Adaptive Online Compressing Schemes Using Flow Information on Advanced Relay Nodes

Mei Yoshino, Hiroyuki Koga
Graduate School of
Environmental Engineering
University of Kitakyushu, Japan
Email: moeu@net.is.env.kitakyu-u.ac.jp
h.koga@kitakyu-u.ac.jp

Masayoshi Shimamura
Global Scientific Information
and Computing Center
Tokyo Institute of Technology, Japan
Email: shimamura@netsys.ce.titech.ac.jp

Takeshi Ikenaga
Graduate School of Engineering
Kyushu Institute of Technology, Japan
Email: ike@ecs.kyutech.ac.jp

***Abstract*—As the number of users and applications continues to grow, Internet traffic is growing explosively. Excessive traffic causes network congestion, though, and significantly degrades communication performance. In this paper, we propose adaptive online compressing schemes that use flow information on advanced relay nodes to efficiently reduce the amount of traffic by utilizing network and computational resources. The proposed schemes compress multiple packets forwarded in the same direction by utilizing the waiting time. Furthermore, we evaluate the proposed schemes compared to an adaptive packet compression scheme previously proposed.**

Keywords-adaptive online compression; advanced relay node; network resource; computational resource

I. INTRODUCTION

Continual growth in the number of users and the frequent data exchange of content such as videos and music are causing Internet traffic to increase explosively. According to [1], mobile data traffic will grow at a compound annual growth rate of 66 percent from 2012 to 2017, reaching 11.2 exabytes per month by 2017. When traffic becomes excessive it causes network congestion, which in turn significantly degrades communication performance. Since network resources are limited, they must be used efficiently to alleviate this problem.

To enable efficient use of network resources, an adaptive packet compression scheme has been proposed [2]. This scheme assumes that advanced relay nodes are located inside networks and that these nodes possess not only network resources (i.e., forwarding functions) but also computational resources (i.e., processing functions) [3]. This scheme compresses an incoming packet at the advanced relay nodes while the packet is waiting in an output queue to transfer when congestion occurs. The authors showed numerical results in terms of compression ratio using a data set from an actual network and confirmed the effectiveness of the adaptive packet compression scheme [2], [3]. Even though the adaptive packet compression scheme could reduce the data size by only 5% (i.e., a compression ratio of 0.95), it improved the packet discard ratio and delay time.

In this paper, we suggest adaptive online compressing schemes that use flow information on advanced relay nodes to improve communication performance by reducing the compressed data size more effectively. A key idea is that the proposed schemes compress a block generated from multiple packets forwarded in the same direction (e.g., towards the same destination host or the same subnet) in an output queue at advanced relay nodes. For the block compression, we used a previously reported approach. In [4], to shrink the data size of archival traffic dump data, the authors focus on correlations between header fields among multiple packets. They then show that the compression ratio can be improved by rearranging header fields so as to store similar fields into a single block. In our case, since we compress a block of multiple packets going in the same direction, these packets have similar header fields (e.g., the destination IP address). Therefore, we expect the compression ratio to be improved. However, if the proposed schemes attempt to compress a large block generated from many packets, the compression opportunity can be lost. This is because the proposed schemes cannot gather the packets before the packets are transmitted from an output queue. To efficiently compress a block, we propose two compression schemes: (1) a flow compression scheme which compresses packets having the same 5-tuple header information, and (2) an edge compression scheme which compresses packets passing through the same egress edge of advanced relay nodes. Furthermore, we investigate the effect of the number of compression packets and the compression time when the proposed schemes are used. Through simulations, we show the potential and effectiveness of the proposed schemes.

The remainder of this paper is organized as follows. In Section 2, we describe related studies in terms of data compression. In Section 3, we explain the proposed schemes. We describe the simulation environment in Section 4 and the simulation results in Section 5. We conclude in Section 6.

II. RELATED WORK

As stated in Section 1, several data compression schemes have been proposed. In this section, we first describe the adaptive packet compression scheme, which is the basis of our proposed schemes. We then describe IPzip from a perspective of efficient multiple packet compression.

Copyright (c) IARIA, 2014. ISBN: 978-1-61208-318-6

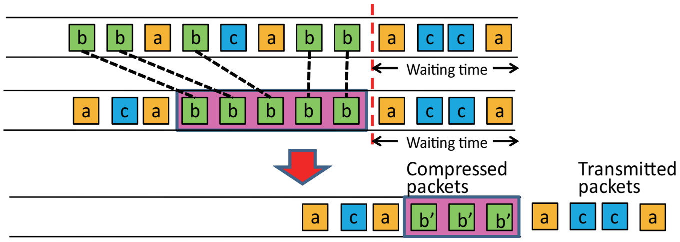


Figure 1. Behavior of the proposed schemes

A. *Adaptive packet compression scheme*

The adaptive packet compression scheme [2] aims to improve communication performance by efficiently using both network and computational resources. When an advanced relay node receives a packet, the node calculates the waiting time of the packet inside its output queue, and then it decides to compress the packets if the waiting time is sufficiently large. Since it compresses packets by exploiting the waiting time, the processing time of compression becomes nearly zero. Therefore, the adaptive packet compression scheme achieves better online packet compression at advanced relay nodes.

Through data analysis using an actual data set, the authors found that packets can be classified into compressible and incompressible packets. They showed that the average compression ratio of all the packets was 0.945 and that of the compressible packets was 0.929. These results showed that actual traffic volume could be reduced by packet compression even if the compression ratio was high (i.e., less than a 10% reduction of data size).

In [2], the authors also showed the effectiveness of the adaptive packet compression scheme through simulation evaluation. In this evaluation, the compression ratio was set to 0.95. Simulation results showed that the adaptive packet compression scheme improved the packet discard ratio and delay time even though it could reduce the data size by only 5%.

B. *IPzip*

IPzip [4] compresses a block created from multiple packets and is used to reduce the data size of stored traffic dump data. The authors focus on similarities among these packets. For example, if packets are transferred to the same destination, these packets have the same destination IP address in their header fields. IPzip rearranges header fields inside stored dump data so as to collect the same or similar information inside header fields, and then it compresses all the data that has rearranged header fields. IPzip can achieve better compression with a low compression ratio through this sophisticated compression approach.

III. ADAPTIVE ONLINE COMPRESSING SCHEMES

In this section, we describe adaptive online compressing schemes. We first describe an overview of our idea and then propose two kinds of compression scheme.

A. *Overview*

Unlike the adaptive packet compression scheme [2], our proposed schemes gather multiple packets adaptively in an output queue at advanced relay nodes and compress them by utilizing the waiting time. Various criteria can be used to create blocks: (1) flow (i.e., same source and destination IP addresses, source and destination port numbers, and protocol number), (2) service (i.e., same destination address and port number, and protocol number), (3) host-by-host (i.e., same source and destination addresses), and (4) destination group (i.e., same network address). The proposed schemes compress multiple packets forwarded in the same direction using flow and destination group information (i.e., 5-tuple header information or information of passing through the same egress edge of advanced relay nodes). Fig. 1 illustrates an example of the proposed schemes' behavior. In this figure, multiple packets forwarded in the same direction (packets b) are grouped and compressed while they are waiting in the output queue. To compress a block generated from multiple packets, our proposed schemes need more processing time than is needed for a packet compression scheme. We define the time needed to compress a block as the "compression time". Moreover, we define the number of packets to be compressed as the "number of compression packets". Let n be the compression time and m be the number of compression packets. The proposed schemes compress m packets forwarded in the same direction when the queue length is more than $n+m$ packets. Note that we normalize the compression time using the packet transmission time, so that we represent the number of packets as the compression time.

B. *Compression schemes*

To efficiently compress a block, we propose two types of block compression: a flow compression scheme and an edge compression scheme. Both generate a block from multiple packets having the same information in a part of the header field. However, the two schemes generate a block differently.

Flow compression scheme

The flow compression scheme generates a block by using information related to end nodes. This scheme gathers multiple packets having the same 5-tuple header information (i.e., source and destination IP addresses, source and destination port numbers, and protocol number) in the output queue, and generates a block from these.

Edge compression scheme

The edge compression scheme gathers multiple packets passing through the same egress edge of an advanced relay node in the output queue, and generates a block from these. Therefore, this scheme compresses a block containing different transport flows.

IV. SIMULATION ENVIRONMENT

In this section, we evaluate the proposed schemes in comparison with the adaptive packet compression scheme through simulations. First, we describe the simulation model and evaluation indices. We use network simulator ver. 2.35 [5] after implementing the proposed schemes.

A. *Simulation model*

Fig. 2 shows the network topology. In this simulation model, congestion can occur at links between an ingress edge

TABLE I. SIMULATION PARAMETERS

Buffer size on each node	50 [packet]
Transport layer protocol	TCP with SACK option
Packet size	1500 [Byte]
Number of compression packets	1–20
Compression time	5–30 [packet]

of advanced relay nodes and a core router. A TCP sender node $S_{i,j}$ sends packets toward a TCP receiver node $R_{i,j}$ connected to an egress edge node E_i, where i represents the number of ingress and egress edge nodes and j represents the number of end nodes connected to a single edge node. The links between each ingress or egress edge node and core routers have a bandwidth of 100 Mb/s and a delay time of 3 ms, while the bottleneck link between core routers has a bandwidth of 200 Mb/s and a delay time of 5 ms. All other access links between each sender or receiver node and the ingress edge or egress edge nodes have a bandwidth of 100 Mb/s and a delay time of 1 ms. The proposed schemes compress packets at ingress edge nodes and expand them at egress edge nodes. If the compressed packets are lost, ingress edge nodes retransmit them.

As the simulation parameters, we set the compression ratio of the adaptive packet compression scheme to 0.95 in accordance with [2]. To determine the compression ratio of the proposed schemes, we preliminarily investigated the compression ratio of multiple packets (from 1 to 100) using the Lempel-Ziv-Oberhumer (LZO) compression algorithm with a data set from an actual network (4.7 GB, approximately 50 million packets). Through this investigation, we found that the compression ratio varied approximately from 0.25 to 0.95. In this simulation, using the mean values of the preliminary results, we set the compression ratio of the proposed schemes to 0.6 or 0.5 when the number of compression packets is 5 or 10, respectively. Other simulation parameters are summarized in Table I.

We investigate the effect of the number of compression packets on communication performance. In this scenario, the number of end nodes pairs varies from 9 to 300 (multiples of three) and a single TCP flow will flow between each pair of end nodes, so there are 9 to 300 TCP flows.

B. Evaluation indices

To evaluate the effectiveness of the proposed schemes, we focus on the total throughput performance as an evaluation index. The total throughput is calculated by summing the throughput of all TCP flows from 10 to 30 seconds after the simulation starts to avoid the influence of a transient period and it is averaged over 10 simulation runs with different random seeds. To analyze the results, we also investigate the number of compression processings.

V. SIMULATION RESULTS

In this section, we show evaluation results of the proposed schemes compared with the performance of the adaptive packet compression scheme. First, we investigate the throughput performance of each scheme. We then examine the effect of each parameter on throughput performance.

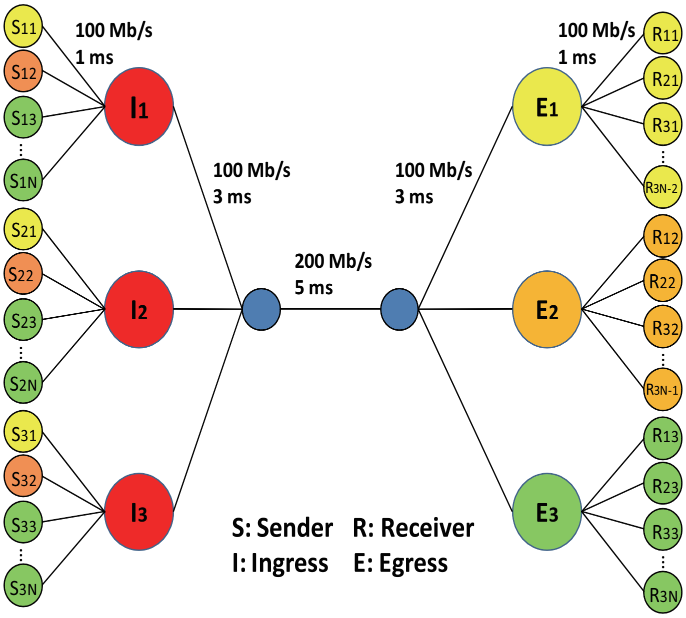


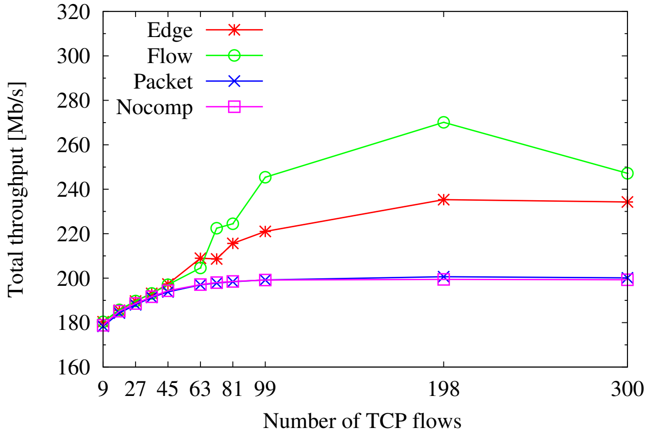
Figure 2. Simulation topology

A. Throughput characteristics of each scheme

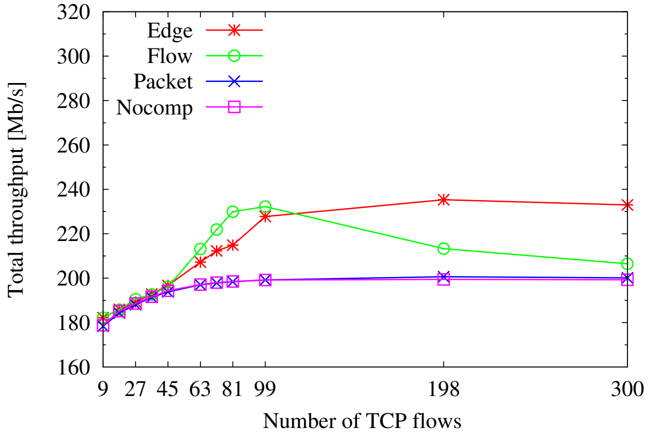
We first evaluate the throughput performance of each scheme. Fig. 3 shows the total throughput of the proposed schemes, the adaptive packet compression scheme, and a no-compression scheme (simply relaying all packets without packet compression) when the number of TCP flows varies from 9 to 300. In this figure, “Edge” denotes the edge compression scheme, “Flow” denotes the flow compression scheme, “Packet” denotes the adaptive packet compression scheme, and “Nocomp” denotes the no-compression scheme. The number of compression packets for the proposed schemes is set to 5 (the compression ratio is 0.6) or 10 (the compression ratio is 0.5), while the compression ratio of the adaptive packet compression scheme is set to 0.95, as described in the previous section. The compression time is set to the time needed to forward 5 packets.

Figs. 3(a) and 3(b) show that the total throughput of the edge and flow compression schemes is higher than that of the other schemes regardless of the number of compression packets. The throughput of the proposed schemes exceeds the bottleneck link bandwidth by effectively compressing multiple packets, while that of the adaptive packet compression and no-compression schemes is limited by the bandwidth. In the case of a small number of compression packets, the flow compression scheme attains higher throughput than the edge compression scheme over a wide range of the number of TCP flows. However, the throughput of the flow compression scheme decreases as the number of TCP flows increases when the number of compression packets is large.

Let’s investigate the reason for the above phenomenon. Figs. 4(a) and 4(b) respectively show the number of compression processings for each scheme when the number of TCP flows varies from 9 to 300 and when the number of compression packets of the proposed schemes is set to 5 or 10. The number of compression processings is approximately the same for the flow and edge compression schemes when the number

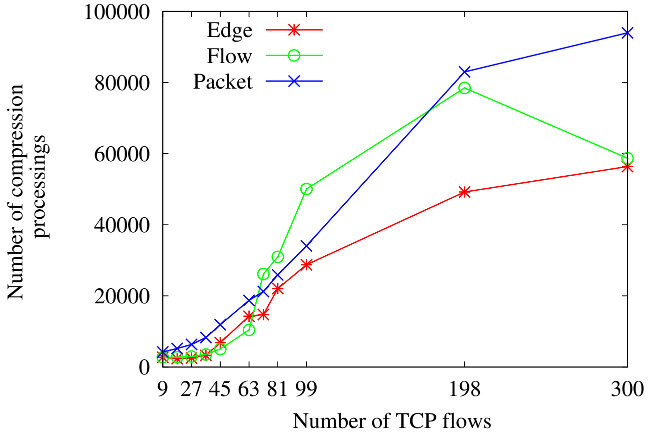


(a) Number of compression packets: 5

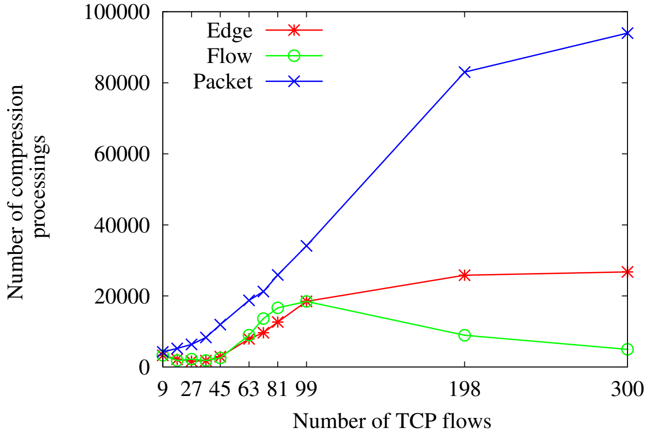


(b) Number of compression packets: 10

Figure 3. Throughput performance



(a) Number of compression packets: 5



(b) Number of compression packets: 10

Figure 4. Number of compression processings

of TCP flows is less than 63. On the other hand, the number of compression processings of the flow compression scheme falls as the number of TCP flows increases, especially in the case of a large number of compression packets, while that of the edge compression scheme increases as the number of TCP flows increases. This is because the flow compression scheme has difficulty gathering multiple packets having the same flow information due to the limited buffer size. Compared to the flow compression scheme, the edge compression scheme can compress packets much more often because each of the ingress edge nodes needs to handle only three types of packet going toward the egress edge nodes. Therefore, the edge compression scheme can better maintain throughput performance than can the flow compression scheme in the case of a large number of compression packets.

The above results demonstrate that the proposed schemes can improve throughput performance compared with the other schemes. In the following subsection, we discuss the effect of each parameter of the proposed schemes on the throughput performance.

B. Effect of each parameter

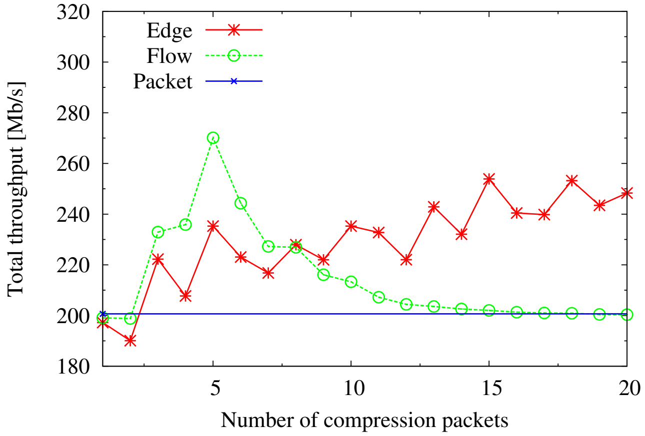
To analyze the performance between the flow and edge compression schemes in detail, we investigate the effect of each parameter on the throughput performance. The performance of the proposed schemes depends on how many packets are successfully compressed. Namely, the main factors determining performance are the number of compression packets and the compression time. In this subsection, we examine the effect of these parameters on the throughput performance and the number of compression processings.

First, we focus on the effect of the number of compression packets on the throughput performance. Figs. 5(a) and 5(b) show the total throughput and the number of compression processings of each scheme when the number of compression packets varies from 1 to 20, respectively. The number of TCP flows is set to 198, where the number of compression packets has a significant impact on the throughput performance of the flow and edge compression schemes as shown in Fig. 3. The compression ratio is set to the average value of 0.6 regardless of the number of compression packets in order to focus on the opportunity of compression in the flow and edge compression schemes. The compression time is set to 5 packets.

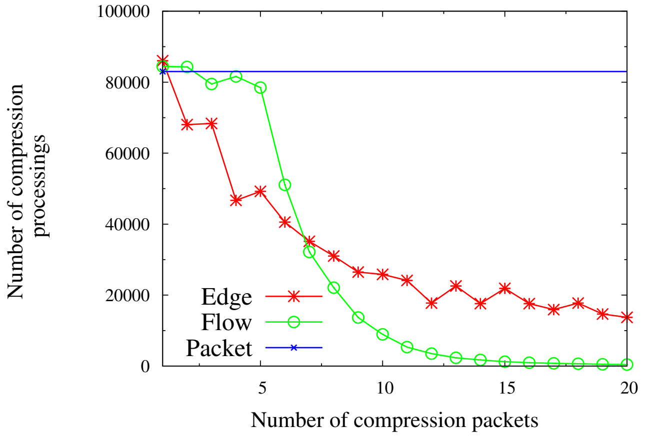
As shown in Fig. 5(a), the flow compression scheme enables excellent throughput when the number of compression packets is small, especially in the case of 5. However, the throughput of the flow compression scheme drastically decreases as the number of compression packets increases. On the other hand, the throughput of the edge compression scheme increases as the number of compression packets increases. Consequently, the edge compression scheme maintains high throughput over a wide range of the number of compression packets.

In order to understand the reason for this, let's consider the number of compression processings shown in Fig. 5(b). The number of compression processings in the flow compression scheme drastically decreases as the number of compression packets increases. This is because the flow compression scheme has difficulty gathering multiple packets having the same flow information due to the limited buffer size as discussed in the previous subsection. On the other hand, although the number of compression processings in the edge compression scheme decreases as the number of compression packets increases, it remains higher than that in the flow compression scheme when the number of compression packets is large. That is, compared to the flow compression scheme, the edge compression scheme can compress packets much more often because each of the ingress edge nodes has to handle only three types of packet going toward the egress edge nodes.

These results demonstrate that the flow compression scheme enables higher throughput than the edge compression scheme with a small number of compression packets, while the edge compression scheme maintains high throughput with a large number of compression packets. The flow compression scheme gathers multiple packets having the same 5-tuple header information; i.e., that belong to a flow. In contrast, the edge compression scheme gathers multiple packets passing through the same egress edge nodes; i.e., that belong to multiple flows. With a small number of compression packets, since the opportunity of compression between the flow and edge compression schemes is almost the same, the flow compression scheme can rapidly increase the throughput of the flow, while the edge compression scheme can increase the throughput of the multiple flows only gradually. Therefore, the flow compression scheme can maintain a large number of compression processings as well as excellent throughput. On the other hand, with a large number of compression packets, the compression opportunity in the flow compression scheme is much smaller than that in the edge compression scheme. As a result, the edge compression scheme can obtain higher throughput than that of the flow compression scheme.



(a) Throughput



(b) Number of compressing processings

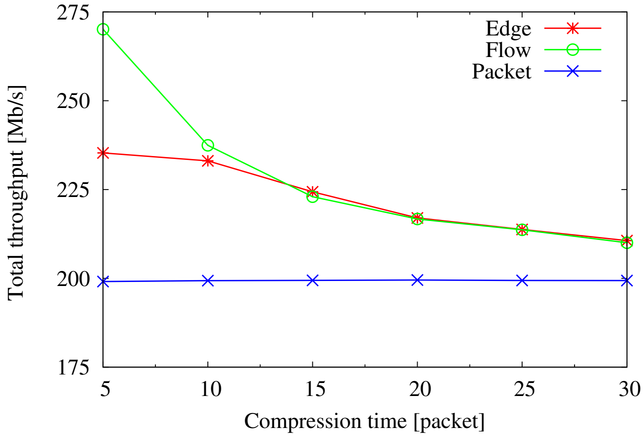
Figure 5. Effect of the number of compression packets

Next, we focus on the effect of compression time on the throughput performance. Figs. 6(a) and 6(b) respectively show the total throughput of each scheme when the compression time varies from 5 to 30 packets and when the number of compression packets of the proposed schemes is set to 5 (compression ratio: 0.6) or 10 (compression ratio: 0.5). The number of TCP flows is set to 198. The throughput of the flow and edge compression schemes increases as the compression time decreases. That is, the proposed schemes improve the performance as the processing speed on the edge nodes will have been higher. Similar to the results above, the flow compression scheme enables higher throughput than the other schemes with a small number of compression packets. However, the throughput of the flow compression scheme decreases as the compression time increases and is appropriately equal to that of the edge compression scheme with a large compression time. On the other hand, with a large number of compression packets, the edge compression scheme attains higher throughput than the other schemes over a wide range of compression time.

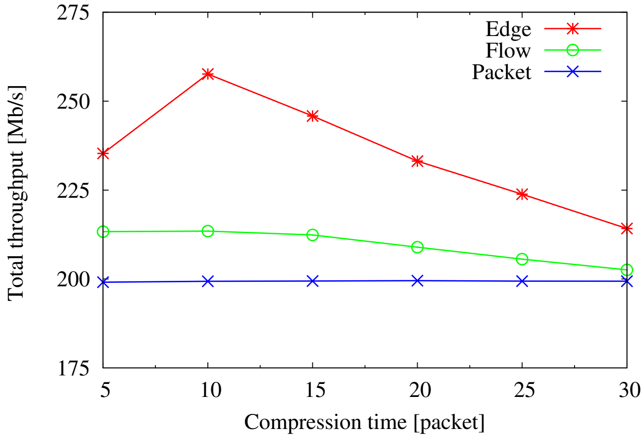
Figs. 7(a) and 7(b) respectively show the number of compression processings when the compression time varies from 5 to 30 packets and when the number of compression packets of the proposed schemes is set to 5 (compression ratio: 0.6) or 10 (compression ratio: 0.5). The number of TCP flows is set to 198. The number of compression processings of each scheme decreases as the compression time increases because a large compression time reduces the buffer capacity available to gather multiple packets having the same information as well as the compression opportunity. With a small number of compression packets, the flow compression scheme attains a larger number of compression processings than the edge compression scheme when the compression time is small. However, with a large compression time, there are no differences between the number of compression processings of each scheme. On the other hand, with a large number of compression packets, the number of compression processings of the edge compression scheme exceeds that of the flow compression scheme and is as large as that of the adaptive packet compression scheme with a large compression time.

These results demonstrate that the proposed schemes can

Copyright (c) IARIA, 2014. ISBN: 978-1-61208-318-6

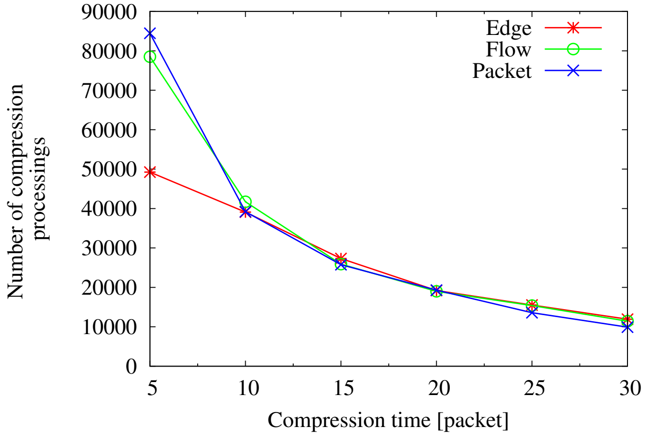


(a) Number of compression packets: 5

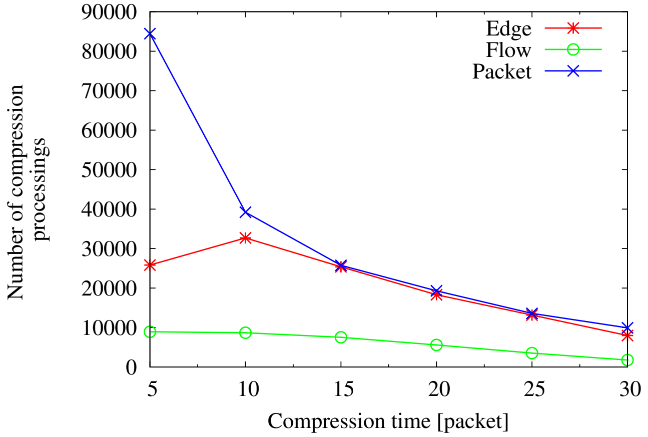


(b) Number of compression packets: 10

Figure 6. Effect of compression time: Throughput



(a) Number of compression packets: 5



(b) Number of compression packets: 10

Figure 7. Effect of compression time: Number of compression processings

improve the throughput by adaptively compressing multiple packets gathered in an output queue at edge nodes through utilization of the waiting time. The flow compression scheme enables high throughput with a small number of compression packets and a small compression time. Otherwise, the edge compression scheme enables higher throughput.

VI. CONCLUSION

To improve communication performance by efficiently decreasing Internet traffic, we have proposed adaptive online compression schemes that use flow information in advanced relay nodes. The proposed schemes gather adaptively multiple packets forwarded in the same direction in an output queue at advanced relay nodes and compress them by utilizing the waiting time. Through evaluations by simulation, we have shown that the proposed schemes enable high communication performance by compressing multiple packets. The flow compression scheme enables high throughput with a small number of compression packets and a small compression time. Otherwise, the edge compression scheme enables higher throughput. In our future work, we will design dynamic online compression algorithms that can adapt compression methods to network conditions and evaluate the proposed schemes using a prototype implementation from a viewpoint of the computational resources (processing time, memory usage, etc.).

ACKNOWLEDGMENT

This work was supported in part by the Japan Society for the Promotion of Science, Grant-in-Aid for Young Scientists (B) (No. 25730063).

REFERENCES

[1] Cisco Systems, Inc., “Cisco visual networking index: Global mobile data traffic forecast update, 2012–2017,” Feb. 2013. Information available at http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white paper c11-520862.pdf. [Retrieved: Dec. 2013]

[2] M. Shimamura, H. Koga, T. Ikenaga, and M. Tsuru, “Compressing packets adaptively inside networks,” IEICE Transactions on Communications, vol. E93-B, no. 3, Mar. 2010, pp. 501–515.

[3] M. Shimamura, T. Ikenaga, and M. Tsuru, “A design and prototyping of in-network processing platform to enable adapting network service,” IEICE Transactions on Information and Systems, vol. E96-D, no. 2, Feb. 2013, pp. 238–248.

[4] S. Chen, S. Ranjan, and A Nucci, “IPzip: A stream-aware IP compression algorithm,” Proc. IEEE Data Compression Conference (DCC’08), Mar. 2008, pp. 182–191.

[5] The Network Simulator, http://www.isi.edu/nsnam/ns. [Retrieved: Dec. 2013]