A Study on Effective Congestion Control to Retrieve Distributed Data in ICN

Naoki Moriyama[†], Yusuke Ito[‡], Hiroyuki Koga[†]
[†]Graduate School of Environmental Engineering, University of Kitakyushu, Japan
Email: morinao@net.is.env.kitakyu-u.ac.jp, h.koga@kitakyu-u.ac.jp
[‡]Department of Electrical Engineering, Tokyo University of Science, Japan
Email: y-ito@ee.kagu.tus.ac.jp

***Abstract*—Information-centric networking (ICN) is expected as a novel network architecture for content delivery services. In ICN, content routers (CRs) with a cache function store a chunk received from upstream CRs or servers. Therefore, clients can retrieve distributed chunks from multiple nodes simultaneously. However, it will cause heavy congestion due to competition of multiple flows in a network as well as long time to acquire content. In this study, we propose an efficient congestion control scheme to retrieve distributed chunks from multiple nodes in ICN. This scheme aggregates competing flows and then allocates an adequate transmission rate to the aggregated flows. We show the effectiveness of the proposed scheme through simulation evaluations.**

***Index Terms*—ICN (Information Centric Networking), Congestion control, Flow aggregation**

I. Introduction

Information-centric networking (ICN) is expected as a new paradigm for content delivery services. In ICN, content routers (CRs) with a cache function store a part of content (called chunk) received from upstream CRs or servers. Each CR sends requested chunks back to clients instead of servers if they are stored in its cache storage [1]. Namely, clients can retrieve distributed chunks from multiple nodes simultaneously. However, it will cause heavy congestion due to competition of multiple flows in a network as well as long time to acquire content when TCP is commonly employed as a reliable transmission protocol.

Several studies on congestion control algorithms for efficient content delivery in ICN have been reported [2]–[4]. In [2], a receiver-based congestion control algorithm called Remote Adaptive Active Queue Management (RAAQM) has been proposed. This algorithm adjusts window size of each flow based on estimated information (e.g., difference between minimum and maximum round-trip times). A congestion control algorithm proposed in [3] identifies a transmission path by using tag information (e.g., path label), and then adjusts window size of each flow to individual path characteristics. The main drawback of these algorithms is difficult adaptation to network condition fluctuations due to their reactive manner. As an application-specific control, a coordinated video transport protocol in smart camera networks has been proposed [4], which consolidates correlated videos and performs coverage-aware bandwidth allocation.

978-1-7281-2700-2/19/$31.00 2019 © IEEE

In this study, we propose an efficient congestion control scheme with a proactive manner to retrieve various kinds of distributed chunks in ICN. This scheme aggregates competing flows generated to acquire a content item, and then allocates an adequate transmission rate to each aggregated flow to retrieve distributed chunks in a network. We show the effectiveness of the proposed scheme through simulation evaluations.

II. Proposed scheme

In this study, we propose an efficient congestion control scheme to aggregate competing flows and retrieve distributed chunks without competition in ICN. Under this scheme, each CR measures the number of competing flows on its link and then calculates a transmission rate allocated to each flow as shown in Fig. 1. The number of flows is updated when CRs receive request (REQ) or FIN packets. The allocated transmission rate is calculated by dividing link bandwidth by the number of flows. It is notified to source nodes by added to REQ or ACK packets. When the FIN packets are received, i.e., the allocated transmission rate of each flow will increase, the source nodes can adapt it immediately. On the other hand, when the REQ packets are received, i.e., the allocated transmission rates of each flow will decrease, network congestion may occur if new flows start at the allocated transmission rate immediately.

To prevent the congestion, we consider three ways to adjust a transmission rate for a new flow: Step, SlowStart, and Wait methods. The Step method immediately adjusts the transmission rate to the allocated one. The SlowStart method exponentially increases the transmission rate up to the allocated one like TCP's slow-start algorithm. The Wait method keeps REQ packets waiting on CRs until the allocated transmission rate for new flow will be available by the reduction of the allocated transmission rate for existing flows. Note that it adjusts the transmission rate in the same way as the Step method.

III. Performance Evaluation

To investigate the efficiency of our scheme in ICN, we evaluated it after its implementation through simulation using Network Simulator ns-3 [5]. Figure 2 shows the simulation topology. In this simulation, three clients first begin to send a request packet toward servers. At 0.5 seconds after the simulation starts, the other three clients begin to send a request

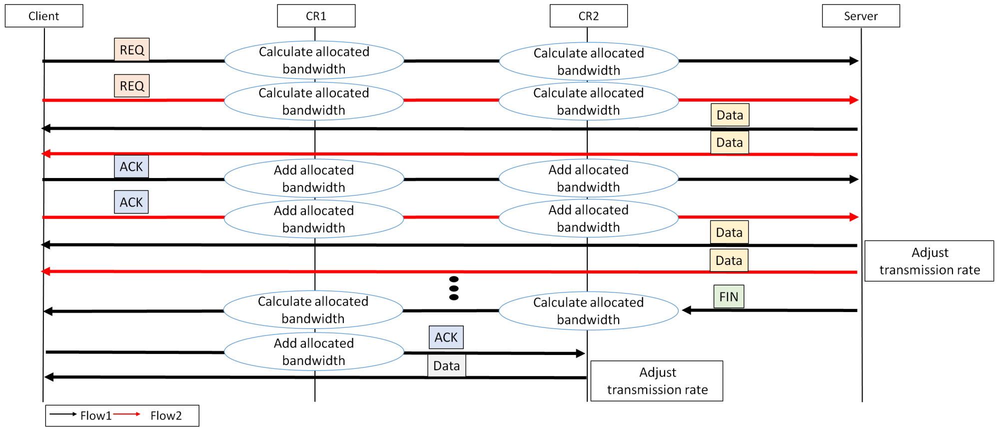

Fig. 1. Sequence diagram

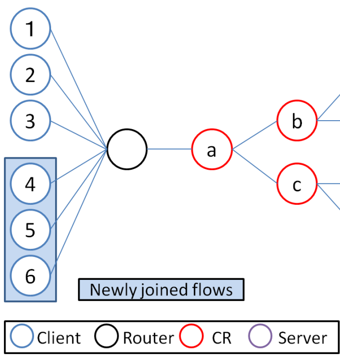

Fig. 2. Simulation topology

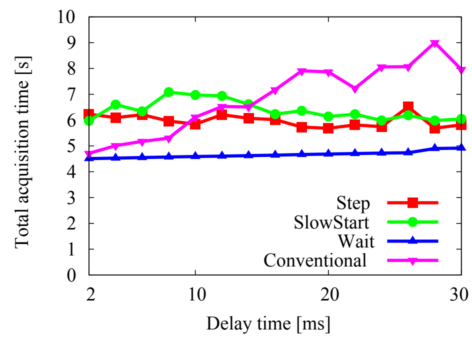

Fig. 3. Acquisition time characteristic

TABLE I
SIMULATION PARAMETERS

Segment size	1000 [Byte]
Data size	1 [MByte]
Buffer size of CRs	400 [packet]
Link bandwidth	100 [Mb/s]
Link Delay time (CRs–servers)	2–30 [ms]
Link Delay time (other links)	5 [ms]
Number of trials	10

packet toward servers. The propagation delay time of links between CRs and servers is set to a uniform random number which ranges from 2 to 30 ms, while that of other links is set to 5 ms. The bandwidth of all links is set to 100 Mb/s. Other simulation parameters are summarized in Table 1. We compare the performance of proposed schemes (Step, SlowStart, and Wait) with that of conventional scheme (original ICN) by focusing on average content acquisition time for all clients.

Figure 3 shows the total content acquisition time for all clients of the proposed and conventional schemes, when the propagation delay time of links between CRs and servers varies from 2 to 30 ms. From Fig. 3, the proposed schemes achieve shorter content acquisition time than the conventional scheme when the delay time is large. In particular, the Wait scheme achieves the shortest content acquisition time among them regardless of the delay time. This is because the proposed schemes can allocate an adequate transmission rate to each flow. Furthermore, the Wait scheme can prevent retransmissions as well as achieve high transmission efficiency by the effect of waiting function to avoid congestion.

IV. CONCLUSION

In this study, we proposed an efficient congestion control scheme to retrieve distributed chunks in ICN. Through simulation evaluations, we have indicated that the proposed scheme can improve content acquisition time by the effect of rate allocation and waiting functions. In our future work, we will evaluate the characteristics of our scheme in detail in more practical environment.

REFERENCES

[1] V. Jacobson et al., "Networking named content," *Proc. ACM CoNEXT2009*, pp. 1–12, Dec. 2009.
[2] G. Carofiglio et al., "Multipath congestion control in content-centric networks," *Proc. IEEE INFOCOM2013*, pp. 363–368, Apr. 2013.
[3] Y. Ye et al., "PTP: Path-specified transport protocol for concurrent multipath transmission in named data networks," *Elsevier Computer Networks*, vol. 144, pp. 280–296, Oct. 2018.
[4] V. Kolar et al., "CTCV: A protocol for coordinated transport of correlated video in smart camera networks," *Proc. IEEE ICNP2016*, 10 pages, Nov. 2016
[5] The ns-3 network simulator, http://www.nsnam.org/.