PAPER Special Section on Network Resource Control and Management for IoT Services and Applications

A Bandwidth Allocation Scheme to Improve Fairness and Link Utilization in Data Center Networks*

Yusuke ITO[†a)], ***Student Member***, **Hiroyuki KOGA**[†b)], ***Member***, ***and*** **Katsuyoshi IIDA**[††c)], ***Senior Member***

SUMMARY Cloud computing, which enables users to enjoy various Internet services provided by data centers (DCs) at anytime and anywhere, has attracted much attention. In cloud computing, however, service quality degrades with user distance from the DC, which is unfair. In this study, we propose a bandwidth allocation scheme based on collectable information to improve fairness and link utilization in DC networks. We have confirmed the effectiveness of this approach through simulation evaluations.

key words: *bandwidth allocation, fairness, link utilization, TCP, data center, and data center networks*

1. Introduction

Users can now connect to the Internet through a wide variety of access networks and communication terminals such as smart phones or PCs. This has brought a great demand for cloud computing, which allows users to enjoy various Internet services provided by data centers (DCs) [4] anytime, anywhere. In the Internet of Things (IoT), cloud computing has been more attractive and important since users can enjoy IoT services using various sensor information collected by DCs. Major services available through cloud computing include file sharing or transaction services which commonly use TCP [5] as a reliable data transmission protocol.

In cloud computing, a problem arises that a user's quality of service in terms of transmission rate unfairly depends on the user's distance from DCs. This problem is caused by how TCP controls congestion. TCP cannot exactly identify the conditions of other flows, so it estimates an available bandwidth based on packet losses. Since congestion control is applied according to round-trip time (RTT), the transmission rate for a user with a short RTT is effectively increased compared with that of a user having a long RTT. Fairness in terms of throughput among flows with different RTTs as well as link utilization in DC networks needs to be improved.

There are two approaches to improving fairness among flows. The first approach is to apply active queue management (AQM) technologies on routers. This approach realizes fair communication among flows by preferentially discarding packets which belong to a high-rate flow. The second approach is to use router-assisted congestion control algorithms. In this approach, routers provide the TCP sender with information about available bandwidth and the TCP sender adjusts the congestion window based on the information received from routers. These approaches enable more fairness than the original TCP. However, the fairness among competing flows will be extremely degraded if the flows do not share the same bottleneck router that uses the AQM technologies or provides the TCP sender with the information about available bandwidth.

In this study, we propose a bandwidth allocation scheme based on collectable information to improve fairness and link utilization in DC networks. This scheme collects flow information including the bandwidth of each link, the number of competing flows, the RTT of each flow, and the actual throughput of each flow from routers and servers in DC networks, and then fairly allocates transmission rates among flows based on the collected information. In addition, this scheme reallocates unutilized bandwidth to other competing flows in DC networks when bottleneck links exist outside of the DC networks. We show the effectiveness of this approach through simulation evaluations.

The rest of this paper is organized as follows. In Sect. 2, we describe related work. In Sect. 3, we propose a bandwidth allocation scheme based on collectable information to improve link utilization in DC networks as well as fairness among flows. We describe our simulation environment and results in Sects. 4 and 5, respectively. We conclude in Sect. 6.

2. Related Work

One of the important issues in DC networks is to improve fairness among flows [6]. Several studies report fairness issues in DC networks such as TCP Outcast problem [7]. For example, data center TCP (DCTCP) [8] and SAB [9] have been proposed to solve this problem. DCTCP employs a threshold on switches and informs the TCP sender about congestion condition by using a marking algorithm when the switches' buffer size exceeds the threshold. SAB notifies the TCP sender of information about available bandwidth at switches in DC networks. Since these studies focus only on fairness among competing flows at one output port of switches in DC networks, fairness among all flows having

Manuscript received March 27, 2017.
Manuscript revised July 15, 2017.
Manuscript publicized September 19, 2017.

[†]The authors are with the Graduate School of Environmental Engineering, University of Kitakyushu, Kitakyushu-shi, 808-0135 Japan.

[††]The author is with the Information Initiative Center, Hokkaido University, Sapporo-shi, 060-0811 Japan.

*The early versions of this work were presented at ACM CoNEXT2014 [1] and IEEE PerCom2016 [3], and published in IEICE ComEX [2].

a) E-mail: wash000i@net.is.env.kitakyu-u.ac.jp
b) E-mail: h.koga@kitakyu-u.ac.jp
c) E-mail: iida@iic.hokudai.ac.jp

DOI: 10.1587/transcom.2017NRP0008

different RTTs existed in DC networks needs to be improved as mentioned in Sect. 1.

To improve fairness among flows, two approaches have been proposed. The first approach is to apply AQM algorithms on routers. In the past, many kinds of AQM algorithms have been proposed for use on routers to improve fairness among flows. A typical AQM algorithm is random early detection (RED) [10]. This approach sets maximum and minimum queue lengths and drops packets when the current queue length is larger than the set maximum queue length. This enables fair communication among competing flows by preferentially discarding packets which belong to a high rate flow. However, it is difficult to determine an appropriate maximum and minimum queue length because the appropriate values depend on network conditions such as the number of flows and delay time. Unlike RED, the CHOKe AQM algorithm does not require parameter settings [11]. This approach randomly selects one packet from the output queue on routers when the router receives a packet. If the source addresses of the received and selected packets are the same, both packets should be dropped. CHOKeW and CHOKeR are improved versions of the CHOKe algorithm [12], [13]. These approaches adjust the packet dropping rate of each flow based on service classes and network conditions.

The second approach is to use router-assisted congestion control algorithms. A typical router-assisted congestion control algorithm is the eXplicit Control Protocol (XCP) [14]. In XCP, routers provide TCP senders with information about available bandwidth. When a TCP sender receives this information, it adjusts a congestion window size based on the received information. Another router-assisted congestion control algorithm is proposed by L. Kalampoukas et al. [15]. This approach modifies a TCP receiver's advertised window size in packets forwarding to TCP senders on intermediate routers. These approaches can improve fairness among competing flows. However, the fairness among flows will be extremely degraded if the flows do not share the same bottleneck router using the AQM algorithms or providing the TCP sender with the information about available bandwidth.

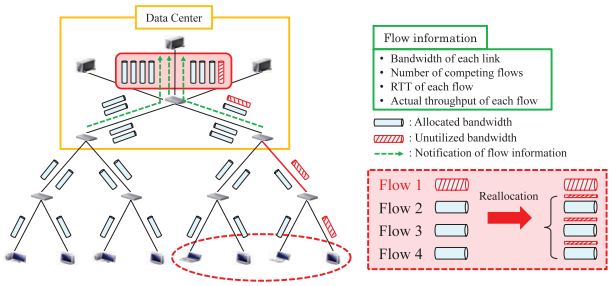


Fig. 1 Overview of bandwidth allocation scheme.

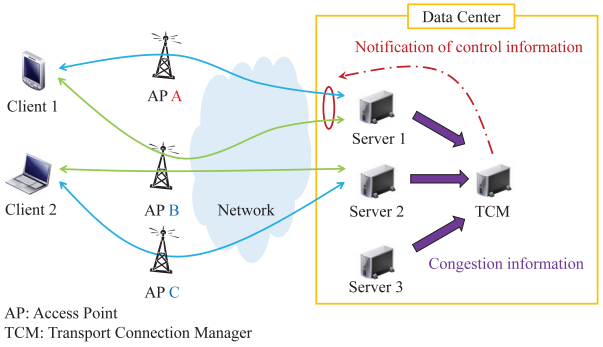


Fig. 2 Unified central congestion control architecture.

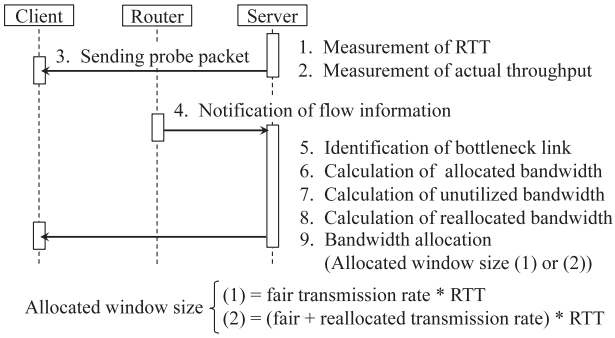


Fig. 3 Operation of bandwidth allocation scheme.

3. Proposed Scheme

As mentioned in Sect. 1, fairness among flows with different RTTs will be extremely degraded because TCP does not all the available bandwidth to be accurately estimated. To solve this problem, we propose a scheme that fairly allocates transmission rates among flows based on collectable information in DC networks. This scheme identifies bottleneck links based on flow information collected from routers and servers in DC networks, and then calculates the transmission rate that should be allocated to each flow by dividing the bandwidth of a bottleneck link by the number of competing flows. In addition, this scheme reallocates any unutilized bandwidth caused by congestion outside DC networks to other competing flows to improve link utilization in DC networks when bottleneck links exist outside DC networks, as shown in Fig. 1. Namely, we propose two bandwidth allocation schemes to improve fairness and link utilization in DC networks. They are the proposed (no reallocation) and (reallocation) schemes, which only allocates a fair transmission rate to each flow and additionally allocates unutilized bandwidth to competing flows, respectively. Flow information such as the bandwidth of each link, the number of competing flows, the RTT of each flow, and the actual throughput of each flow is collected by a Unified Central Congestion Control Architecture (UC^3) [16], which uniformly manages the congestion information in the networks, as shown in Fig. 2. Namely, this scheme can adapt to network environment where is managed by an organization such as DC networks because it is easy to collect flow information from their routers, although it is hard to collect flow information from any routers in the Internet.

We describe here the operation of the bandwidth allocation scheme shown in Fig. 3. The server measures the RTT

and actual throughput R_{th} of each flow and then calculates the exponential moving average E_{th} of each flow's R_{th} by $\alpha R_{th} + (1 - \alpha)E_{th}$, where α is a weighting factor. The RTT of each flow can be measured through normal TCP operation. If the server does not send any data to the client, it periodically sends probe packets to measure RTT. We then move onto operations of routers. Our proposal requires that routers in DC networks must have a special function that periodically informs the number of existing flows. There are many ways to enable this. For example, OpenFlow-enabled routers can receive a "read state" message that requests to send statistic information like the number of existing flows [17]. Another example is IP Flow Information Export (IPFIX), which enables to send statistic information like the number of flows [18]. The server identifies bottleneck links based on information about the number of existing flows received from the routers and the bandwidth of each link, and then it calculates the transmission rate which should be allocated to each flow by dividing the bandwidth of the bottleneck link by the number of competing flows. If E_{th} of each flow multiplied by a threshold factor β is smaller than the allocated transmission rate, the server calculates the transmission rate which should be reallocated to other competing flows. The reallocated transmission rate is calculated by dividing the total unutilized bandwidth of flows through the same edge router (ingress routers in DC networks) by the number of other competing flows which fully utilize the allocated transmission rate. Namely, the allocated transmission rate for each flow which cannot fully utilize the allocation is the available bandwidth fairly divided by the number of flows. On the other hand, the allocated transmission rate for each flow which fully utilizes the allocation is added to the reallocated transmission rate. These bandwidth allocation procedures are periodically performed every time the flow information about the number of competing flows and the actual throughput of each flow is updated. Namely, the allocated transmission rates converge to adequate values even when network conditions change.

To allocate transmission rates to TCP flows with window-based congestion control, the bandwidth delay product (BDP) of each flow that is used as the allocated window size should be calculated based on the bandwidth which should be allocated and the RTT of each flow. When the BDP of each flow cannot be divided by the maximum segment size (MSS), the allocated transmission rate becomes a little bit lower due to the remainder. This may degrade link utilization in DC networks. To prevent this, we consider three ways to control the allocated window size: round-up, round, and round-down methods. In each method, the BDP allocated to each flow is respectively rounded up, rounded, and rounded down by the MSS units. If the BDP allocated to each flow is rounded up by the MSS units, the total allocated transmission rate will exceed the bandwidth of bottleneck links, although this effect can be sufficiently absorbed by routers' buffer. It will also result in the convergence to the appropriate amount of allocated bandwidth.

For downwards communication from a server to a client, the server can simply allocate the calculated window size to each flow. On the other hand, for upwards communication from a client to a server, the server needs to inform the client of the allocated window size. Our scheme uses TCP's advertised window size to notify the client of the window size. This bandwidth allocation enables fair communication among flows which have different RTTs as well as high link utilization in DC networks.

4. Simulation Model

To investigate the effectiveness of our schemes, we evaluated them through simulation using Network Simulator ns-3 [19] after their implementation. We implemented functions to measure the number of competing flows and actual throughput of each flow, to identify the bottleneck links, to calculate the transmission rates allocated to each flow, and to report the flow information in this simulator.

Our scheme allocates an adequate bandwidth to each flow based on the information including RTT obtained by end-to-end measurement and the number of flows obtained from routers. We evaluate the performance of our scheme focusing on the effect of such information in this simulation. Figure 4 shows the simulation topology. In this simulation, there are two client groups A and B, and these groups have access links with different delay times. In addition, Group A consists of two groups A' and A". The clients of each group continuously send data to the server using TCP NewReno. Note that our proposed scheme works independently of TCP congestion control algorithms, so we employ TCP NewReno as a simple typical TCP. We assume that Nodes R2 and R3 are edge routers in DC networks. To realize an environment where bottleneck links exist outside DC networks, two UDP clients are also located in Group A so that TCP and UDP flows coexist on the link between Nodes R2 and R5. The propagation delay time of access links is set to a uniform random number which ranges from 2 to 6 ms for Group A or from 10 to 16 ms for Group B. The propagation delay time of other links is set to 1 ms. The bandwidth of all links is set to 100 Mb/s. Other simulation parameters are summarized in Table 1.

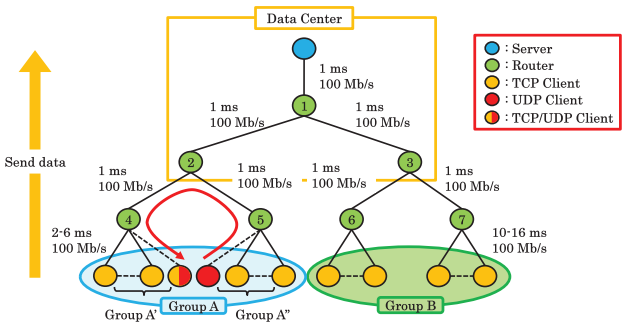
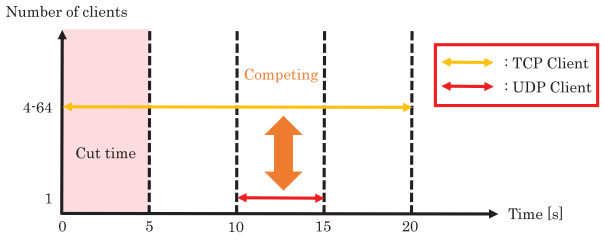


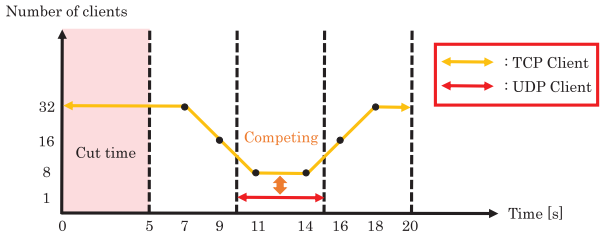
Fig. 4 Simulation model.

Table 1 Simulation parameters.

TCP algorithm	TCP NewReno
Measurement interval time for actual throughput of each flow	0.1 [s]
Notification interval time for flow information	0.005 ~ 0.5 [s]
Weighting factor α	0.1 ~ 1.0
Threshold factor β	0.1 ~ 1.0
Buffer size on routers	200 [packet]
Number of TCP clients	4 ~ 64
Number of UDP clients	2
Transmission rate of UDP client	85 [Mb/s]
Number of trials	10



(a) Fixed scenario



(b) Variable scenario

Fig. 5 Simulation scenario.

4.1 Simulation Scenario

We perform two simulation scenarios; the fixed and variable scenarios about the number of TCP clients. These scenarios are illustrated in Fig. 5. In the fixed scenario, the fixed number (4–64) of TCP clients communicate with the server after the simulation starts. In the variable scenario, 32 TCP clients communicate with the server after the simulation starts. 4 TCP clients of each group stop to communicate with the server at intervals of 2 s during the period from 7 to 11 s, and 4 TCP clients of each group start to communicate with the server at intervals of 2 s during the period from 14 to 18 s. In both scenarios, at 10 seconds after the simulation starts, one UDP client of Group A" begins to send data to one UDP client of Group A' and then stops sending at 15 seconds. We ignore the first 5 seconds.

4.2 Evaluation Indices

We evaluated Jain's fairness index [20] and total throughput for the proposed scheme and conventional TCP. The fairness index is defined as Eq. (1), where x is the throughput of each flow and n is the number of existing flows. Fairness is higher as the index gets closer to 1.

$$f(x_1, x_2, x_3, \ldots, x_n) = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n \sum_{i=1}^{n} x_i^2} \tag{1}$$

The fairness index and total throughput are calculated at intervals of 0.1 s.

4.3 Comparison Schemes

We evaluate the performance of the proposed schemes (Proposed (no reallocation) and Proposed (reallocation)) compared with that of conventional TCP (Conventional) with the averaged fairness index and total throughput. The Proposed (no reallocation) scheme allocates a fair transmission rate to each flow, while the Proposed (reallocation) scheme additionally allocates unutilized bandwidth to competing flows which fully utilizes the allocated bandwidth.

5. Simulation Results

In this section, we now show simulation results and discuss the effectiveness of the proposed schemes as compared with the conventional scheme. In the fixed scenario about the number of TCP clients, we investigate the effect of the number of TCP clients and the notification interval time for flow information in Sects. 5.1 and 5.2, respectively. In the variable scenario about the number of TCP clients, we also evaluate the effect of the weighting factor α and threshold factor β in Sects. 5.3 and 5.4, respectively.

5.1 Effect of the Number of TCP Clients

Figures 6 and 7 show the fairness index and total throughput of the proposed and conventional schemes, respectively, when the number of TCP clients varies from 4 to 64. Under the proposed (reallocation) scheme with "Round-down" method, queue length on Router 1 is also shown as functions of time when the number of flows varies from 4 to 64 in Fig. 8. Here, the notification interval time for flow information is set to 0.05 s, α is set to 0.4, and β is set to 0.8. In these figures, "Round-up", "Round", and "Round-down" represent the methods to control the allocated window size.

In Fig. 6, the proposed schemes achieve higher fairness than the conventional scheme. This is because the proposed schemes allocate an adequate window size to each flow according to its RTT. In Fig. 7, the proposed (reallocation) scheme achieves higher total throughput than the proposed (no reallocation) scheme. This is because the proposed (reallocation) scheme effectively reallocates the unutilized bandwidth from flows which cannot fully utilize their allocated bandwidth to other competing flows which can fully utilize the allocated bandwidth. Moreover, the proposed schemes with the "Round" and "Round-down" methods obtain lower total throughput than the conventional scheme,

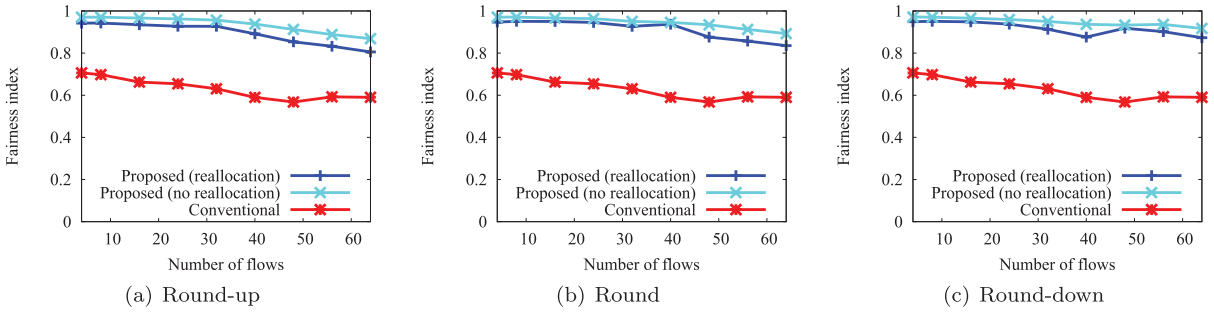


Fig. 6 Effect of the number of TCP clients: Fairness index.

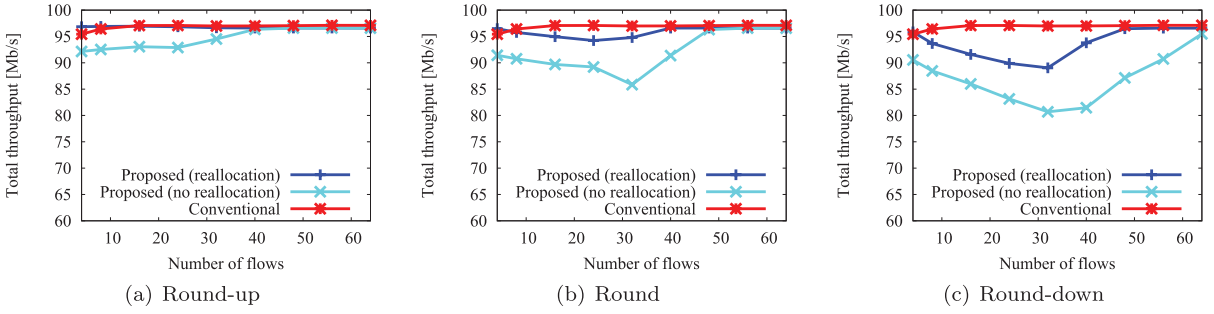


Fig. 7 Effect of the number of TCP clients: Total throughput.

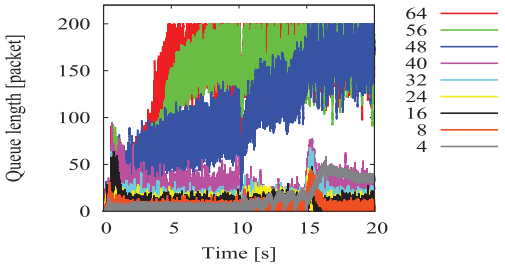


Fig. 8 Effect of the number of TCP clients: Queue length.

while the proposed (reallocation) scheme with the "Round-up" method comprehensively achieves total throughput that is higher or almost the same as for the conventional scheme regardless of the number of TCP clients. This is because the "Round" and "Round-down" methods might not allocate transmission rates satisfactorily when the BDP of each flow cannot be divided by the MSS. When the number of flows is small (4 to 32), the total throughput of the proposed schemes with "Round" and "Round-down" methods decreases as the number of flows increases. This is because these methods allocate smaller window size to each flow than BDP and the total unallocated bandwidth of all flows increases as the number of flows increases. On the other hand, when the number of flows is large (32 to 64), the total throughput of the proposed schemes with "Round" and "Round-down" methods increases as the number of flows increases. A large number of flows will cause simultaneous receiving of packets on routers, so that it will increase the queue length on routers as shown in Fig. 8. It causes longer RTT as well as larger allocated window size of each flow. As a result, the total allocated bandwidth of all flows increases as the number of flows increases.

To analyze this phenomenon more deeply, we investigate the fairness index and total throughput performance of each group. The fairness index and total throughput for each group as functions of time are respectively shown in Figs. 9 and 10 for the proposed and conventional schemes, and those for all groups are shown in Fig. 11. Here, the proposed schemes employ the "Round-up" method, the number of TCP clients is set to 32, the notification interval time for flow information is set to 0.05 s, α is set to 0.4, and β is set to 0.8. In Group A", both schemes degrade the fairness index during the period from 10 to 15 s due to a heavy UDP flow as shown in Fig. 9(b), although the proposed schemes achieve a good fairness index for other groups as shown in Fig. 9(a) and 9(c). The proposed schemes thus improve fairness in all groups as shown in Fig. 11(a). On the other hand, the proposed (no reallocation) scheme degrades the total throughput during the period from 10 to 15 s due to a heavy UDP flow, while the proposed (reallocation) scheme improves the total throughput (to almost the same as that of the conventional scheme) as shown in Fig. 11(b). Clearly, the proposed (reallocation) scheme can effectively reallocate the unutilized bandwidth in GroupA" flows to other group flows as shown in Fig. 10. Consequently, the proposed (reallocation) scheme achieves fair communication among flows which have different RTTs as well as high link utilization in DC networks.

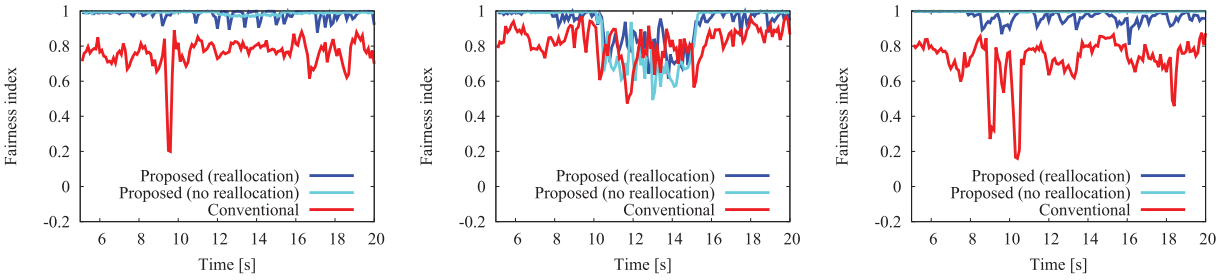


Fig. 9 Effect of the number of TCP clients: Fairness index of each group.

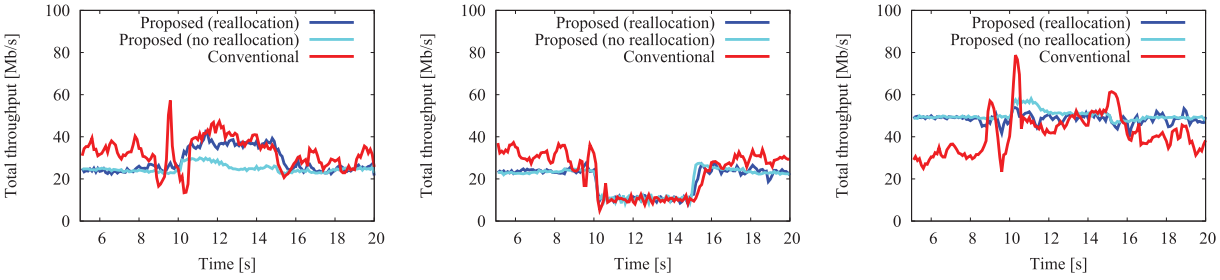


Fig. 10 Effect of the number of TCP clients: Total throughput of each group.

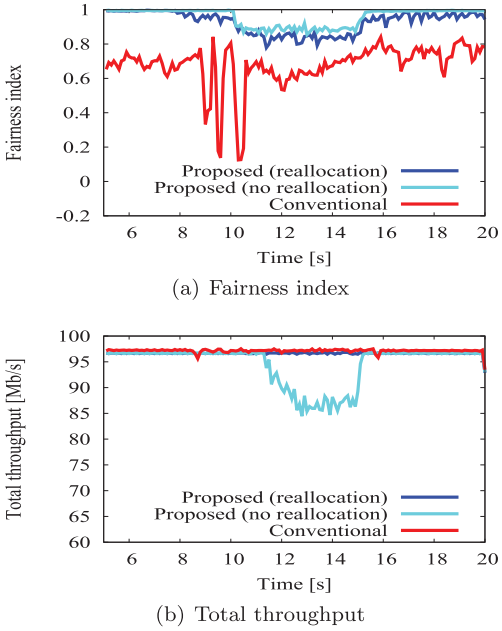


Fig. 11 Effect of the number of TCP clients: All groups.

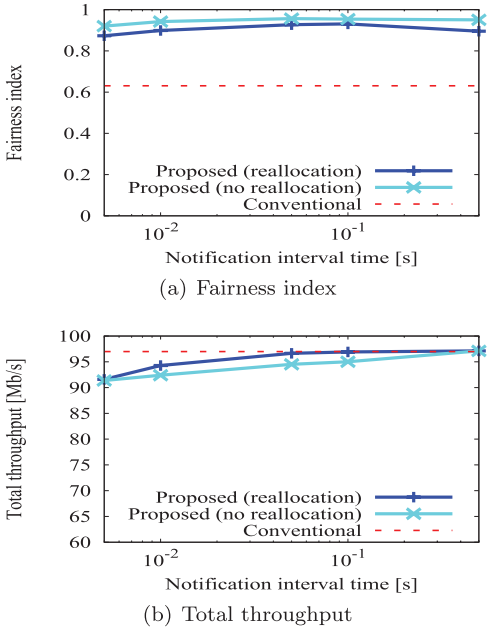
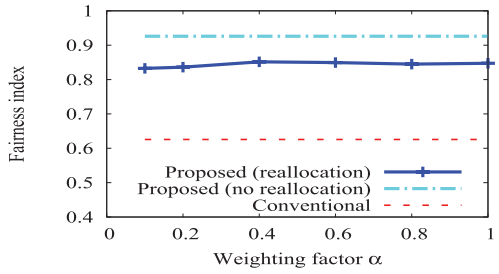


Fig. 12 Effect of notification interval time.

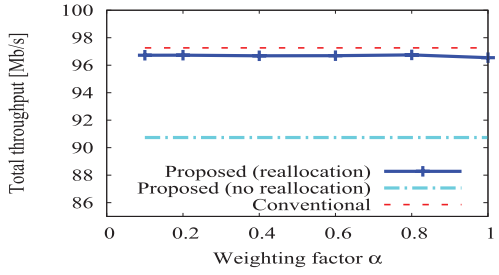
5.2 Effect of Notification Interval Time for Flow Information

Figures 12(a) and 12(b) respectively show the fairness index and total throughput of the proposed and conventional schemes when the notification interval time for flow information varies from 0.005 to 0.5 s. Here, the number of TCP clients is set to 32, α is set to 0.4, and β is set to 0.8.

In Fig. 12(a), the proposed schemes achieve a higher fairness index than the conventional scheme regardless of the notification interval time for flow information. On the other hand, the total throughput of the proposed (reallocation) scheme with the "Round-up" method is lower than that of the conventional scheme in particular when the notification interval time for flow information is from 0.005 to 0.01 s,

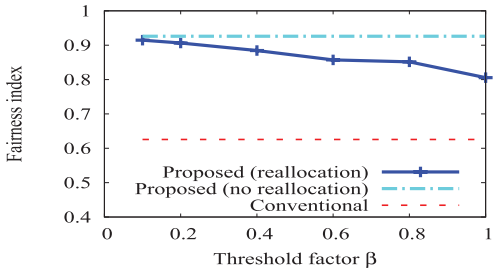


(a) Fairness index

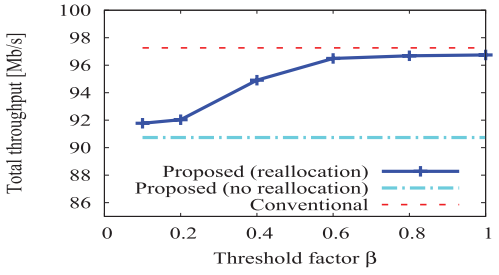


(b) Total throughput

Fig. 13 Effect of weighting factor α.



(a) Fairness index



(b) Total throughput

Fig. 14 Effect of threshold factor β.

as shown in Fig. 12(b). When the notification interval time for flow information is too short, whether the allocated bandwidth is fully utilized cannot be correctly determined since such a short notification interval time is less than the measurement interval time for the actual throughput of each flow. This causes unnecessary retransmissions as well as throughput degradation. When the notification interval time for flow information is from 0.05 to 0.5 s, the proposed (reallocation) scheme with the "Round-up" method achieves total throughput equal to that of the conventional scheme. Consequently, when the notification interval time for flow information is set to longer than 1 RTT, the proposed (reallocation) scheme with the "Round-up" method achieves good fairness and total throughput.

5.3 Effect of Weighting Factor α

Figures 13(a) and 13(b) respectively show the fairness index and total throughput for the proposed and conventional schemes when the weighting factor α varies from 0.1 to 1.0. Here, the notification interval time for flow information is set to 0.05 s and β is set to 0.8.

In Fig. 13(a), as α approaches 0, the fairness index of the proposed (reallocation) scheme slightly decreases, while the total throughput does not change. This is because the proposed (reallocation) scheme allocates transmission rates to each flow using out-of-date flow information about the actual throughput of each flow. When the flow information is too old, the proposed (reallocation) scheme will reallocate unutilized bandwidth to other flows even if the flow can fully utilize the allocated transmission rate. This degrades fairness among flows. On the other hand, as α approaches 1, the fairness index and total throughput of the proposed (reallocation) scheme decreases. This is because the allocated transmission rate of each flow is not stable. Overall, the appropriate value of α is from 0.4 to 0.6 in this simulation.

5.4 Effect of Threshold Factor β

Figures 14(a) and 14(b) respectively show the fairness index and total throughput of the proposed and conventional schemes when the threshold factor β varies from 0.1 to 1.0. Here, the notification interval time for flow information is set to 0.05 s and α is set to 0.4.

As β increases, the total throughput of the proposed (reallocation) scheme significantly increases, while the fairness index decreases. In particular, the fairness index of the proposed (reallocation) scheme is extremely degraded when β is larger than 0.8. This is because the allocated transmission rate for each flow is not stable when β is set to 1.0. Consequently, the proposed (reallocation) scheme achieves a high fairness index as well as total throughput almost the same as that of the conventional scheme when β is set from 0.6 to 0.8 in this environment.

6. Conclusion

In this study, we have proposed a bandwidth allocation scheme based on collectable information to improve fairness and link utilization in DC networks. This scheme collects flow information including the bandwidth of each link, the number of competing flows, the RTT of each flow, and the actual throughput of each flow from routers and servers in DC networks, and then it fairly allocates transmission rates among flows based on the collected information. In addition, this scheme reallocates unutilized bandwidth to other competing flows in DC networks when bottleneck links exist outside of the DC networks. Simulations indicated that the

proposed scheme enables fair communication among flows which have different RTTs as well as high link utilization in DC networks by setting appropriate parameter values.

Acknowledgments

This work was supported in part by JSPS KAKENHI Grant-in-Aid for Scientific Research (B) Number 16H02806.

References

[1] Y. Ito, H. Koga, and K. Iida, "A bandwidth allocation scheme to improve fairness in data center networks," Proc. ACM International Conference on emerging Networking EXperiments and Technologies (CoNEXT2014), pp.7–8, Sydney, Australia, Dec. 2014. DOI: 10.1145/2680821.2680834

[2] Y. Ito, H. Koga, and K. Iida, "A bandwidth allocation scheme to improve fairness in data center networks," IEICE Commun. Express, vol.5, no.5, pp.129–134, May 2016. DOI: 10.1587/comex.2016XBL0005

[3] Y. Ito, H. Koga, and K. Iida, "A bandwidth reallocation scheme to improve fairness and link utilization in data center networks," Proc. IEEE 14th International Conference on Pervasive Computing and Communications (PerCom2016), pp.65–68, Sydney, Australia, March 2016. DOI: 10.1109/PERCOMW.2016.7457064

[4] J. Zhang, F. Ren, and C. Lin, "Survey on transport control in data center networks," IEEE Netw. Mag., vol.27, no.4, pp.22–26, July/Aug. 2013. DOI: 10.1109/MNET.2013.6574661

[5] R. Meza, "A survey on congestion control," European Journal of Computer Science and Engineering, vol.10, 5 pages, 2013.

[6] P. Sreekumari and J. Jung, "Transport protocols for data center networks: A survey of issues, solutions and challenges," Photonic Network Communications, vol.31, no.1, pp.122–128, Feb. 2015. DOI: 10.1007/s11107-015-0550-y

[7] P. Prakash, A. Dixit, Y.C. Hu, and R. Kompella, "The TCP outcast problem: Exposing unfairness in data center networks," Proc. USENIX Conference on Networked Systems Design and Implementation (NSDI2012), pp.413–426, San Jose, USA, April 2012.

[8] M. Alizadeh, A. Greenberg, D.A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan, "Data center TCP (DCTCP)," ACM SIGCOMM Computer Communication Review, vol.40, no.4, pp.63–74, Oct. 2010. DOI: 10.1145/1851275.1851192

[9] J. Zhang, F. Ren, X. Yue, R. Shu, and C. Lin, "Sharing bandwidth by allocating switch buffer in data center networks," IEEE J. Sel. Areas Commun., vol.32, no.1, pp.39–51, Jan. 2014. DOI: 10.1109/JSAC.2014.140105

[10] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," IEEE/ACM Trans. Netw., vol.1, no.4, pp.397–413, Aug. 1993. DOI: 10.1109/90.251892

[11] R. Pan, B. Prabhakar, and K. Psounis, "CHOKe: A stateless active queue management scheme for approximating fair bandwidth allocation," Proc. IEEE INFOCOM2000, pp.942–951, Tel Aviv, Israel, March 2000. DOI: 10.1109/INFCOM.2000.832269

[12] S. Wen, Y. Fang, and H. Sun, "Differentiated bandwidth allocation with TCP protection in core routers," IEEE Trans. Parallel Distrib. Syst., vol.20, no.1, pp.34–47, Jan. 2009. DOI: 10.1109/TPDS.2008.71

[13] L. Lu, H. Du, and R.P. Liu, "CHOKeR: A novel AQM algorithm with proportional bandwidth allocation and TCP protection," IEEE Trans. Ind. Informat., vol.10, no.1, pp.637–644, Feb. 2014. DOI: 10.1109/TII.2013.2278618

[14] D. Katabi, M. Handley, and C. Rohrs, "Congestion control for high bandwidth-delay product networks," ACM SIGCOMM Computer Communication Review, vol.32, no.4, pp.89–102, Oct. 2002. DOI: 10.1145/964725.633035

[15] L. Kalampoukas, A. Varma, and K.K. Ramakrishnan, "Explicit window adaptation: A method to enhance TCP performance," IEEE/ACM Trans. Netw., vol.10, no.3, pp.338–350, June 2002. DOI: 10.1109/TNET.2002.1012366

[16] K. Kusuhata, I. Kaneko, K. Iida, H. Koga, and M. Shimamura, "A Unified congestion control architecture design to improve heterogeneous wireless network efficiency and accommodate traffic by various rich applications," Proc. IFIP HET-NETs2013, 5 pages, Ilkley, UK, Nov. 2013.

[17] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling innovation in campus networks," ACM SIGCOMM Computer Communication Review, vol.38, no.2, pp.69–74, April 2008. DOI: 10.1145/1355734.1355746

[18] P. Aitken, B. Claise, and B. Trammell, "Specification of the IP flow information export (IPFIX) protocol for the exchange of flow information," IETF RFC7011, Sept. 2013.

[19] The ns-3 network simulator, http://www.nsnam.org/

[20] R. Jain, D.M. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared systems," DEC Research Report TR-301, Sept. 1984.

Yusuke Ito received the B.E. and M.E. degrees in Information and Media Engineering from the University of Kitakyushu, Japan in 2013, 2015, respectively. Presently, he is a doctoral student at the University of Kitakyushu, Japan. His research interests include network architecture, cloud computing, and mobile edge computing.

Hiroyuki Koga received the B.E., M.E. and D.E. degrees in computer science and electronics from Kyushu Institute of Technology, Japan, in 1998, 2000, and 2003, respectively. From 2003 to 2004, he was a postdoctoral researcher in the Graduate School of Information Science, Nara Institute of Science and Technology. From 2004 to 2006, he was a researcher in the Kitakyushu JGN2 Research Center, National Institute of Information and Communications Technology. From 2006 to 2009, he was an assistant professor in the Department of Information and Media Engineering, Faculty of Environmental Engineering, University of Kitakyushu, and then has been an associate professor in the same department since April 2009. His research interests include performance evaluation of computer networks, mobile networks, and communication protocols. He is a member of the ACM and IEEE.

Katsuyoshi Iida received the B.E., M.E. and Ph.D. degrees in Computer Science and Systems Engineering from Kyushu Institute of Technology (KIT), Iizuka, Japan in 1996, in Information Science from Nara Institute of Science and Technology, Ikoma, Japan in 1998, and in Computer Science and Systems Engineering from KIT in 2001, respectively. Currently, he is an Associate Professor in the Information Initiative Center, Hokkaido University, Sapporo, Japan. His research interests include network systems engineering such as network architecture, performance evaluation, QoS, and mobile networks. He is a member of the WIDE project and IEEE. He received the 18th TELECOM System Technology Award, and Tokyo Tech young researcher's award in 2003, and 2010, respectively.