

## Working Set Model for a Shared Multimedia Content in Content Centric Networks

Seung-Seok Kang<sup>1</sup>, Longzhe Han<sup>2</sup> and Matt W. Mutka<sup>3</sup>

<sup>1</sup>*Department of Computer Science, Seoul Women's University,  
Seoul, Korea*

<sup>2</sup>*School of Information Engineering, Nanchang Institute of Technology,  
Nanchang 330099, China*

<sup>3</sup>*Department of Computer Science and Engineering, Michigan State University,  
East Lansing, Michigan, USA*

<sup>1</sup>*msukang@swu.ac.kr*, <sup>2</sup>*lzhan@nit.edu.cn*, <sup>3</sup>*mutka@cse.msu.edu*

### Abstract

*Content Centric Network (CCN) is proposed to facilitate future multicast services. The special attributes of the CCN are the named-data scheme, receiver-initiated transmission, in-network caching, and support of multicasting. The CCN node has three main components: Content Store (CS), Pending Interest Table (PIT), and Forwarding Information Base (FIB). Among the components, the content store caches some portion of content expecting to reuse the content in near future by other clients. This paper describes that when CCN nodes reserve a reasonable amount of the content store memory for the shared content of multiple users, the network provides stable transmission performance due to in-network caching and multicast transmission. On the contrary, if the CCN nodes do not reserve a proper amount of content store space for the shared content, it results in degrading the transmission performance. The simulation results indicate that it is better to employ a FIFO-based replacement algorithm for a shared content flow than to choose usage-based algorithms such as LRU and LFU.*

**Keywords:** *Content Centric Network, Content Store, Multimedia Transmission, In-Network Caching, Working Set Model*

### 1. Introduction

Content search and its distribution is one of the growing service areas in the Internet. Especially, some of the popular Internet service areas include real-time on air broadcast, video on demand, multimedia streaming, and video conferencing. According to the Cisco Visual Networking Index (VNI), 80% of the total Internet traffic comes from P2P and video streams in 2011. In addition, 34% of the Internet traffic is carried by the content delivery network (CDN) in 2012 and the portion will be expected to increase up to 51% in 2017. The Internet users more like to know what kinds of content provided by the Internet rather than where the content is stored.

Content Centric Network (CCN) is a new communication paradigm proposed by Palo Alto Research Center Incorporated (PARC) and shifts the traditional end-to-end perspective to a content-oriented model, thereby creating opportunities to facilitate video transmission in the future Internet [1, 2]. Existing Internet service uses a client-server model in which a client specifies the server address in order to receive content from the server. CDN is one of the popularly used service networks and is one of the interesting research topics [3]. CDN consists of a large distributed system of servers deployed in multiple data centers across the

Internet and serves content to end-users with high availability and high performance. However, CDN still keeps the end-to-end communication. In CCN, the receiver initiates data transmission by sending data request packet, called *interest* packet, which contains a content name specifying a uniform resource identifier (URI). Any CCN node including the content server that receives the interest packet and holds the matched data, it will transmit the corresponding *data* packet to the receiver.

The key features of CCN are its named-data scheme, in-network caching, receiver-driven transmission, and support of multicasting. Named data is the unified and essential element of CCN's architecture. In order to decouple the data from where it is stored, data is identified, disseminated, retrieved and routed completely based on name. Every CCN node is able to cache data in its local storage to satisfy upcoming data requests from other nodes. Each CCN node maintains three components: content store (CS) for caching data content, pending interest table (PIT) for storing interest packet information, and forwarding information base (FIB) for maintaining routing information towards content source(s).

A CCN node stores data into the content store based on the expectation that some clients will request the cached data in near future. If many users share the real time video streaming or video conferencing stream, caching the shared content by the intermediate CCN nodes help improving the transmission performance. Usually, when the content store is full and a new data element has to be inserted, a replacement algorithm selects a victim to replace with a new data. However, if each client starts requesting the shared content at slightly different time or there is a network delay between clients, insufficient content store memory space for the shared flow results in degrading the transmission performance. By reserving a proper amount of memory space for a shared flow in the content store, the cached content will be delivered efficiently. This is similar for the operating system to allocate a proper working set size of memory to each process. If the working set size is not large enough, it causes an unexpected large number of page faults. This paper introduces a necessity of applying the working set model to manage the content store for a shared content flow in order to improve transmission performance.

The remainder of this paper is organized as follows. Section 2 describes related work. Section 3 explains the details of CCN and the working set model in the content store. Section 4 illustrates the simulation and analyzes the experimental results. Finally, section 5 concludes this paper.

## 2. Related Work

Working set model [4] was introduced by Peter Denning who wrote his dissertation about the locality of the main memory usage pattern. According to the model, each process accesses only some part of the main memory in some specific time interval. In a paging system, if an operating system allocates smaller number of pages to a process than the working set size, the process suffers from frequent page faults and experiences the degradation of the execution performance. The solution of the problem is to give a little larger number of pages than the working set size.

Paging algorithm [5, 6] employed the working set model to assign a proper number of pages to each process. The model was applied for a file server to prefetch files based on each application's file working set detected by the file usage pattern [7]. P2P file backup system [8] was built based on the file working set. When a thread is migrated and restarted, the working set size of cache memory is prefetched in order to improve execution performance [9].

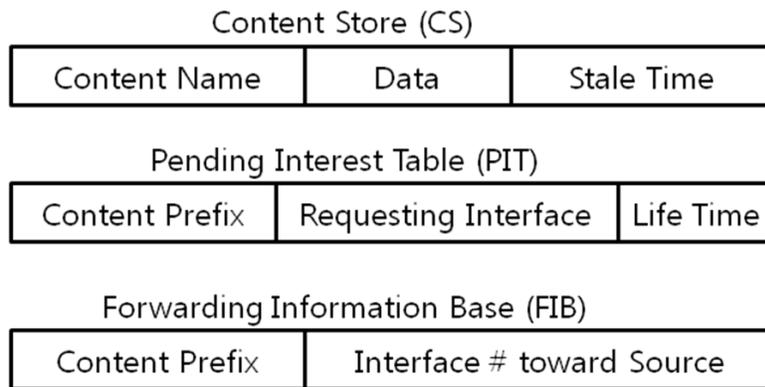
In-network caching allows intermediate CCN nodes to store popular content in the content store, and to share the content with other clients to enhance the content delivery performance

over the CCN. CodingCache [10] employs network coding and random forwarding to deliver content over diverse routes, which improves the network caching performance. When the CCN nodes give priority to each data in the content store while performing caching and forwarding content, it improves the cache hit ratio [11]. In [12], content is not cached in all CCN nodes of a route from the source to the requesting clients, but in some specific nodes in which there is a high probability of cache hits in order to improve the transmission performance as well as the storage efficiency of the content store.

### 3. CCN and Working Set Model in CS

#### 3.1. Content Centric Network Architecture

In CCN, there are two types of packets used: interest packet and data packet. User content is divided into chunks, and CCN uses a *data packet* to encapsulate each chunk. When a client wants to receive the content chunk, it sends an *interest packet*. Because the CCN architecture is based on named data, both the interest packet and the data packet are associated with a hierarchical unique name to identify a specific chunk. The naming scheme follows the URI structure.



**Figure 1. Three Main Components in CCN Node**

The CCN node has three main components: content store, PIT and FIB, as shown in Figure 1. Data packets are cached in the content store. The FIB contains the route information for forwarding interest packets, and the PIT traces forwarded interest packets. In order to send a relevant data packet to the requesting client, the incoming interface of the interest packet is registered in the PIT.

The primary task of the CCN node is to process both the interest packets and the data packets. Upon receiving an interest packet, the CCN node uses its content name to perform a longest-match search in the content store, PIT, and FIB. If the matching data is found in the content store, the CCN node sends it to the requesting client. Otherwise, if an exact-match entry is found in the PIT, the incoming interface of the interest packet is added to the requesting interface list of the PIT entry, and the interest packet is discarded. This is because an identically named interest packet has already been forwarded, so it is not necessary to resend the interest packet. If a PIT entry is not found, the CCN node will forward the interest packet to the data source through the FIB match. If there are no data in the FIB that match the interest packet, the interest packet is discarded.

In addition, when receiving a data packet, the CCN node will perform a longest-match search in the content store, PIT, and FIB. If a PIT match is found, the data packet will be sent to every interface on the requesting interface list. In addition, if no match is found in the content store, the data packet can be added to the content store. However, if a PIT match is not found, then the data packet must be discarded because the data packet is unsolicited. The CCN protocols take into account that underlying networks might have unreliable transmission services. In CCN, the retransmission approach is adopted to recover interest or data packet losses. Each PIT entry has a timer. The timeout value is indicated by the InterestLifetime field of the interest packet header. The application layer protocols usually assign the value of the InterestLifetime. When the timeout occurs at intermediate node, the corresponding entry is removed from the PIT. Otherwise, if the timeout occurs at the data client, the interest packet will be retransmitted.

### 3.2. Content Store Replacement Algorithm and Working Set

In CCN, a large file or a multimedia streaming data are divided into chunks and put a unique name. A client sends an interest packet to CCN in order to request the content chunk. When any intermediate CCN node including the content source has its corresponding content, it immediately packs the content chunk into a data packet and transmits to the client. If an intermediate CCN node receives the data packet and does not have the content, it keeps the content in the content store for a while. However, if the content store is already full and a new data has to be inserted, one of the elements in the content store will be removed. The replacement algorithm will select a victim in the content store to replace with a new content. There are many replacement algorithms introduced and this paper uses first in first out (FIFO), least recently used (LRU), least frequently used (LFU), and random replacement algorithms.

The most important feature of CCN is to retrieve content by its name rather than a server address. Other features are in-network caching and the support of multicast transmission. These two features play a crucial role in terms of the transmission performance of the CCN. In-network caching obviates the delivery of an interest packet to the content source when an intermediate CCN node caches the content and replies it immediately to the requesting client. If several clients want to play multimedia content such as Internet on-air broadcast or video streaming, a CCN node on their common route to the clients may multicast the content, which results in reducing the network traffic.

Suppose a content store in a CCN network can cache four content chunks, and three clients request the same content. Further assume that the content name increases sequentially from 1, and currently the content store caches content 1, 2, 3, and 4. In addition, because of the network delay and initiation time gap by the clients, they start at slightly different time. Each client requests the content as follows: client C1 requests content 5, client C2 asks content 3, and client C3 wants content 1. When content 5 arrives at the CCN node on the common route of the three clients, the node has to replace one of the existing content chunks with content 5. In this case, whichever content is selected as a victim, the content has to be requested again in near future and the delivery performance degrades. One solution is to increase the size of the content by one. Then there is a high probability that fetching a content chunk once will be enough to deliver it to the three clients. As a result, it could be beneficial when the content store allocates enough memory to cover actively accessing content chunks of a content flow, called a dynamic *working set*, in order to efficiently distribute the content.

In terms of the replacement algorithm, selecting a victim also affects the content delivery performance. For example, suppose the content store size is five and content 1, 2, 3, 4, and 5 are stored. Further assume that each client requests the content chunk as follows: C1 requests content 6, C2 asks content 4, and C3 needs content 2. When the content 6 is arrived to the

CCN node, one of the five content chunks has to be selected as a victim. In case of the LRU replacement algorithm, content 2 or 4 is selected as a victim because content 1, 3, and 5 are recently used. When the LFU is adopted, content 4 or 5 is selected as a victim. If the FIFO algorithm is applied, content 1 is chosen. Without introducing a new client, content 1 is highly unlikely to be used in near future. This example implies that when a flow of content is accessed sequentially, the FIFO replacement algorithm is suitable for managing the content store.

#### 4. Simulation Environment and Results

This section describes the simulation environment and the results about the necessity of the working set model in the content store. For the simulation, authors use ndnSim [13] over ns-3 [14] network simulator. The simulation network topology includes five clients, three CCN routing nodes and one content server. The topology is depicted in Figure 2. All clients are connected with 10 Mbps links and 5 msec delay. The links between routing nodes are 100 Mbps with 5 msec delay except that 10 Mbps with 100 msec link is in between the node R3 and the content server S.

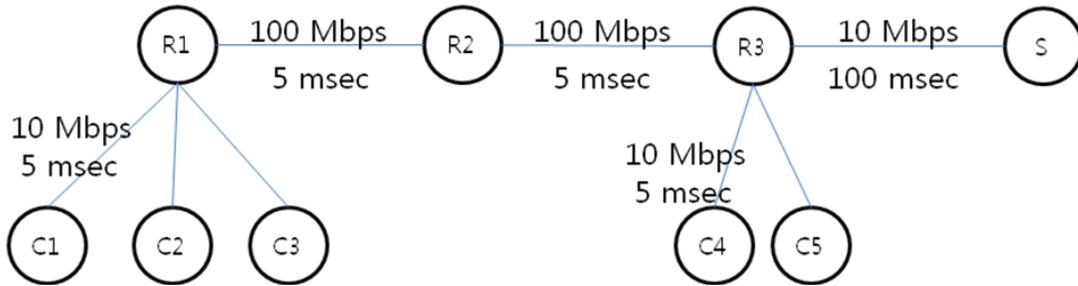
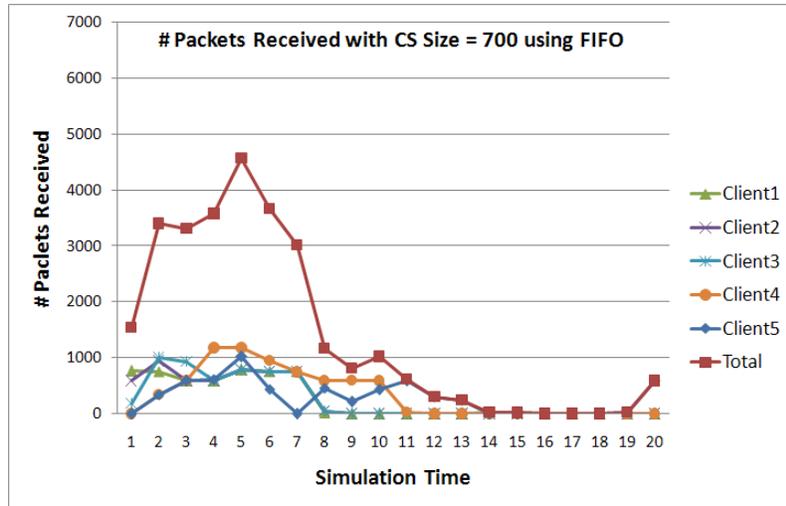


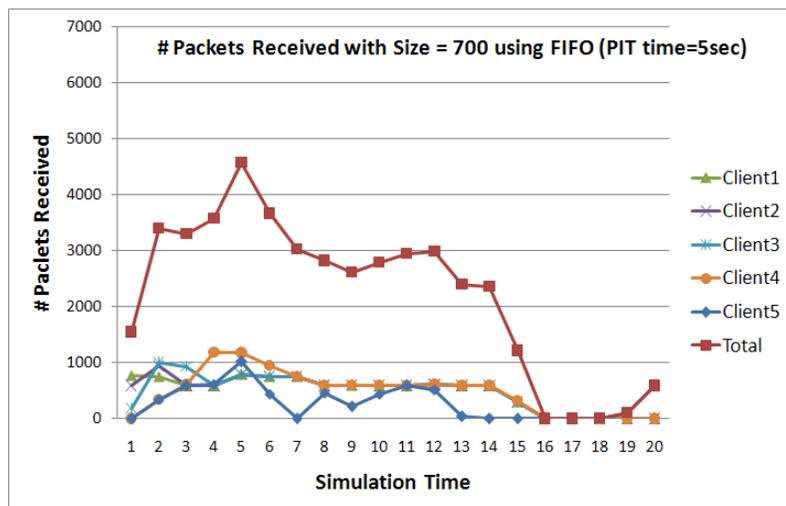
Figure 2. Network Topology used in this Simulation

The five clients download the same content from the content server S. They send 1000 interest packets per second to their directly connected routing nodes. The data packet size is 1024 bytes. Each client starts transmitting the interest packets with different intervals. For example, C1 starts to send the first interest packet at simulation time 0 second, C2 at 0.4 second, C3 at 0.8 second, C4 at 1.2 second, and C5 at 1.6 second. The size of the CS varies from 700 data entries to 1400 entries. The CS replacement algorithms used in this simulation are LRU, LFU, FIFO, and Random.

Figure 3 displays the number of received data packets by the five individual clients within 20 seconds. The content store can save 700 entries for the content flow and FIFO replacement algorithm is used when the content store is full. The label “Total” in the legend means the sum of the received data packets by all the participating clients per second. Initially, all clients send 1000 interest packets per second and expect to receive that amount of data packets. Most clients receive some number of packets, but it does not meet the expected number of data packets. Client 4 and 5 temporarily receive more than 1000 data packets. However, all clients did not get any data packet at time 14. It is because the PIT entry stores the route backward to the client requesting the data packet, and the loss of the PIT entry caused by the timeout (set to 1 second in ndnSim) results in dropping the data packets in CCN nodes.

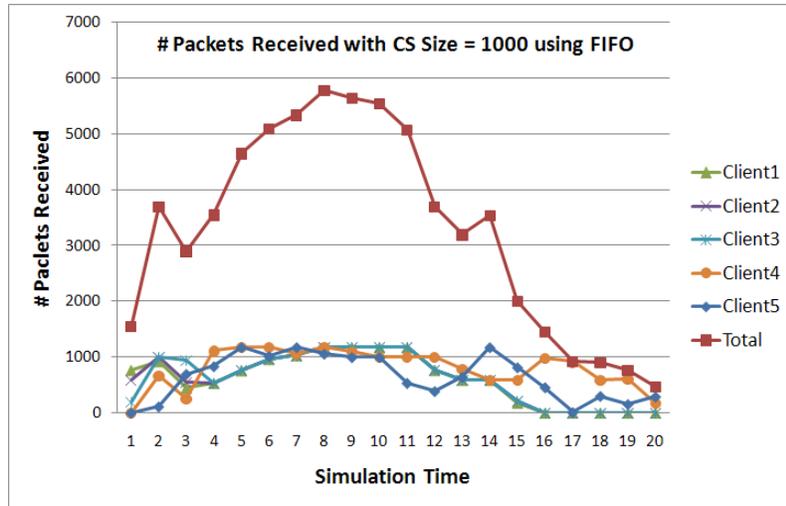


**Figure 3. # of received Packets when CS Size is 700 with FIFO Algorithm**



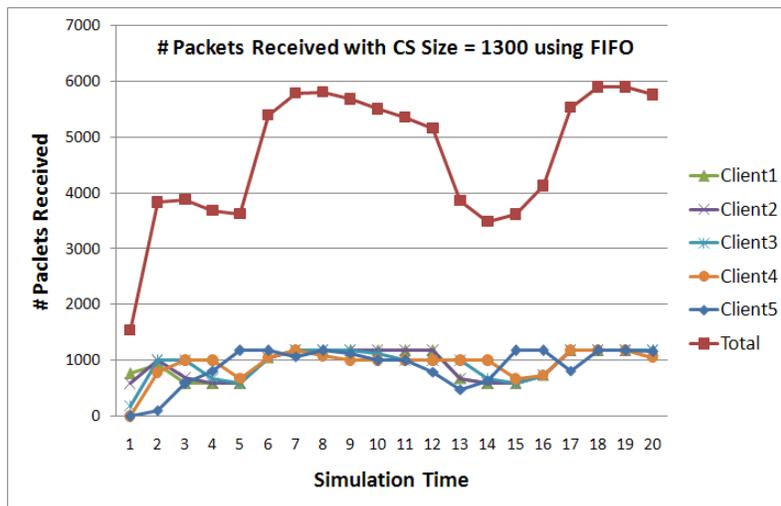
**Figure 4. # of received Packets when CS Size is 700 and PIT Expire Time is 5 sec**

Figure 4 shows the result of using the similar condition to the previous simulation except that the timeout value is extended to 5 second rather than 1 second used as a default value in ndnSim. That is, when a client transmits an interest packet, the data request information including the backward route to the client is stored in the PIT entry up to 5 seconds. When the timeout expires, the PIT entry is deleted and the requested data cannot be forwarded to the client. Due to the extended timeout, clients receive more packets and longer period of time than before, but, eventually at time 16, all clients did not receive any more packets due to the loss of the backward route information to the clients. For the remaining simulations, the timeout value is set to 1 second.



**Figure 5. # of received Packets when CS Size is 1000 with FIFO Algorithm**

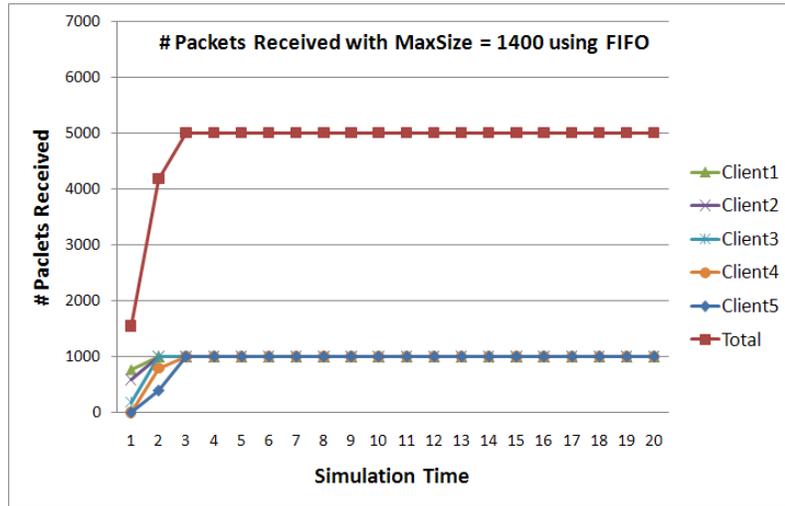
Figure 5 illustrates the results of the data packets received by the five clients and the total number of the data packets in each simulation second when the size of the content store for the flow becomes 1000 with the FIFO replacement algorithm. As the content store size increases for the flow, so does the number of packets received by the clients. On the contrary, as the simulation time goes by, the total number of received packets dropped due to the loss of the backward route information in the PIT.



**Figure 6. # of received Packets when CS Size is 1300 with FIFO Algorithm**

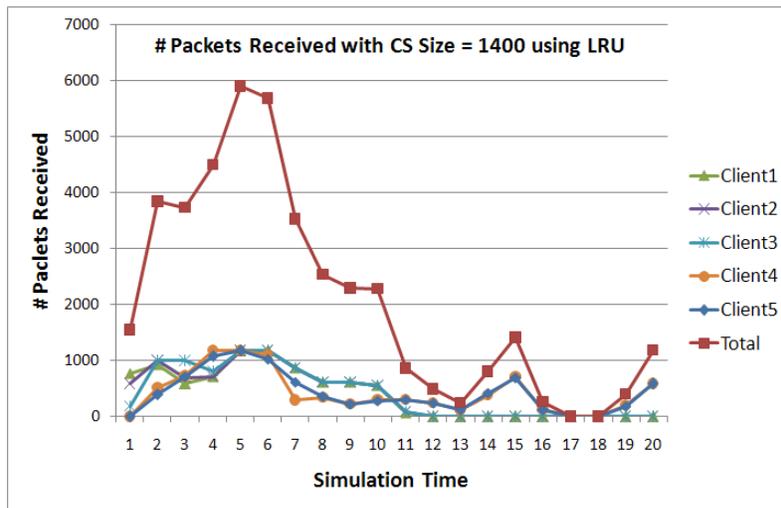
When the content store size becomes 1300 for the content flow, the data packet reception status by the five clients is depicted in Figure 6. The clients receive more data packets than when the content store size was 1000. As the graph indicates, the average of the total received packets from 2 to 20 second is about 4823. However, the standard deviation of the total value is 1191. This result implies that there is a high fluctuation of the data reception, and it causes a long delay and a delay jitter for a multimedia traffic. Therefore, it is not good for enjoying

multimedia content because it experiences a lack of decodable data packets from time to time and the belated data packets may be useless to decode.



**Figure 7. # of received Packets when CS Size is 1400 with FIFO Algorithm**

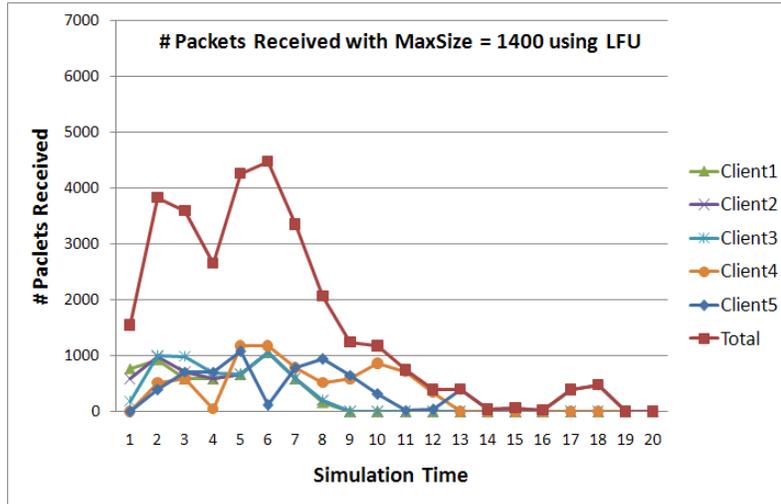
Figure 7 represents the data reception results by the five clients when the content store size becomes 1400. All clients transmit 1000 interest packets per second, and they receive the requested amount of data packets after 3 second. As a result, each client receives 1000 data packets per second and a total of 5000 packets in each second. This result does not imply that the content server S transmits 5000 data packets to all clients per second, but indicates that the data packets requested by the client C1 are cached in the CCN routing nodes R1 and R3 which distribute the packets to their connected clients requesting the same content.



**Figure 8. # of received Packets when CS Size is 1400 with LRU Algorithm**

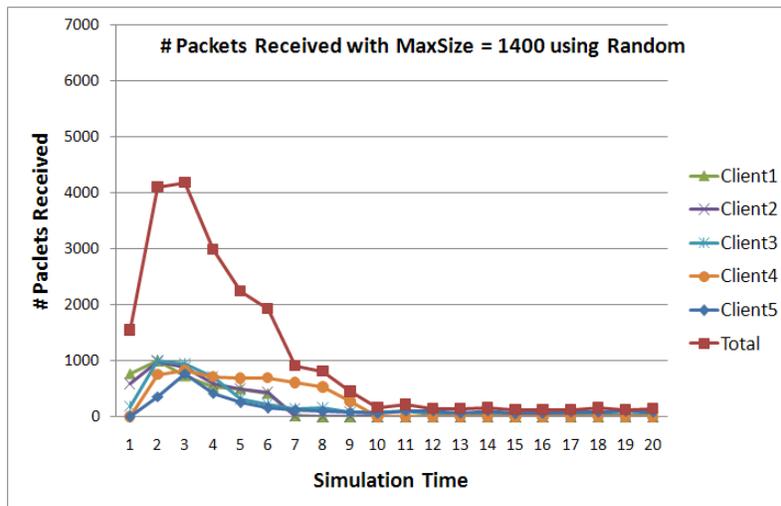
Figure 8 presents the results when the least recently used (LRU) algorithm is used for the content store replacement, and the content store size is the same as 1400 elements for the content flow. The total number of the received packets decreases from simulation time at 10

second. As described in section 3, the least-recently-used element might be reused in near future, and LRU may not be an efficient replacement algorithm. If an individual element in the content store is popular to several clients, the element has a high probability to reuse in near future, and LRU replacement algorithm may be a good selection. However, If a group of content chunks are involved in a working set of a content flow and it has to select one victim in the working set of elements, FIFO replacement algorithm works better than that of the LRU algorithm.



**Figure 9. # of received Packets when CS Size is 1400 with LFU Algorithm**

Figure 9 shows the simulation result when the least frequently used (LFU) replacement algorithm is applied to select a victim when the content store is full. LFU algorithm works worse than that of LRU as the total number of received data packets decreased sharply from the simulation time at 8 second. This is because the recently added element has less frequently accessed by the clients and is selected as a victim even though it may be used in near future.



**Figure 10. # of received Packets when CS Size is 1400 with Random Algorithm**

Figure 10 illustrates the data packet reception results by the five clients with the random replacement algorithm and the content store size of 1400. This algorithm shows the worst performance among the other replacement algorithms. The total number of received data packets drops from the simulation time at 7 second.

## 5. Conclusion

This paper presents that, when a CCN network delivers popularly-shared content such as real time broadcast or video conferencing, the content store in CCN nodes prefer FIFO replacement algorithm for a shared content flow rather than others. CCN nodes require to analyze a requesting pattern of the interest packets and to decide a working set size in the content store for a shared content with several clients, which lead to improve the transmission performance as well as to reduce the number of packets delivered over the network. This paper shows the necessity of the working set for the shared content and the size of the working set in the content store will be dynamically decided based on the requested data pattern and the network condition. The replacement algorithm in a working set for a content flow prefers FIFO than others.

## Acknowledgements

This work was supported by a research grant from Seoul Women's University (2014).

## References

- [1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking Named Content", Proceedings of ACM CoNEXT, (2009) December 1-4, Rome, Italy, pp. 341-357
- [2] V. Jacobson, D. K. Smetters, N. H. Briggs, M. F. Plass, P. Stewart, J. D. Thornton, and R. L. Braynard, "VoCCN: Voice over Content-Centric Networks", Proceedings of ACM Workshop on Re-architecting the Internet (2009) December 1-4, Rome, Italy, pp. 1-6.
- [3] H. Jiang, J. Li, Z. Li, and X. Bai, "Efficient Large-scale Content Distribution with Combination of CDN and P2P Networks", International Journal of Hybrid Information Technology, vol. 2, no. 2, (2009), pp. 13-24
- [4] P. Denning, "The Working Set Model for Program Behavior. Communications of the ACM, vol. 11, no. 5, (1968), pp. 323-333
- [5] A. Smith, "A Modified Working Set Paging Algorithm", IEEE Transactions on Computers, vol. C-25, no. 9, (1976), pp. 907-914
- [6] C. Wang, "The Research on the Dynamic Paging Algorithm based on Working Set", Proceedings of International Conference on Future Information Technology and Management Engineering (2009), December 13-14, Sanya, China, pp. 396-399
- [7] C. Tait and D. Duchamp, "Detection and Exploitation of File Working Set. International Conference on Distributed Computing Systems", (1991), May 20-24, Arlington, Texas, pp. 2-9.
- [8] E. Colaco, M. Oliveira, A. Soares, F. Brasileiro, and A. Guerrero, "Using a File Working Set Model to Speed Up the Recovery of Peer-to-Peer Backup System", SIGOPS Operating Systems Review, vol. 42, no. 6, (2008), pp. 64-70.
- [9] J. Brown, L. Porter, and D. Tullsen, "Fast Thread Migration via Cache Working Set Prediction", IEEE High Performance Computer Architecture (2011), February 12-16, San Antonio, Texas, pp. 193-204.
- [10] Q. Wu, Z. Li, and G. Xie, "CodingCache: Multipath-aware CCN Cache with Network Coding", Proceedings of SIGCOMM Workshop - Information Centric Networking (2013) August 12, Hong Kong, China, pp. 41-42.
- [11] S. Guo, H. Xie, and G. Shi, "Collaborative Forwarding and Caching in Content Centric Networks", Proceedings of International Federation for Information Processing (2012), May 21-25, Prague, Czech Republic, pp. 41-55
- [12] W. Chai, D. He, I. Psaras, and G. Pavlou, "Cache Less for More in Information Centric Networks", Proceedings of International Federation for Information Processing (2012), May 21-25, Prague, Czech Republic, pp. 27-40.
- [13] NS-3 based Named Data Network (NDN) simulator. <http://ndnsim.net>
- [14] The network simulator 3, <http://www.nsnam.org>

## Authors



**Seung-Seok Kang**, He received his B.Sc. degree in Computer Science from the Korea University in 1992, and his M.S. degree and Ph.D. in Computer Science and Engineering from Michigan State University, East Lansing, Michigan, in 1998 and 2004, respectively where he worked as a research and teaching assistant. Currently, he is with the Seoul Women's University in Seoul, Korea (ROK) as an associate professor. His research interests include the area of ad hoc networks, anonymous communications, content sharing, multimedia communications, QoS, P2P networks, mobile computing and CCN.



**Longzhe Han**, He received his Ph.D. degree in the Dept. of Computer Science at Korea University, in 2013. Currently, he is with the school of Information Engineering at Nanchang Institute of Technology. His research interests include cognitive radio networks, future Internet, network security, multimedia communications, machine-to-machine communication and heterogeneous network in 5G.



**Matt W. Mutka**, He received the B.S. degree from the University of Missouri-Rolla, Rolla, MO, USA, the M.S. degree from Stanford University, Stanford, CA, USA, both in electrical engineering, and the Ph.D. degree in computer sciences from the University of Wisconsin-Madison, Madison, WI, USA. He is on the faculty of the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI, USA, where he is currently a Professor and Chairperson. He has been a Visiting Scholar with the University of Helsinki, Helsinki, Finland, and a Member of technical staff with Bell Laboratories, Denver, CO, USA. His current research interests include mobile computing, sensor networking, wireless networking, and multimedia networking.

