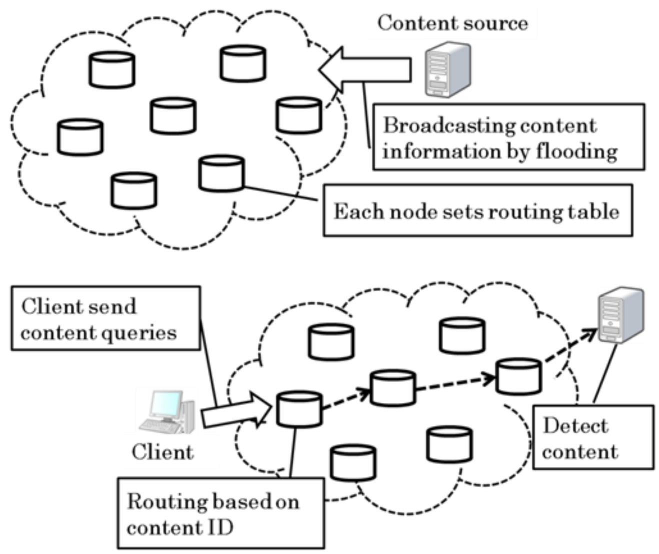

Fig. 1: Overview of content oriented networks

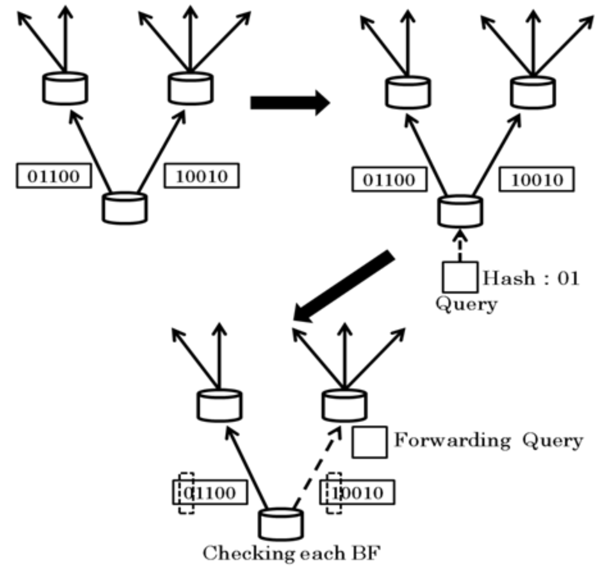

Fig. 2: Process of determining next hop in routing scheme using Bloom filters

in the event of a memory bottleneck. For example, it is effective when the size of the private network is small or in networks with relatively small memory devices such as smart phones.

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 our scheme's effectiveness. We conclude the paper in Section 5.

II. Related Works

A. Content oriented networks

A content oriented network has 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. The process of broadcasting content information to all nodes in the network through flooding is referred to as "advertising" in this paper. 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. Some architectures that focus on content detection are i3 (Internet Indirection Infrastructure) [9], DONA (Data-Oriented Network Architecture) [10], and TRIAD [11]. In particular, TRIAD is said to be the subject of a pioneering study. Content Centric Networking (CCN) [12], [13] involves an architecture that focuses on both content detection and provision. In CCN, each node has two tables called a forwarding information base (FIB) and a pending interest table (PIT). The FIB is a routing table that stores content information; the entries are content IDs. The PIT stores information indicating where the content query has passed; content provision is performed by using the PIT. To achieve a content oriented network, it is 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.

These tasks increase the amount of unnecessary information can cause queries to be forwarded on the wrong route. Queries are then repeatedly forwarded on the wrong route, and the time to live (TTL) of queries becomes zero. Hence, queries fail to detect the required content. Our proposed scheme is able to solve this problem.

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 element is fed to each of the hash functions to determine the array positions. The bits at these positions are set to 1. 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. One problem with BFs is that the positions in a BF may have by chance been set to 1 during the insertion of other elements. This results in a false positive and makes it impossible to remove an element from the BF because false negatives are not permitted.

In existing routing schemes using BFs in content oriented networks, the links of all nodes have a BF that stores content information that content sources broadcast by using flooding in

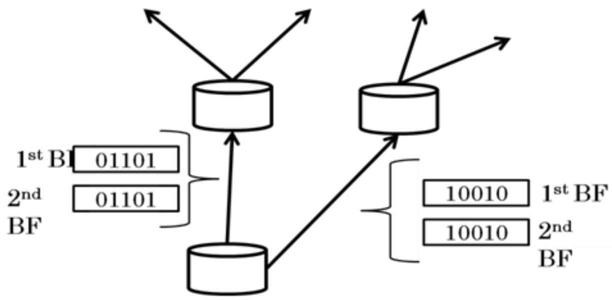

Fig. 3: Scheme diagram of node with two Bloom filters

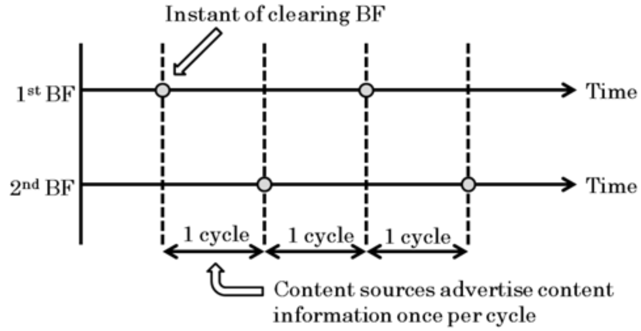

Fig. 4: Cycle of clearing BFs and advertising content information

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 checks all the BFs and determines that the BF of the link on the right is storing content information that has hash:01. That node then forwards the query to the right side.

However, routing schemes that use BFs in content oriented networks are problematic in that they cannot cope with content that is moved or deleted because removing an element from a BF is impossible. When content sources are mobile terminals, it is possible that some content will be moved content [21], [22].

III. PROPOSED METHOD

A. Overview of proposed scheme

This section describes a routing information management scheme that can cope with moved content by using two BFs in a content oriented network. The use of two BFs makes it possible to significantly reduce the amount of memory required in each node. However, it may be necessary to control a large number of packets in order to control the two BFs. This is somewhat of a trade-off. However, our method is effective in the case of a memory bottleneck. For example, as mentioned in section I, it is effective when the private network is small or when a network consists of relatively small memory devices such as smart phones.

In the proposed scheme, 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 of content information from content sources and the clearing of the BFs. This prevents unnecessary information from accumulating, which might cause queries to be forwarded on the wrong route. The scheme also updates the states of all BFs. The node checks its own BFs and determines the next hop when the query including the required content information arrives at the node. Thus, the query is forwarded repeatedly, and the required content is finally detected.

B. Routing information management capable of coping with moved content

First, we describe the architecture of the nodes in our proposed scheme. In addition to the routing using the existing BF, the links of all nodes have BFs that store content information that content sources broadcast by flooding instead of a normal routing table. However, in our proposed scheme, the links of all nodes have two BFs in order to clear them alternately.

Next, we explain the operation of nodes in our proposed scheme. Each node extracts content ID from content information and converts it to hash values when content information that is broadcast from the content source by flooding arrives at each node. After that, the content information is stored in the corresponding position of two BFs that link content information that has arrived. 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 the first BF. The first BF is then cleared at a fixed period after clearing the second BF. All nodes clear the first and second BFs, and all content sources broadcast their content information repeatedly.

Here, we describe how the node determines the next-hop node. When a content query arrives at the node, the node checks all the BFs it has. A specific link will have two BFs that store requested content information, and the node determines that link as the next-hop preferentially. For example, in Fig. 3, when the query that includes content information containing hash:01 arrives at the node, the node checks the BFs and infers that the first and second BFs of the right link are storing the requested content information (that has hash:01). Therefore, that node forwards the query to the right side. If it is not possible to determine the next hop using the above procedure, the node checks either the first or second BF and determines the next-hop node. Each node performs the above process repeatedly to detect the requested content.

Content sources perform the movement in the network and advertise content information by flooding. The movement is the changing location of the content source in the network. Routing information that forwards a query to the location where the content source existed before moving becomes unnecessary information. Thus, our proposed scheme repeats the advertising of content information from content sources and clearing of the BFs. As a result, unnecessary information is deleted from the network, and each node updates the state of the BFs. Basically, all content sources advertise their content information once in

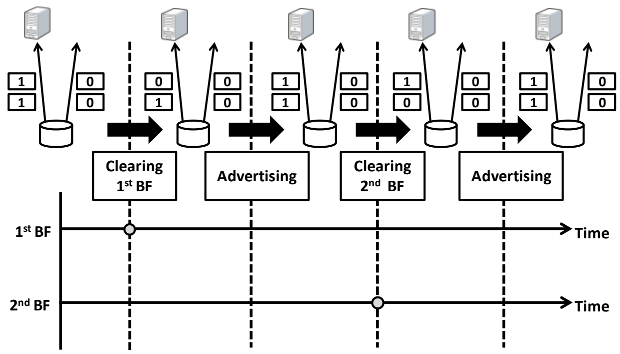

Fig. 5: Example when content source does not move

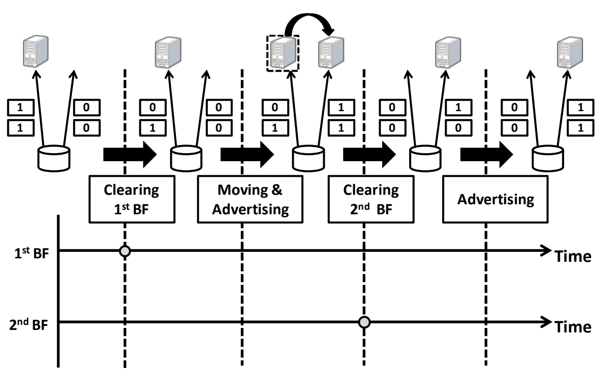

Fig. 6: Example when content source moves

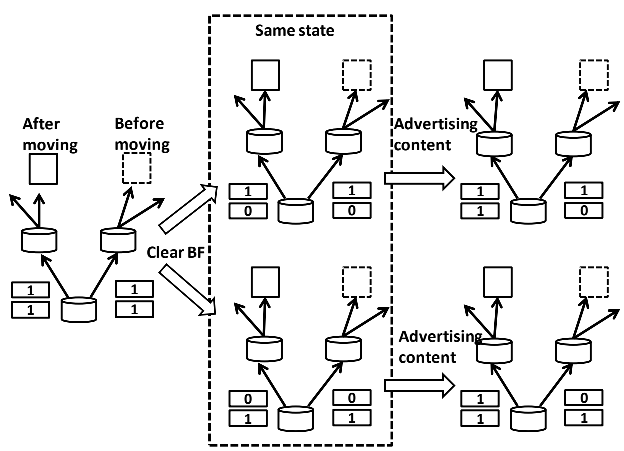

Fig. 7: The state in which nodes cannot determine the next-hop node in the proposed scheme

the period between clearing the first BF and clearing the second BF (Fig. 4). In addition, content sources advertise their content information after moving.

In the proposed scheme, routing information is compressed by using BFs. Moreover, each node can cope with moved or deleted content. However, problems can arise when the number of times that content information is advertised from content sources increases and when multiple BFs of the nodes have the same state.

C. *Example of the proposed scheme*

We show examples of the proposed scheme in Figs. 5 and 6. When the content source does not move, the node retains the content information in either of the two BFs by advertising content information after clearing the BFs (Fig. 5). Therefore, the latest content information is stored in the BFs, and the node is able to determine the direction where the required content exists. In Fig. 6, the same content information is stored in the BFs that plural links have temporarily after a content source has moved and advertised its content information. However, the states of the plural links are different. The link on the left has content information only in the second BF, but the link on the right has content information in both the first and second BFs. This difference enables the node to determine the direction where the required content exists.

The proposed scheme has fllowing drawback. In the proposed scheme, if content information is advertised twice in one cycle from different locations, the multiple BFs of the nodes would have the same state (Fig. 7). 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.

IV. SIMULATION

A. *Simulation model and conditions*

In this section, we describe the simulation model and the results of evaluating the proposed scheme. Table 1 lists the parameter values used in the simulation. Many simulations have been carried out and reported in previous studies, but not many simulations have been done using large scale networks. Therefore, we conducted 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 its BFs. Content sources were moved once to random locations during the simulation time. The content query was sent to the random location every unit time. The network topology was constructed in such a way that each node was placed in a random location and established links with other nodes in the order of the nearest to furthest distance. Each node had at least two links and a maximum of

Table 1: Parameter value in simulations

Item	Value
Simulation time [unit time]	500
Number of nodes	100, 1000
Network topology	Random allocation
Number of content items	50, 100, 300, 500
BF size [bits]	10, 30, 50, 70, 100
Length of advertising cycle [unit time]	10, 20, 30, 40, 50

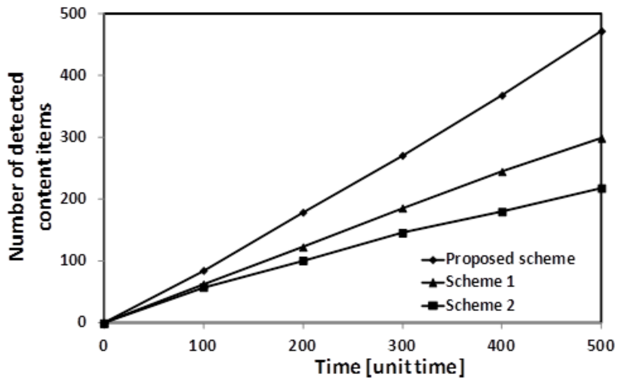

Fig. 8: Number of detected content items with 100 nodes

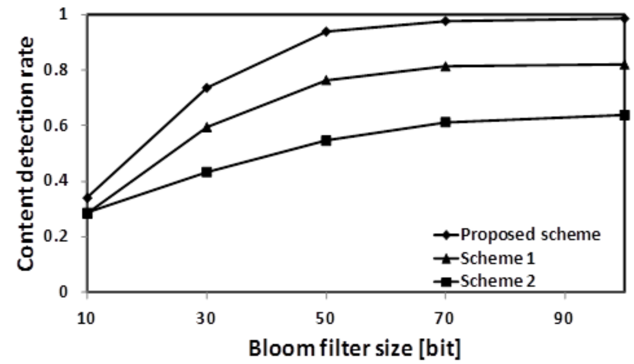

Fig. 10: Content detection rate when BF size was changed

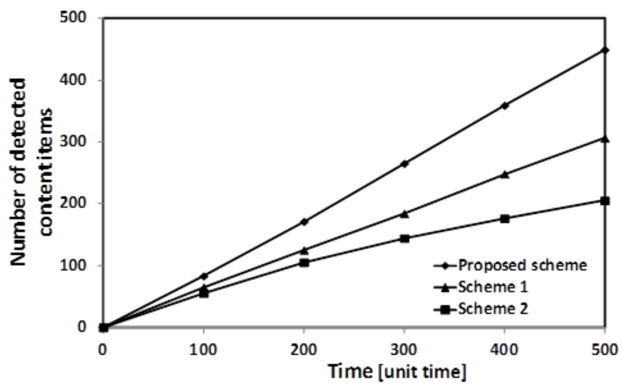

Fig. 9: Number of detected content items with 1000 nodes

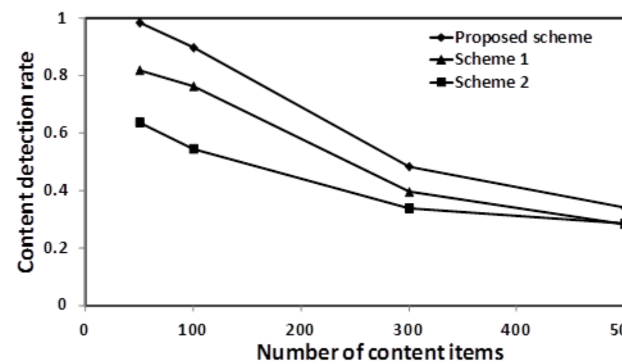

Fig. 11: Content detection rate when number of content items varied

five links. Content sources were placed in random locations and established one link with other nodes.

B. Simulation 1

We compare the proposed scheme with schemes 1 and 2 in this paper because we confirmed that more efficient searching was possible with the proposed scheme than with an existing routing scheme using BFs, even when the content was moved in the network. In scheme 1, each link in a node has 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 has one BF, but the nodes do not clear the BFs. Scheme 2 is a simple scheme using BFs in a content oriented network.

We plot the results of the simulation in Fig. 8 and Fig. 9. The results in Fig. 8 are for the case when there were 100 content items, the BF size was 50 bits, the length of the advertising cycle was 10 unit time, and there were 100 nodes. The details are the same for the results in Fig. 9, 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 from Fig. 8 and Fig. 9 that a more efficient search than in schemes 1 and 2 was possible even when content sources were moved. In scheme 1, the 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, not all nodes cleared their BFs, so the amount of unnecessary information continued to increase. Therefore, the number of content items that were detected was smaller than in the proposed scheme.

C. Simulation 2

We show the results of the simulation in Fig. 10. There were 100 nodes, 100 content items, the length of the advertising cycle was 10 unit time, and the BF sizes were 10, 20, 30, 40, or 50 bits.

We confirmed that the content detection rate decreased as the BF size decreased. The amount of data to be stored in the BF is 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 therefore set an adequate BF size. However, the BF must not be set too large because the BF consumes node memory.

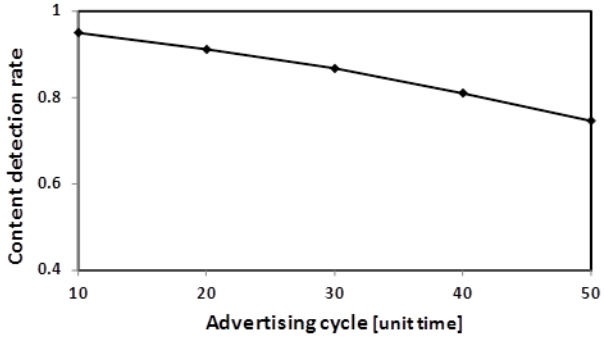

Fig. 12: Content detection rate when the length of advertising cycle was varied

D. Simulation 3

The simulation results are plotted in Fig. 11. There were 100 nodes, the BFs were 50 bits, the length of the advertising cycle was 10 unit time, and there were 50, 100, 300, or 500 content items.

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

E. Simulation 4

The results of this simulation are plotted in Fig. 12. There were 100 nodes, 100 content items, the BF size was 50 bits, and the length of the advertising cycle was 10, 20, 30, 40 or 50 unit time.

We confirmed that the content detection rate decreased as the advertising cycles became longer. Lengthening the advertising cycles led 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 was moved increased in the network. The links each node had 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 advertising content information more frequently increases the amount of network traffic and the network load. Accordingly, we must set an appropriate advertising cycle.

F. Simulation 5

In the proposed scheme, content queries are forwarded to the next hop node by checking BFs repeatedly, which leads to the detection of content items. However, content queries can be forwarded to the wrong node when plural BFs of nodes have the same state. False positives in BFs and advertising content information twice in one cycle from different locations results in plural BFs of nodes having the same state. Therefore, we checked whether false positives or moving content caused queries to be forwarded on the wrong route. The results of the simulation are shown in Fig. 13 and Fig. 14.

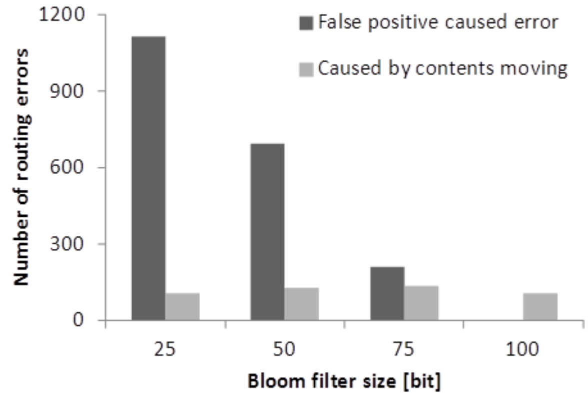

Fig. 13: Number of routing errors when the BF size was changed

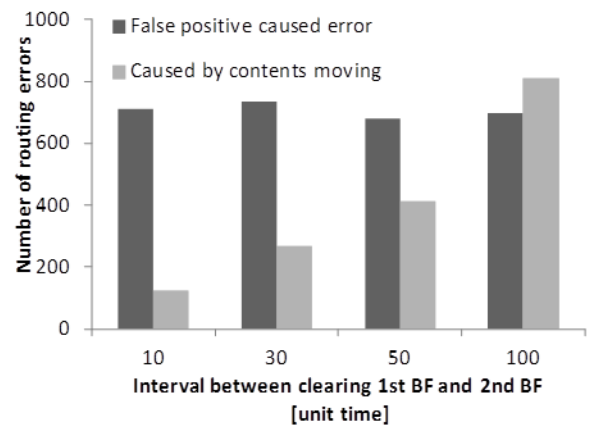

Fig. 14: Number of routing errors when the advertising cycle length was changed

We confirmed from Fig. 13 that the number of routing errors caused by false positives increased as the BF size decreased. The accuracy of a BF is dependent on the size. The amount of data to be stored in the BF is 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.

We also found that the number of routing errors caused by false positives increased as the length of the advertising cycle increased. The number of times the BFs were updated decreased as the interval between clearing the BFs became longer, as shown in Fig. 14. Consequently, the amount of unnecessary information increased in the network, and content queries were sometimes forwarded to the wrong node. Accordingly, it is necessary to set the appropriate parameters.

V. Conclusion

In this study, we proposed a routing information management method that can cope with content that is moved 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 was moved in the network. We also found that in the proposed scheme, it was necessary to appropriately set parameters in order to achieve 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 cannot 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, to confirm that the amount of memory usage is reduced by using BFs, it is necessary to compare memory usage between the proposed scheme using BFs with an adequate size and another scheme that does not use BFs.

References

[1] F. Douglis and M. Kaashoek, "Scalable Internet Services," *IEEE Internet 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 Magazine*, 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 Magazine*. pp. 123-127, Vol. 50. 2011.

[15] V. Dimitrov and V. Koptchev, "PRISP Project: Publish-Subscribe Internet 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 2010*, 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 Magazine*. pp. 36-43. 2012.

[20] P. K. Agyapong and M. Sirbu, "Economic Incentives in Information-Centric Networking: Implications for Protocol Design and Public Policy," *IEEE Communications Magazine*. 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 Magazine*. 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.