

In-Network Caching for Simultaneous Download from Multiple Receivers in Content-Centric Networking

TAKAHIKO KATO¹ KYOUHEI SASANO¹ MASAKI BANDAI² MIKI YAMAMOTO³

Received: May 22, 2016, Accepted: November 1, 2016

Abstract: This paper first investigates how a network operates when multiple receivers download content simultaneously in content-centric networking (CCN) when the receivers' downloading speeds differ. The results indicate that the performance of the download completion time of a faster user degrades excessively due to a decrease in the cache-hit rate in the router. Based on the investigation, this paper proposes a novel in-network caching method for simultaneous download from multiple receivers in CCNs. The proposed method keeps cached data packets in a router until slower receivers download the data, in order to prevent slower users from directly downloading data from the content provider. We conduct computer simulations and confirm the effectiveness of the proposed method. We show that the proposed method can improve the download completion time performance in the situation where multiple receivers download content at different speeds in CCN.

Keywords: content-centric networking (CCN), caching, simultaneous download

1. Introduction

In recent years, the ratio of content traffic, such as video streaming, to total network traffic has been increasing. Based on research by Cisco [1], the amount of content traffic will be 62 percent of the global network traffic by 2019. In traditional TCP/IP, the layer structure is clearly separated, and the role of TCP is to control congestion at end hosts. Therefore, a user requesting content must obtain the IP address of the content provider. Thus, the traditional TCP/IP is location-oriented. However, since downloading content is essentially content-oriented, it is more important for users to obtain the content itself than its location. There is a gap between the user's content-oriented requirement and the actual network architecture providing the location-oriented model. Content-centric networking (CCN) [2] is one method of addressing this gap.

A communication model of CCN is based on the content name in URL-like presentation, instead of IP address. In CCN, a content is divided into multiple data packets. An end user (receiver, hereinafter) initiates a data request packet (Interest, hereinafter) to download a data packet. Each Interest is a request for a data packet. A content router (router, hereinafter) forwards an Interest according to its name-based routing table. As a response to the Interest, a data packet is transmitted to the receiver. CCN has two special features: Interest aggregation and data cache. In Interest aggregation, a router aggregates multiple Interests requesting for the same data from multiple receivers in order to reduce net-

work traffic. If a router receives multiple Interests from different receivers, the router aggregates them into one Interest, which is forwarded upstream. When receiving a data packet requested by an aggregated Interest, the router copies the data packet and transmits the copies downstream, as in multicast communications. On the other hand, in data cache, a router has a buffer memory to cache data packets. When receiving a data packet, a router determines whether the data packet should be cached. If the data packet should be cached, the router caches the data packet. At that time, if there is no space to cache the data packet, the router deletes one of the cached data packets so that the new data packet can be cached. Unlike the traditional TCP/IP, in CCN, a user is allowed to download a content from both the data provider and routers, which cache some data packets.

In a real network environment, it is natural for multiple receivers to download popular content simultaneously. In such a situation, the special features of CCN, i.e., Interest aggregation and downloading cached data from routers, are expected to work well to reduce network traffic and speed up downloading. However, we do not have sufficient findings about how the network works in this situation. Reference [3] presented a cache policy that takes into account the downloading of multiple receivers in CCN. This method assumes a situation where multiple routers exist between the content provider (the server) and receiver. Routers probabilistically cache a data packet in order to decrease downloading from a server. However, this method does not consider the case where the downloading speeds of multiple receivers are different.

In this paper, we first investigate how a network works when multiple receivers simultaneously download content in CCN when the receivers' downloading speeds differ. In particular, we focus on the situation where the receivers' downloading speeds

¹ Graduate School of Science and Technology, Sophia University, Chiyoda, Tokyo 102-0071, Japan

² Department of Information and Communication Science, Sophia University, Chiyoda, Tokyo 102-0071, Japan

³ Faculty of Engineering Science, Kansai University, Suita-shi, Osaka 564-8680, Japan

are different due to the difference in the bandwidth of the bottleneck link. The results show that the performance of the download completion time of a faster user degrades excessively due to a decrease in the cache-hit rate in the router. Based on the investigation, we propose a novel in-network caching for simultaneous download from multiple receivers in a CCN. The proposed method keeps cached data packets in a router until slower receivers download the data, in order to prevent a slower user from directly downloading data from the content provider. We conduct computer simulations, and confirm the effectiveness of the proposed method. We show that the proposed method can improve the download completion time performance in the situation in which multiple receivers download content at different speeds in CCN. The basic concept of this paper is presented in Ref. [4]. This paper extends the proposed method to the case the number of simultaneous downloads exceeds two.

The rest of this paper is organized as follows. Related work is presented in Section 2. In Section 3, we describe the system design and conduct a fundamental evaluation. The proposed method is presented in Section 4 and is evaluated via computer simulations in Section 5. Section 6 concludes this paper.

2. Related Work

2.1 Content-centric Networking (CCN)

The CCN communication model [2] uses two types of packets: Interest packets and data packets. Content such as video files is divided into multiple data packets. In order to download content, a receiver generates and transmits Interests. A router forwards the Interest according to its routing table. As a response to an Interest, a data packet is transmitted back to the receiver. The communication model of CCN is based on the content name in a URL-like presentation, instead of as IP address. For example, content named “ComponentA/Component1/foo.mpg” is divided into hundred data packets. In this case, the first data packet is named “ComponentA/Component1/foo.mpg/seq1,” the second data packet is named “ComponentA/Component1/foo.mpg/seq2,” and so on. In this paper, we refer to “ComponentA/Component1/foo.mpg” as the name prefix and the data packet, i.e., “ComponentA/Component1/foo.mpg/seq1,” as the data name.

A router has three main control components: content store (CS), pending interest table (PIT), and forwarding information base (FIB). The CS is a buffer memory to cache (store) data packets. When a data packet arrives at a router, the router replaces a cached data packet in the CS with the newly arrived data packet according to its cache policy. The use of a cache in a router reduces network traffic because a data packet is not downloaded from the content provider (server), which is usually farther away than a router. The PIT keeps track of a forwarded Interest as a PIT entry so that the router transmits a data packet back to the receiver. A PIT entry is deleted when the corresponding data packet is transmitted or the PIT timer expires. The FIB is a routing table used for Interest transmissions.

When receiving an Interest, a router processes the Interest as the follows:

- The router checks its CS to determine whether the corre-

sponding data packet is cached. If the router has already cached the data packet in the CS, the copy of the data packet is transmitted to the interface at which the Interest is arrived.

- Otherwise, the router checks the PIT entry to determine whether the Interest having the same content name has already been forwarded. If the router already has the PIT entry, the router adds the interface at which the Interest is arrived and discards the Interest.
- Otherwise, the router creates a new PIT entry and forwards the Interest according to the FIB.

2.2 Congestion Control in CCN

The congestion control mechanism in CCN is classified into two methods: hop-by-hop methods and end-to-end methods.

In a hop-by-hop method, each router controls the Interest forwarding rate and regulates it if necessary. Hop-by-hop interest shaping (HoBHIS) [5], [6] and hop-by-hop and receiver-driven interest control protocol (HR-ICP) [7] are classified as hop-by-hop methods. In HoBHIS, each router observes the occupancy of its sending buffer. The router controls the Interest forwarding rate so that the buffer occupancy converges to a pre-defined threshold. In the HR-ICP, routers have some per-content-flow counters and buffers. Each router observes the balance of Interest and data flows using these counters. When the data packet transmission rate of a content flow is extremely high, the router decreases the Interest forwarding rate of the content flow for congestion avoidance.

On the other hand, in an end-to-end method, a receiver (end host) controls the Interest forwarding rate. The Interest control protocol (ICP) [8], the information centric transport protocol (ICTP) [9], content centric TCP (CCTCP) [10], and remote adaptive active queue management (RAAQM) [11] are classified as end-to-end methods. In these methods, a receiver controls the Interest forwarding rate using a window-size based additive increase multiplicative decrease (AIMD) mechanism. This paper focuses on the end-to-end method because of the simplicity of router architectures.

There are two ways to detect congestions in CCN: timer-based mechanisms and notification-based mechanisms. In a timer-based mechanism, the round trip time (RTT) between an Interest transmission and its corresponding data packet arrival is used for congestion detection. The ICP, ICTP, and CCTCP are classified as timer-based mechanisms. The second way is a notification-based mechanism. The method presented in Ref. [12] detects congestions by using a leaky bucket in a router. When detecting a congestion, a router generates a notification packet called a negative acknowledgement (NACK) packet and transmits it to the receiver. Unlike TCP/IP, in CCN, a receiver may receive data packets from multiple content providers, such as servers and routers on the way to the server (cached data packet in the CS). In the timer-based method, since multiple content providers have different RTTs, setting an appropriate timeout value is difficult. Therefore, this paper focuses on a notification-based mechanism.

2.3 Cache Management in CCN

Another important research topic in CCN is cache manage-

ment [13]. The purpose of the cache management is to determine whether a passing data packet is to be cached at each router. Some cache management methods such as probabilistic approach, popularity approach, and congestion aware approach, were presented [13]. Among them, our paper focuses on the congestion aware approach. The methods Refs. [14] and [15] are the promising methods in the congestion aware approach. In these methods, each router caches the high congestion-cost content in order to alleviate congestion. The method Ref. [14] considers the situation multiple content flows share one bottleneck link. The method Ref. [14] can effectively reduce the effect of congestion when more than one bottleneck links exist in the network. These methods mainly focus on how to manage cache among multiple contents. In that case, a router determines whether a passing data packet is to be cached based on the name prefix, instead of the data name. On the other hand, our paper focuses on the cache management when multiple users download the same content simultaneously. In other words, a router determines whether a passing data packet is to be cached based on the data name. Therefore, the methods Refs. [14] and [15] and our proposed method do not conflict each other.

3. Fundamental Evaluation

3.1 Motivation

In a real network environment, it is natural for multiple receivers to simultaneously download popular content. Reference [3] presented a cache policy considering downloads of multiple receivers. This method assumes multiple routers exist between a server and the receivers. The routers probabilistically cache a data packet to decrease the downloading from the server. However, this method does not consider the situation where the download speeds of multiple receivers are different. In this paper, we investigate in detail how a network operates in such a situation.

3.2 System Model

We consider an end-to-end congestion control method. The congestion control method consists of the following two elements:

- (1) Congestion detection at a router,
- (2) Controlling the Interest forwarding rate at a receiver.

3.2.1 Congestion Detection

A router observes the number of data packets in its data transmission buffer to detect congestions. A threshold is set in the data transmission buffer. When a data packet arrives at a router, the router checks whether the buffer occupancy exceeds the threshold. If the buffer occupancy exceeds the threshold, the router detects a congestion. In that case, the router generates a NACK packet and transmits it to the receiver.

3.2.2 Controlling the Interest Forwarding Rate

When receiving a NACK packet, a receiver realizes that a congestion occurs and halves its window size to decrease the Interest forwarding rate. Controlling the Interest forwarding rate is based on an AIMD window control. To prevent an extreme regulation of the Interest forwarding rate, the window-size decreases at most once per RTT. The exponential weighted average of the observed

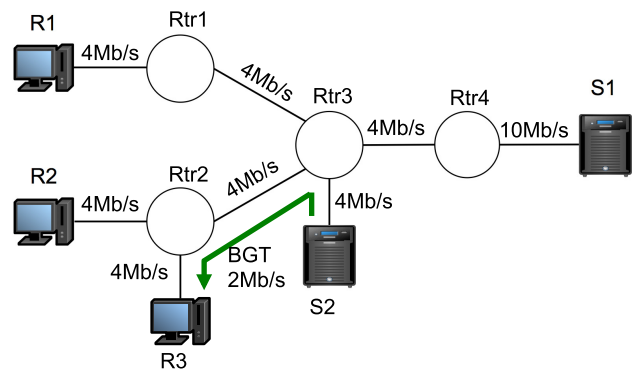


Fig. 1 Simulation environment for fundamental evaluation.

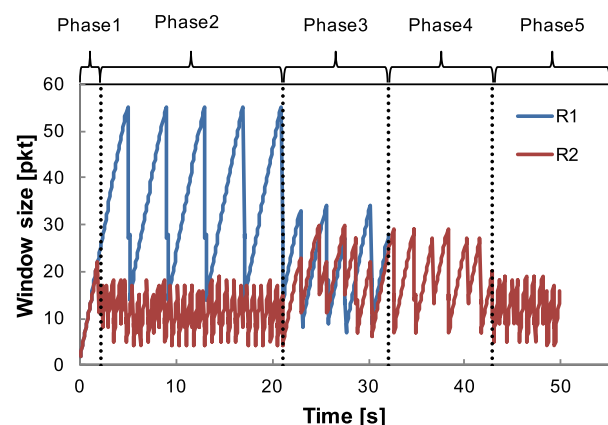


Fig. 2 Transition of window size in the fundamental evaluation.

RTTs is used as the RTT.

3.3 Evaluation Environment

We consider the situation where multiple receivers download the same content simultaneously. **Figure 1** illustrates the network environment in the evaluations. Two receivers, R1 and R2, download the same content from server S1. R1 and R2 start downloading at the same time. In addition, server S2 generates background traffic (BGT) to the receiver R3 according to a Poisson distribution. The BGT shares the link between Rtr2 and Rtr3 with the downloading flow of R2. Thus, R1 is the faster receiver, and R2 is the slower receiver.

We use the ndnSIM[16] network simulator. As shown in Fig. 1, the propagation delay of all links is set to 10 milliseconds. Rtr3 has a CS to cache up to 5,000 data packets. The cache policy is the least recently used (LRU). The other routers have no in-network cache. The content size is 10 Mbytes. The content is divided into 10,000 data packets. The size of each data packet is 1,000 bytes.

3.4 Results

We evaluate the transition of window size at R1 and R2 and the download completion times. The definition of the download completion time is the duration from when the receiver forwards the first Interest to when it receives the last data packet.

Figure 2 shows the transition of window size at R1 and R2 in the experiment. The download completion time of R1 is 32.1 seconds, and that of R2 is 49.9 seconds. Due to the BGT, the down-

load completion time of R2 is approximately 55 percent longer than that of R1. From Fig. 2, we can find that the network situation transits as the follows:

- Phase1 (before 2.1 s): The Interest from R2 arrives at Rtr3 while Rtr3 waits for the R1's requested data packet. Rtr3 aggregates two Interests from R1 and R2. Only the aggregated Interests are sent to Rtr4.
- Phase2 (2.1 to 21.1 s): R1 downloads from S1. The Interest from R2 hits the cache at Rtr3. Therefore, R2 receives the cached data packets from Rtr3.
- Phase3 (21.1 to 32.1 s): Both R1 and R2 download data packets from S1. Since the CS at Rtr3 is replaced by data packets downloaded by R1 and R2, the Interests from R2 do not cache-hit at the CS of Rtr3.
- Phase4 (32.1 to 42.8 s): After R1 finishes downloading all of the requested data packets, no cache-hit occurs at Rtr3. R2 continues to download from S1.
- Phase5 (after 42.8 s): R2 downloads the data packets, which are downloaded and cached in Phase3 by R1, from the CS at Rtr3.

In Phase1, the download speeds of the two receivers (R1 and R2) are equal. When receiving a NACK packet from Rtr3 due to the BGT, R2 decreases its window size. As a result, the download speed differs between R1 and R2.

In Phase2, we observe that the maximum window size at R1 is 55 packets. This implies that R1 can use the entire bandwidth of the link between Rtr3 and Rtr4 without sharing with R2. On the other hand, the window size at R2 is up to 18 packets. In Phase2, the downloaded data packet from S1 is cached in the CS of Rtr3. Therefore, the Interests from R2 hit the cache at Rtr3. At the end of Phase2, no cache hit occurs because the CS at Rtr3 does not have any available cached data packets for R2. This is because the download speed of R1 is much higher than that of R2.

In Phase3, the window sizes at both R1 and R2 are almost the same. In other words, compared with Phase1, the window size at R1 decreases, while that at R2 increases. This occurs because R2 starts downloading from S1. Therefore, both R1 and R2 share the bandwidth of the link between Rtr3 and Rtr4. Thus, the downloading flow of R1 is regulated to half, and the window size decreases. On the other hand, the window size at R2 increases due to the change of the data source from Rtr3 to S1. At the end of Phase3, R1 finishes downloading all of the requested data packets.

In Phase4, R2 still needs to download data packets. R2 continues to download data packets from S1. Finally, in Phase5, R2 can download data packets from the CS at Rtr3. These cached data packets are stored by R1's downloading during Phase3, and these data packets are kept without being replaced. In Phase5, the window size is smaller than that in Phase4. Note that the smaller window size does not mean that the download speed becomes slow. The decrease in window size is due to the shortened RTT for downloading resulting from the cache-hits in Phase5.

Based on the above observation, we found that a problem occurs in the faster receiver (R1) in Phase3. In Phase3, the window size at R1 decreases because R1 and R2 share the link between Rtr3 and Rtr4. As a result, the download speed of R1 be-

comes slow. This phenomenon excessively increases the download completion time of faster receivers when multiple receivers simultaneously download content in a CCN. Therefore, increasing cache-hit opportunities at routers is important for shortening the download completion time. To the best of our knowledge, this is the first paper to report such a problem.

4. Proposed Method

This paper proposes a novel in-network caching for simultaneous download from multiple receivers in a CCN in order to shorten the download completion time of all receivers. The proposed method is designed for the situation where receivers' downloading speeds are different. The proposed method keeps cached data packets in a router until slower receivers download the data, in order to prevent a slower user from directly downloading data from the content provider. The proposed method avoids bandwidth sharing on the bottleneck link (between Rtr3 and Rtr4 in Fig. 1). Consequently, the faster receiver can use the link without sharing it with the slower receiver. Using the proposed method, we attempt to shorten the download completion time.

The proposed method consists of the following two elements:

- (1) Detection of a simultaneous download from multiple receivers at a router.
- (2) Transition of a data cache policy at a router.

4.1 Detection of Simultaneous Download from Multiple Receivers

The proposed method defines a "simultaneous download" as the situation in which a router receives multiple content requests for the same content from different interfaces. Thus, a router detects a simultaneous download based on the name prefix and the Interest's arriving interface.

We introduce a SimultaneousList for each router to identify whether simultaneous downloading occurs. The SimultaneousList tracks the name prefix and arriving interface of the Interest as SimultaneousList entries. **Figure 3** shows an example of a SimultaneousList.

In the proposed method, when a router receives an Interest, the detection of a simultaneous download is performed before checking the CS to determine whether the corresponding data packet is cached. The detection of a simultaneous download consists of the following two procedures:

- (1) The router checks whether the name prefix included in the arrived Interest matches the name prefix of SimultaneousList entry,
- (2) The router checks whether the interface at which the Interest arrives is different from the interface of the SimultaneousList entry.

Name prefix	Interest's arriving interface
ComponentA/Component1/foo.mpg	0
ComponentA/Component1/foo.mpg	1
ComponentB/Component2/bar.mpg	0
...	...

Fig. 3 Example of a SimultaneousList.

Cached Data name	Cache-hit Counter
ComponentA/Component1/foo.mpg/seq5	0
ComponentB/Component2/bar.mpg/seq1	1
...	...
ComponentA/Component1/foo.mpg/seq2	1
ComponentA/Component1/foo.mpg/seq1	2

Fig. 4 Example of the proposed cache structure.

In the first procedure, if the router has already created the SimultaneousList entry containing the name prefix, the router transmits to the second procedure. Otherwise, the router identifies that a simultaneous download did not occur. In this case, the router creates a SimultaneousList entry containing the name prefix and the Interest's arriving interface.

In the second procedure, if the interface at which the Interest arrives is different from the interface of the SimultaneousList entry found in the first procedure, the router determines that a simultaneous download has occurred in terms of the required content. The router adds the interface at which the Interest arrives to the SimultaneousList entry. Otherwise, the router determines that a simultaneous download did not occur.

A router detects a simultaneous download for each content. In other words, if a router receives a data packet of simultaneous download content, the router changes its data cache policy. Otherwise, the router does not change its data cache policy. This is because the router increases its cache-hit rate only for simultaneous download content.

Note that the meaning of “simultaneous” is not strict. The proposed method allows the situation in which a receiver starts downloading content, and, after some time (a few seconds, tens of seconds, or longer, for the case in which the downloading durations of the receivers overlap), another receiver starts to download the same content.

4.2 Transition of Data Cache Policy

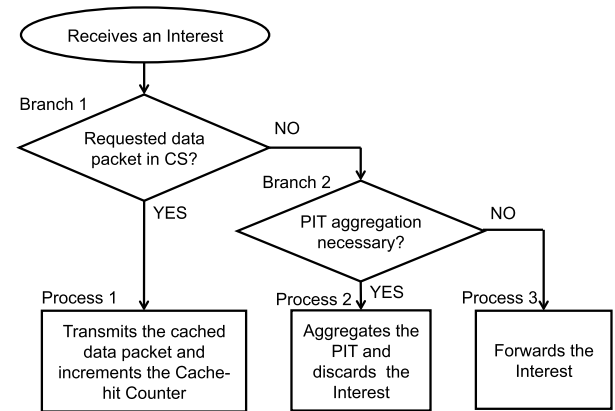
The proposed method has two data cache modes:

- Normal Mode,
- Simultaneous Mode.

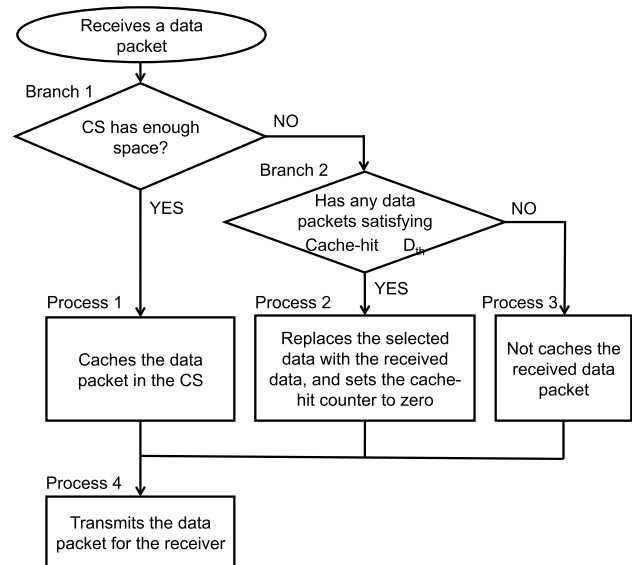
The Normal Mode is used before detecting a simultaneous download from multiple receivers. In the Normal Mode, the cache policy is LRU. On the other hand, the Simultaneous Mode is used after detecting a simultaneous download from multiple receivers. The goal of the proposed method is to keep cached data packet until it hits in the CS at a router.

In the proposed method, we introduce a Cache-hit Counter for counting the number of cache-hits for each data packet in the CS. Figure 4 shows an example of the proposed cache structure.

When caching a data packet, a router sets its Cache-hit Counter to zero. When a cache-hit occurs at a router, the router increments its Cache-hit Counter corresponding to the cache-hit data packet after transmitting it to a receiver. We also introduce a Cache-hit Counter threshold D_{th} to determine whether the cache replacement is necessary when a router receives a data packet.



(a) Operation when a router receives an Interest



(b) Operation when a router receives a data packet

Fig. 5 Flowchart of the Simultaneous Mode of the proposed method.

The flowchart of the Simultaneous Mode in the proposed method is shown in Fig. 5. The operation of the Simultaneous Mode in the proposed method is as follows (the italicized points emphasize the main features of the proposed method):

- (1) When receiving an Interest packet, a router checks whether it has the requested data packet in its CS (Branch 1 in Fig. 5 a).
 - If the requested data packet is cached in its CS, the router transmits the cached data packet to the receiver. *The router increments the Cache-hit Counter corresponding to the cached data packet in its CS* (Process 1 in Fig. 5 a).
 - If not, the router checks whether a PIT aggregation is necessary (Branch 2 in Fig. 5 a).
 - If necessary, the router aggregates the PIT and discards the received Interest (Process 2 in Fig. 5 a).
 - If not, the router forwards the Interest according to its FIB table without PIT aggregations (Process 3 in Fig. 5 a).
- (2) When receiving a data packet, the router checks whether it has enough space in the CS (Branch 1 in Fig. 5 b).
 - If the router has enough space in the CS, the router caches the data packet (Process 1 in Fig. 5 b).
 - If not, *the router checks whether it has any cached data*

packet for which the Cache-hit Counter is equal to or greater than D_{th} (Branch 2 in Fig. 5 b).

- If the router has such cached data packets, the router selects the cached data packet for which the Cache-hit Counter is the largest among those in its CS. If multiple data packets have the largest value, the router selects the least recently used data packet to replace the data packet. The router then replaces the selected data packet with the received data packet. The Cache-hit Counter of the newly cached data packet is set to zero (Process 2 in Fig. 5 b).
- If the router does not have any data packet for which the Cache-hit Counter is equal to or greater than D_{th} , the router discards the data packet without caching it (Process 3 in Fig. 5 b).

The router transmits the data packet through the interface indicated in the PIT for the receiver (Process 4 in Fig. 5 b).

Figure 6 illustrates an example of a cache-hit when a router receives an Interest that contains the data name of “ComponentA/Component1/foo.mpg/seq2”. As shown in the left-hand side of Fig. 6, when receiving an Interest, the router has the corresponding data packet in its CS. Thus, the router transmits the data packet to a receiver. After transmitting the data packet, the router increments the Cache-hit Counter for counting the number of cache-hits as shown in the right-hand side of Fig. 6.

Figure 7 illustrates an example of a replacement when a router receives a data packet that contains the data name of “ComponentA/Component1/foo.mpg/seq6”. As shown in the left-hand side of Fig. 7, the router has two data packets, the Cache-hit Counter of which is the largest in its CS. In this case, the router selects the least recently used data packet (“ComponentA/Component1/foo.mpg/seq1,” in Fig. 7). The router then deletes the selected data packet. After that, the router caches the received data packet (“ComponentA/Component1/foo.mpg/seq6,” in Fig. 7). Finally, the Cache-hit Counter corresponding to the received data packet is set to zero.

Cached Data name	Cache-hit Counter		Cached Data name	Cache-hit Counter
ComponentA/Component1/foo.mpg/seq5	0	→	ComponentA/Component1/foo.mpg/seq5	0
ComponentB/Component2/bar.mpg/seq1	1		ComponentB/Component2/bar.mpg/seq1	1
...
ComponentA/Component1/foo.mpg/seq2	1		ComponentA/Component1/foo.mpg/seq2	2
ComponentA/Component1/foo.mpg/seq1	2		ComponentA/Component1/foo.mpg/seq1	2

Fig. 6 Example of a cache-hit.

Cached Data name	Cache-hit Counter		Cached Data name	Cache-hit Counter
ComponentA/Component1/foo.mpg/seq5	0	→	ComponentA/Component1/foo.mpg/seq6	0
ComponentB/Component2/bar.mpg/seq1	1		ComponentA/Component1/foo.mpg/seq5	0
...
ComponentA/Component1/foo.mpg/seq2	2		ComponentA/Component1/foo.mpg/seq3	1
ComponentA/Component1/foo.mpg/seq1	2		ComponentA/Component1/foo.mpg/seq2	2

Fig. 7 Example of a replacement.

In the proposed method, after finishing a simultaneous download, the router transits from the Simultaneous Mode to the Normal Mode. The router determines the end of the simultaneous download based on the PIT entry. Transmitting all of the data packets of the simultaneous download content (i.e., all of the PIT entries for the content are erased), the router waits for a time equal to the PIT timer. After this time, the router determines the end of the simultaneous download. The router then erases the entry corresponding to the content of the finished simultaneous download from its SimultaneousList.

In the Simultaneous Mode of the proposed method, cached data packets for which the Cache-hit Counter is equal to or greater than D_{th} in its CS, are replaced with the received data packet. When a simultaneous download by multiple receivers occurs, the cached data packets are cache-hit by the slow receiver. Therefore, once a cached data packet is hit (downloaded), the cached data packet is rarely cache-hit, although the router continues to keep the data packet. In the proposed method, a router selects the cached data packet for which the Cache-hit Counter is incremented. The router then replaces the selected data packet with the received packet to be cache-hit in the future.

5. Performance Evaluation

In this section, we evaluate the performance of the proposed method by conducting computer simulations.

5.1 Performance Comparison

We use the same simulation environment described in Section 3.3. In the experiment, we set D_{th} to one.

Figure 8 shows the transition of the window size of the proposed method. In the proposed method, the download completion time of R1 is 27.9 seconds, and that of R2 is 47.4 seconds. The improvement of the download completion time in comparison with the conventional LRU policy is 13.2 percent at R1 and 5.1 percent at R2. Based on this result, we confirm that the proposed method shortens the download completion time of the faster receiver without any performance degradation in the slower receiver.

We observe Phase1 and Phase2 in Fig. 8, as well as in Fig. 2. However, Phase3 does not appear. The download from R1 finishes before Phase3 begins. This result is due to the cache management policy of the proposed method. When transmitting a

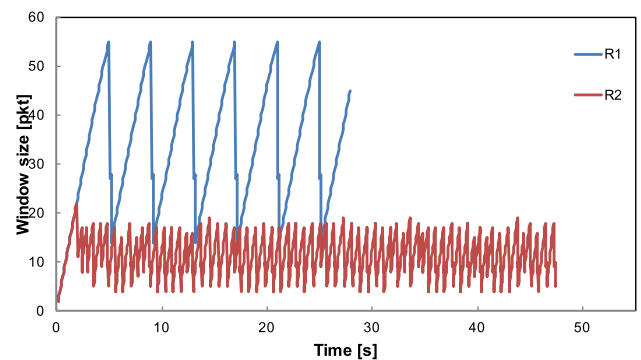


Fig. 8 Transition of window size of the proposed method. (The CS size of Rtr3 is 5,000 packets.)

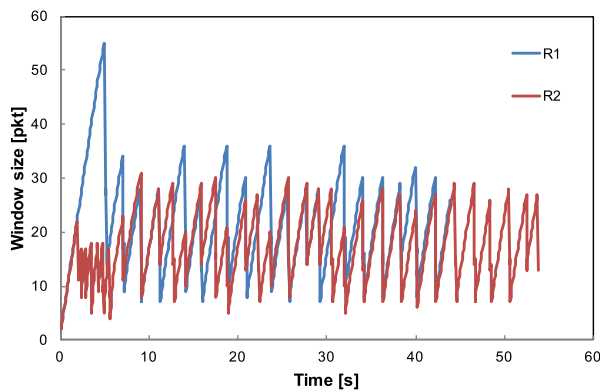


Fig. 9 Transition of the window size of the LRU policy. (The CS size is 1,000 packets.)

data packet from the CS at Rtr3 to R2, the router deletes the cache-hit data packet in its CS. Due to this feature of the proposed method, the CS is not filled with data packets while completing the download of all data packets of R1. Thus, R1 continues to download data packets from S1, and R2 continues to download from the CS at Rtr3. Since R1 can use the link between Rtr3 and Rtr4 without sharing the link with R2, the download speed of R1 does not decrease. This is the reason why the proposed method is effective for decreasing the download completion time of both receivers.

5.2 Evaluation Using Various CS Sizes

Here, we evaluate the transitions of the window size when the CS size of Rtr3 is 1,000 packets. **Figure 9** shows the transition of the window size of the conventional LRU policy, where the CS size of Rtr3 is set to 1,000 packets. From Fig. 9, the interval of Phase2 is approximately from 2.1 to 6.0 seconds, which is shorter than the case in which the CS size is 5,000 packets. This is because when the CS size of Rtr3 is changed to 1,000 packets, the time until the CS is filled with cached data packets is shortened. Therefore, after the CS has no space, cached data packets in the CS are replaced before downloading by R2. This increases the duration of Phase3. Phase3 is approximately from 6.0 to 43.9 seconds, and is longer than that for the case in which the CS size is 5,000 packets. Since Phase3 starts faster, the number of data packets downloaded from S1 increases. This is because both R1 and R2 share the bandwidth of the link between Rtr3 and Rtr4 for the download. Consequently, the download completion time of R1 increases. Phase4 is observed from approximately 43.9 seconds. On the other hand, Phase5 is not observed. All of the cached data packets are replaced by R2's download in Phase4. Hence, Phase5 is not observed.

Figure 10 shows the transition of the window size of the proposed method when the CS size of Rtr3 is set to 1,000 packets. Phase2 finishes at 10.2 seconds. After 10.2 seconds, the window size at R1 decreases and that at R2 increases. This is because R2 starts downloading data packets from S1. Rtr3 caches the received data packets until the CS is full of cached data packets. After being filled with cached data packets, Rtr3 caches a data packet only if the router has a data packet for which the Cache-hit Counter is incremented by one. If Rtr3 does not have such a cached data packet when it receives a data packet, it does not

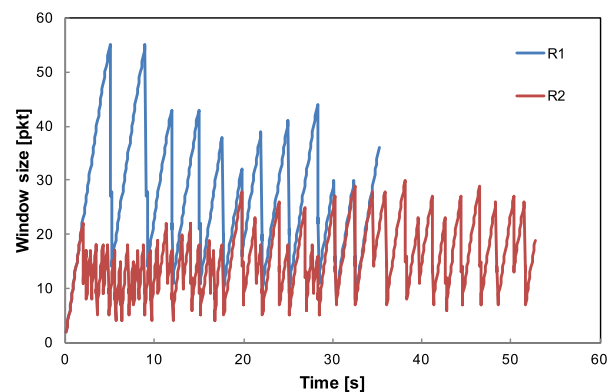


Fig. 10 Transition of window size of the proposed method. (The CS size is 1,000 packets.)

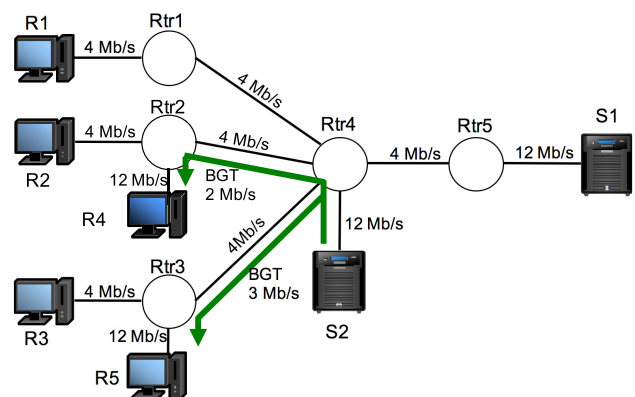


Fig. 11 Simulation environment (three-receiver model).

cache the arriving data packet. Thus, Rtr3 often transmits data packets to R1 without caching in its CS. As a result, R2 downloads from both Rtr3 and S1. Since both R1 and R2 share the bandwidth of the link between Rtr3 and Rtr4, the window size at R1 decreases and that at R2 increases.

We compare the download completion time of the conventional LRU policy to that of the proposed method. In the conventional LRU policy, the download completion time of R1 is 43.9 seconds and that of R2 is 53.8 seconds. In the proposed method, the download completion time of R1 is 35.2 seconds and that of R2 is 52.7 seconds. Therefore, the improvement of the download completion time in comparison with the LRU policy is 19.9 percent at R1 and 2.0 percent at R2. We confirm that the proposed method improves the download completion times of both R1 and R2.

5.3 Three-receiver Model

We evaluate the proposed method in the three-receiver model as illustrated in **Fig. 11**. The propagation delay of all links is set to 10 milliseconds. Rtr4 has a CS to cache up to 5,000 data packets. The other routers have no in-network cache. The content size is 10 Mbytes. The content is divided into 10,000 data packets. The size of each data packet is 1,000 bytes. We set D_{th} to 1. R1, R2, and R3 start downloading at the same time. S2 generates BGT of 2 Mb/s to R4, which shares the link between Rtr2 and Rtr4 with the downloading flow of R2. Moreover, S2 generates BGT of 3 Mb/s to R5, which shares the link between Rtr3 and Rtr4 with the downloading flow of R3. Thus, R1 is the fastest receiver, followed by R2 and R3, which is the slowest receiver.

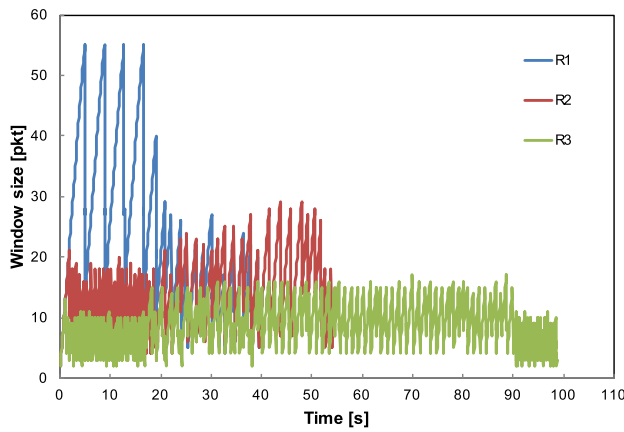


Fig. 12 Transition of the window size of the LRU policy (three-receiver model).

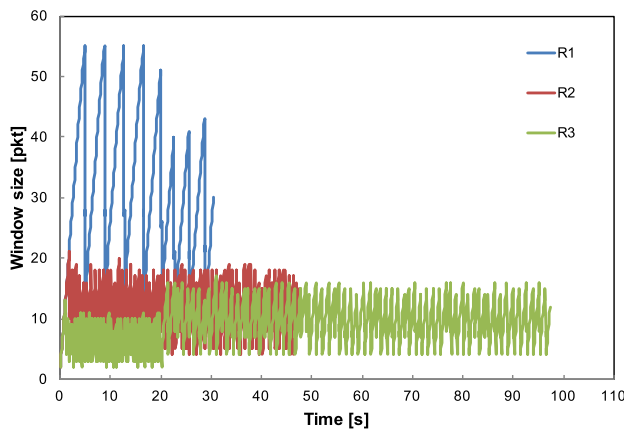


Fig. 13 Transition of the window size of the proposed method (three-receiver model).

Figure 12 shows the transition of the window size of the conventional LRU policy. From Fig. 12, the similar problem when the number of receivers is two is observed in the interval approximately from 19.0 to 40.5 seconds. In this interval, the window size at R1 decreases, and those at R2 and R3 increase. This is because R2 and R3 start downloading from S1. Therefore, R1, R2, and R3 share the bandwidth of the link between Rtr4 and Rtr5. Thus, the downloading flow of R1 is regulated, and the window size at R1 decreases. On the other hand, the window size at R2 and R3 increases due to the change in the data source from R4 to S1. As a result, the download speeds of R1, R2, and R3 become slow.

Figure 13 shows the transition of the window size of the proposed method. As shown in the figure, the above-described problem does not occur. This result is due to the cache management policy of the proposed method. When transmitting a data packet from the CS at Rtr4 to R2 and/or R3, the router increments the Cache-hit Counter related to the cache-hit data packet. Then, Rtr4 replaces the data packet that incremented Cache-hit Counter in the CS with the data packet received from S1. Due to this feature of the proposed method, Rtr4 continues to cache data packets from S1 while R1 completes the download of all data packets. Thus, R1 continues to download data packets from S1, and R2 continues to download from the CS at Rtr4 without downloading from S1. Since R1 can use the link between Rtr4 and Rtr5 without sharing the link with R2, the download speeds of both R1 and

R2 do not decrease. This is the reason why the proposed method decreases the download completion times of both R1 and R2.

Moreover, from Fig. 13, R3 starts downloading data packets from S1 after 20.0 seconds. This is because cached data packets in the CS are replaced before downloading by R3. When R2 downloads data packets from the CS at Rtr4, the router increments the Cache-hit Counter of the cache-hit data packets. After receiving the data packet downloaded for R1, Rtr4 replaces the cached data packet with the received data packet. Thus, in this situation, R1 shares the bottleneck link between Rtr4 and Rtr5 with R3, instead of R2. Therefore, the proposed method with $D_{th} = 1$ can reduce the data traffic in the bottleneck link effectively. If we set $D_{th} = 2$, the router cannot replace a cached data packet until the cached data packet is cache-hit by R3. In this case, before R1 completes to download the entire content, R2 starts to download content from S1. R1 and R2 share the bottleneck link. As a result, it is clear that the download speeds of both R1 and R2 become slower than the case of $D_{th} = 1$.

From Figs. 12 and 13, we compare the download completion time of the conventional LRU policy to that of the proposed method. In the conventional LRU policy, the download completion times of R1, R2, and R3 are 40.5 seconds, 54.4 seconds, and 98.7 seconds, respectively. In the proposed method, the download completion times of R1, R2, and R3 are 30.6 seconds, 47.5 seconds, and 97.3 seconds, respectively. Therefore, the improvement of the download completion time in comparison with the LRU policy is 24.4 percent at R1, 12.7 percent at R2, and 1.4 percent at R3. We confirm that the proposed method shortens the download completion time of all receivers, i.e., R1, R2, and R3. Based on the result in the three-receiver model, we confirm the effectiveness of the proposed method when three receivers download the content simultaneously. In particular, we confirm that the appropriate value of D_{th} is one when the download speed of R2 is faster than that of R3.

5.4 Consideration

The concept of the proposed method is to avoid sharing a bottleneck link among fast receivers. For example, the data packets downloaded by the fastest receiver should be cached for the second fastest receiver. The data packet with D_{th} times of cache-hit becomes a candidate to replace with another data packet. In other words, if we set $D_{th} = 1$, the router caches the data packet downloaded by the fastest receiver for the second fastest receiver. If we set $D_{th} = 2$, the router caches the data packet downloaded by the fastest receiver for both the second fastest receiver and the third fastest one. The appropriate setting of D_{th} depends on the situation. For example, there are three receivers (R1, R2, and R3) downloading the same content with different download speeds. We assume that R1 is the fastest receiver. If the download speed of R2 is much faster than that of R3, $D_{th} = 1$ is appropriate because after a cache-hit by R2, the cached data packet should be replaced with new data packet for R2 rather than the cached data packet kept in the CS for R3. In this case, R1 shares the bottleneck link with R3, instead of R2. However, if we set $D_{th} = 2$, R1 shares the bottleneck link with R2, which is a worse situation than the case $D_{th} = 1$. On the other hand, if the download speed

of R2 is almost the same as that of R3, $D_{th} = 2$ is appropriate because after a cache-hit by R2, the cached data packet should be kept in the CS for R3.

The appropriate setting of D_{th} depends on the situation such as the number of receivers, download speed of each receiver, cache size of routers, and the initial download timing of each flow. Therefore, it is difficult to systematically determine the setting of D_{th} in real networks. D_{th} should be set empirically. One method to empirically set D_{th} is based on some observation at each router. For example, each router observes the cache-hit rate for each data packet of a content flow, which is defined as the number of continuous cache-hits per unit of time. When the rate is large, D_{th} should be large because cached data packets tend to be effectively cache-hit. When the rate is small, D_{th} should be small. In real networks, each router should set D_{th} dynamically based on the observation. The adaptive control of D_{th} is one of the future work.

6. Conclusion

In this paper, we first investigated how a network operates when multiple receivers download content simultaneously in a CCN when the download speeds of the receivers are different. In particular, we focus on the situation where the receivers' downloading speeds differ due to the difference in the bandwidth of the bottleneck link. Based on the results, we proposed a novel in-network caching method for simultaneous download from multiple receivers in a CCN. In order to prevent a slower user from directly downloading data packet from the content provider, the proposed method keeps cached data packets in a router until slower receivers download the data packet. Based on simulation results, we confirmed that the proposed method can improve the download completion time performance for the situation in which multiple receivers download content at different speeds in a CCN. As the future work, we will conduct performance evaluations of the proposed method in large scale networks such as the grid topology and the Rocketfuel model [17]. In addition, we will extend the proposed method in order to control D_{th} adaptively at each router.

Acknowledgments This work was supported by JSPS KAKENHI Grant Number 26280034.

References

- [1] Cisco Visual Networking Index: The Zettabyte Era—trends and analysis, *White Paper* (2015).
- [2] Jacobson, V., Smetters, D., Thornton, J., Plass, M., Briggs, N. and Braynard, R.: Networking named content, *Proc. ACM CoNext2009*, pp.1–12 (2009).
- [3] Psaras, I., Chai, W.K. and Pavlou, G.: Probabilistic in-network caching for information-centric networks, *Proc. ACM ICN2012*, pp.55–60 (2012).
- [4] Sasano, K., Bandai, M. and Yamamoto, M.: A cache management method for simultaneous downloading from multiple receivers for content centric networking, *Proc. TeNAS2016*, pp.872–877 (2016).
- [5] Fdida, S. and Rozhnova, N.: An effective hop-by-hop interest shaping mechanism for CCN communications, *Proc. IEEE INFOCOM2012*, pp.322–327 (2012).
- [6] Rozhnova, N. and Fdida, S.: An extended hop-by-hop interest shaping mechanism for CCN communications, *Proc. IEEE Globecom2014* (2014).
- [7] Carofiglio, G., Gallo, M. and Muscariello, L.: Joint hop-by-hop and receiver-driven interest control protocol for content-centric networks,

Proc. ACM ICN2012, pp.37–42 (2012).

- [8] Carofiglio, G., Gallo, M. and Muscariello, L.: ICP: Design and evaluation of an interest control protocol for content-centric networking, *Proc. IEEE INFOCOM2012*, pp.304–309 (2012).
- [9] Salsano, S., Detti, A., Cancellieri, M., Pomposini, M. and Blefari-Melazzi, N.: Transport-layer issues in information centric networks, *Proc. ACM ICN2012* (2012).
- [10] Saino, L., Cocora, C. and Pavlou, G.: CCTCP: A scalable receiver-driven congestion control protocol for content centric networking, *Proc. IEEE ICC2013* (2013).
- [11] Carofiglio, G., Gallo, M., Muscariello, L. and Papalini, M.: Multipath congestion control in content-centric networks, *Proc. IEEE INFOCOM2013*, pp.3403–3408 (2013).
- [12] Yi, C., Afanasyev, A., Moiseenko, I., Wang, L., Zhang, B. and Zhang, L.: A case for stateful forwarding plane, *Computer Communications*, Vol.36, No.7, pp.779–791 (2013).
- [13] Yamamoto, M.: A survey of caching networks in content oriented networks, *IEICE Trans. Commun.*, Vol.E99-B, No.5, pp.961–973 (2016).
- [14] Badov, M., Seetharam, A., Kurose, J., Firoiu, V. and Nanda, S.: Congestion-aware caching and search in information-centric networks, *Proc. ACM ICN'14*, pp.37–46 (2014).
- [15] Nguyen, D., Sugiyama, K. and Tagami, A.: Congestion price for cache management in information-centric networking, *Proc. IEEE INFOCOM WKSHPs'15* (2015).
- [16] Afanasyev, A., Moiseenko, I. and Zhang, L.: ndnSIM: NDN simulator for NS-3, NDN, Technical Report NDN-0005 (2012), available from <http://named-data.net/techreports.html>.
- [17] Spring, N., Mahajan, R., Wetherall, D. and Anderson, T.: Measuring ISP topologies with rocketfuel, *IEEE/ACM Trans. Networking*, Vol.12, No.1, pp.2–16 (2004).



Takahiko Kato received his B.E. degree from Sophia University in 2016. He is currently a student in Graduate School of Science and Technology, Sophia University. His research interest is information centric networking. He is a member of the IEICE.



Kyouhei Sasano received his B.E. and M.E. degrees from Sophia University in 2014 and 2016, respectively. His research interest is information centric networking.



Masaki Bandai received his B.E., M.E. and Ph.D. degrees from Keio University in 1996, 1998 and 2004, respectively. From 1998 to 2000, he joined Sony Corporation. From 2004 to 2010, he was an assistant professor of Shizuoka University. From 2006 to 2007, he was a visiting assistant professor of the University of British Columbia. He is currently an associate professor of Department of Information and Communication Sciences of Sophia University. His current research interests include computer network systems. He is a member of ACM, IEEE, IEICE and IPSJ.



Miki Yamamoto received his B.E., M.E., and Ph.D. in communications engineering from Osaka University in 1983, 1985, and 1988. He joined the Department of Communications Engineering at Osaka University in 1988. He moved to the Department of Electrical and Electronic Engineering of Kansai University

in 2005, where he is a professor. He visited the University of Massachusetts at Amherst in 1995 and 1996 as a visiting professor. His research interests include content oriented networks, high-speed networks, wireless networks, and the evaluation of performance of these systems. He is a fellow of IEICE and a member of IEEE, ACM, and IPSJ.