IEICE Communications Express, Vol.13, No.6, 222–226

ComEX

LETTER

Fairness improvement method using explicit congestion notification for QUIC

Nobuhiro Uchida[1, a)], **Daiki Nobayashi**[2, b)], **Dirceu Cavendish**[2, c)], and **Takeshi Ikenaga**[2, d)]

Abstract QUIC is a transport protocol that adds congestion control, retransmission control, and TLS to UDP. QUIC can use the same congestion control algorithms as TCP. In previous work, we have shown that communication performance becomes unfair concerning the buffer size of the shared bottleneck link of two flows, one using CUBIC and the other BBR congestion control algorithms within QUIC. In this study, we improve the communication fairness performance at different bottleneck link buffer sizes by using Round Trip Time (RTT) and Explicit Congestion Notification (ECN) to regulate congestion control aggressiveness when CUBIC and BBR compete within QUIC through actual experiments.

Keywords: QUIC, congestion control, BBR, CUBIC, ECN
Classification: Network

1. Introduction

QUIC is a new transport protocol standardized by the Internet Engineering Task Force (IETF) in 2021 [1]. QUIC, unlike TCP, doesn't require support from an operating system or kernel and introduces congestion control, retransmission control, and TLS over UDP to transfer data.

On the other hand, researchers have been improving congestion control algorithms used in transport protocols to improve communication performance. BBR is a new congestion control algorithm developed by Google in 2016. BBR operates with inflight packets close to the optimal point proposed by Leonard Kleinrock [2].

QUIC can use the same congestion control algorithms as TCP. In previous work, communication performance was evaluated by the actual environment when both CUBIC and BBR flows compete within QUIC. Our results showed that communication performance became unfair with the buffer size of the shared bottleneck link [3], similar to TCP fairness behavior [4]. In other previous work, we improved communication performance at different bottleneck link buffer sizes using ECN and measured RTT when CUBIC and BBR compete within QUIC communication through actual experiments [5]. In this paper, we examine whether our proposed method is effective in cases where it conflicts with existing methods and in environments with more than two flows.

The remainder of this paper is as follows. Section 2 includes related works. Section 3 and Section 4 presents our proposed method and experimental environment, respectively. In Section 5, the results of the evaluation are presented and discussed. In Section 6, we discuss the fairness between CUBIC and BBR. Finally, we conclude the paper in Section 7.

2. Related works

Various studies have been conducted on congestion control competition in the transport layer. Y. Han et al. [6] focus on fairness in the same congestion control in QUIC. When multiple BBR share a bottleneck link, excessive packet loss occurs, which will cause a loss of fairness. In that study, packet loss and queuing status are added to the determinants of BBR inflight packets to improve fairness.

Fairness is low when CUBIC and BBR share a bottleneck link in TCP. Katumata et al. [7] measure RTT using ICMP in user space, and improve fairness by passing accurate RTT values to BBR. Y.-J. Song et al. [4, 8] propose a method of decreasing *cwnd* to improve the congestion situation.

Ranysha Ware et al. [9] note that when TCP BBR competes with loss-based TCP, TCP BBR occupies the bandwidth regardless of the number of loss-based flows. In this study, one flow of TCP BBR competes with 16 flows of CUBIC TCP, indicating that TCP BBR occupies the bandwidth.

Satoshi Utsumi et al. [10] devised a model for estimating the average packet transmission rate of CUBIC when CUBIC and BBR compete on a bottleneck link and improved fairness.

3. Proposed method

When CUBIC and BBR compete in QUIC, the communication performance of CUBIC is low when the buffer size of the bottleneck link is small, and the performance of BBR is low when that of the bottleneck link is large. In this study, we aim to improve the fairness of CUBIC and BBR in QUIC by judging the approximate size of the buffer size from the amount of increase in RTT during congestion, in addition to detecting the congestion condition by ECN. As a premise, a router marks packets with ECN when detecting congestion.

[1] Graduate School of Engineering, Kyushu Institute of Technology 1-1 Sensui-cho, Tobata-ku, Kitakyushushi, Fukuoka, 804-8550 Japan
[2] Faculty of Engineering, Kyushu Institute of Technology 1-1 Sensui-cho, Tobata-ku, Kitakyushushi, Fukuoka, 804-8550 Japan
a) uchida.nobuhiro682@mail.kyutech.jp
b) nova@ecs.kyutech.ac.jp
c) cavendish@net.ecs.kyutech.ac.jp
d) ike@ecs.kyutech.ac.jp

DOI: 10.23919/comex.2024XBL0014
Received January 18, 2024
Accepted March 8, 2024
Publicized April 8, 2024
Copyedited June 1, 2024

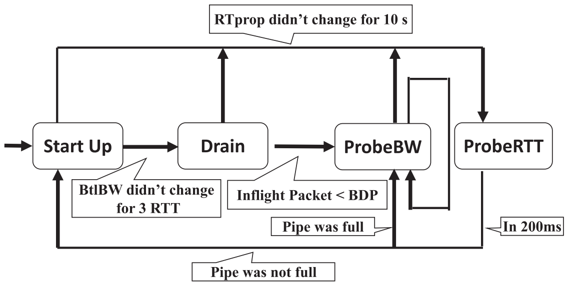


Fig. 1 Statement of BBR

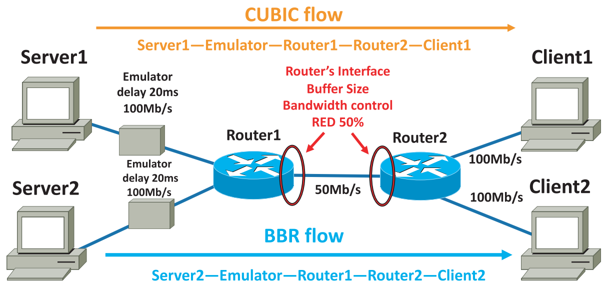


Fig. 2 Experimental testbed environment

3.1 Proposed method for CUBIC

TCP CUBIC is a Loss-based TCP that has achieved widespread usage as the default TCP of the Linux operating system. During congestion avoidance, its congestion window [11].

Proposed CUBIC introduces the same behavior as TCP: receiving an ECN reduces congestion window (*cwnd*) in the same manner as duplicate ACKs. However, this ECN window reduction is performed only if RTT is at least 1.5 times the minimum RTT, otherwise CUBIC ignores ECNs. When RTT is more than 1.5 times the minimum RTT, CUBIC determines buffer size to be large, decreasing *cwnd* as follows:

$$cwnd_{cubic} = C \cdot (t - K)^3 + cwnd_{current}, \quad (1)$$

$$K = \sqrt[3]{\frac{W_{max} \cdot (1 - \beta)}{C}}, \quad (2)$$

$$cwnd = \max(cwnd_{cubic}, \frac{cwnd_{current}}{2}). \quad (3)$$

Here C is a constant to determine the increase of *cwnd* in a high BDP network, which is usually 0.4. t is the elapsed time since congestion avoidance. β is the reduction factor of *cwnd* of CUBIC, and in this study, the value of β is 7/8 by adopting the specified value in the implementation used for evaluation. $cwnd_{current}$ is current *cwnd*. First, The *cwnd* of CUBIC is calculated from the Eq. (1) and Eq. (2). Finally, from the Eq. (3), CUBIC compares and determines the value of *cwnd*. Thus, when the buffer size of the bottleneck link is large, CUBIC prevents packets from accumulating in the buffer and consuming bandwidth.

3.2 Proposed method for BBR

BBR has four phases, StartUp, Drain, ProbeBW, and ProbeRTT, as shown in Fig. 1 [12]. In StartUp and ProbeBW, we change phase transition conditions and gain based on ECN and RTT as follows.

When the RTT is less than 1.5 times the minimum RTT in StartUp and BBR receives ECN, BBR enters Drain. BBR determines bottleneck link buffer size to be small, entering Drain to prevent consuming bandwidth.

In probeBW, BBR counts the number of ECN packets and determines the *gain* based on RTT and the number of ECN packets in the previous ProbeBW when entering the next ProbeBW. When RTT is more than 1.5 times the minimum RTT, BBR determines buffer size to be large, and multiplies the *gain* by 1.25, so preventing CUBIC from consuming bandwidth. When RTT is less than 1.5 times the minimum RTT and the number of ECN packets is more than a certain value, BBR determines buffer size to be small, and multiplies the *gain* by 0.75 to prevent from consuming bandwidth. The values of 0.75 times and 1.25 times are the values of the *gain* cycle in ProbeBW.

4. Experimental testbed

In this work, we demonstrate the effectiveness of our proposed method using an experimental testbed environment, as per Fig. 2. Servers and clients are desktop PCs, equipped with either Intel Core-i7 or Intel Core-i9 processors, with 16 GB of memory. Both servers and clients run Debian Linux version 10 OS. An IP network emulator NXS7000F (Fujitsu) is used to adjust path delays and configure router interfaces with bandwidth capacity, buffer size, and Random Early Detection (RED) for marking ECNs. In all experiments, RED is configured to not drop packets, but only mark ECN if packets exceed 50% the interface buffer size. PICOQUIC [13] was used to implement QUIC communication. Link bandwidth is tuned as per Fig. 2, with a bottleneck link between router 1 and router 2. The resulting bottleneck link delay is adjusted to 20 ms using the emulators between server 1 and router 1 and between server 2 and router 1. The resulting RTT between server 1 and client 1 and between server 2 and client 2 then becomes approximately 40ms. Note that in our experiments, 1 BDP corresponds to 1 Mbit of data. We have devised the following scenarios: two flow experiment, with three bottleneck buffer size scenarios of 1 BDP, 3 BDP, and 6 BDP; in four flow experiments, with three bottleneck buffer size scenarios of 1 BDP, 6 BDP, and 10 BDP. We have chosen these scenarios because the throughput of CUBIC and BBR in the existing QUIC is fair in 3 BDP for two flows and 6 BDP for four flows. We have also created three bottleneck buffer sizes. We evaluate the performance of file downloads of 400 MB size per scenario from the server to the client. The number of trials in each experimental data point collected was five. The performance measure used was throughput per second when two or four flows, CUBIC and BBR, were generated simultaneously.

5. Experimental results

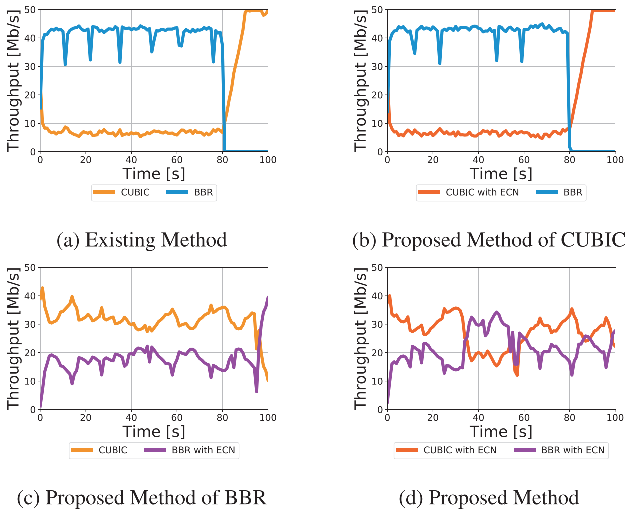
In this section, we show experimental results for several bottleneck link buffer sizes, ranging from 1 to 6 BDP worth of data for two flow experiments, and from 1 to 10 BDP worth of data for four flow experiments.

5.1 Two flow performance

Figure 3(a) shows flow throughput performance of regular CUBIC and BBR, Fig 3(b) shows the flow throughput performance of CUBIC with ECN change proposed and regular BBR, Fig. 3(c) shows the flow throughput performance of regular CUBIC and BBR with ECN change proposed, and finally Fig. 3(d) shows the flow throughput of CUBIC with ECN change and BBR with ECN change proposed for the elapsed time, when the buffer size of the bottleneck link is 1 BDP. BBR consumes more bandwidth than CUBIC in Fig. 3(a) and Fig. 3(b) while throughput of BBR flow decreases and that of CUBIC increases in Fig. 3(c) and Fig. 3(d). That is because BBR with ECN immediately moves to Drain via ECN notification in StartUp, and BBR with ECN multiplies its gain by 0.75 based on RTT and the number of ECN packets in probeBW.

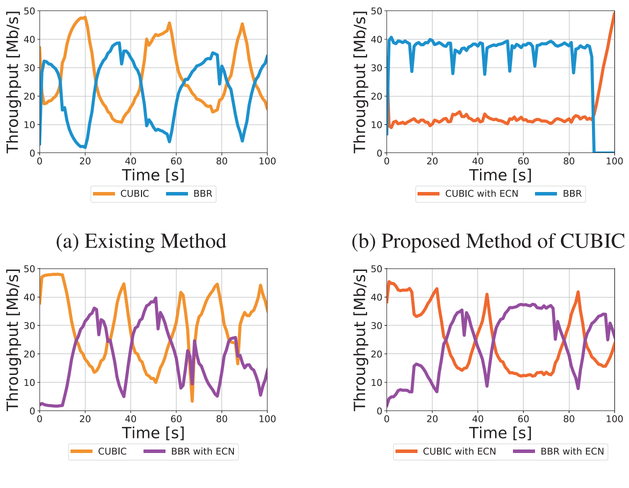
Figure 4(a) shows the flow throughput of CUBIC and BBR, and Fig. 4(b) shows the flow throughput of CUBIC with ECN change proposed and BBR, Fig. 4(c) shows the flow throughput of CUBIC and BBR with ECN change proposed, and Fig. 4(d) shows the flow throughput of CUBIC with ECN change proposed and BBR with ECN change proposed, for the elapsed time when the buffer size of the bottleneck link is 3 BDP. Fairness becomes apparent in Fig. 4(a), Fig. 4(c), and Fig. 4(d). However, BBR consumes more bandwidth than CUBIC in Fig. 4(b) because CUBIC with ECN receives ECN and acts as duplicate ACKs, increasing retransmissions.

Figure 5(a) shows the flow throughput of regular CUBIC and BBR, Fig. 5(b) shows the flow throughput of CUBIC with ECN enhancement proposed and regular BBR, Fig. 5(c) shows flow throughput of CUBIC and BBR with ECN enhancement, and Fig. 5(d) shows the flow throughput of CUBIC with ECN enhancement and BBR with ECN enhancement for the elapsed time when the buffer size of the bottleneck link is 6 BDP. Fig. 5(a) shows that CUBIC con-



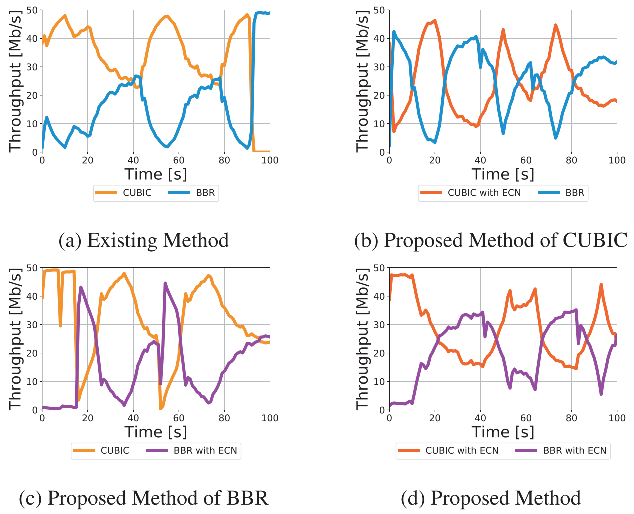
(a) Existing Method
(b) Proposed Method of CUBIC
(c) Proposed Method of BBR
(d) Proposed Method

Fig. 3 Throughput - 1 BDP



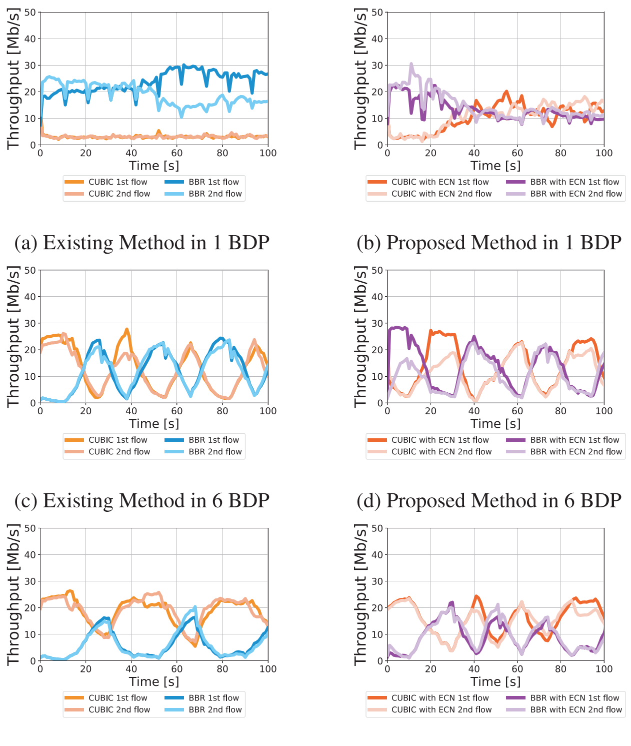
(a) Existing Method
(b) Proposed Method of CUBIC
(c) Proposed Method of BBR
(d) Proposed Method

Fig. 4 Throughput - 3 BDP



(a) Existing Method
(b) Proposed Method of CUBIC
(c) Proposed Method of BBR
(d) Proposed Method

Fig. 5 Throughput - 6 BDP



(a) Existing Method in 1 BDP
(b) Proposed Method in 1 BDP
(c) Existing Method in 6 BDP
(d) Proposed Method in 6 BDP
(e) Existing Method in 10 BDP
(f) Proposed Method in 10 BDP

Fig. 6 Throughput in four flow

sumes more bandwidth than BBR, while flow throughput of CUBIC decreases and that of BBR increases in Fig. 5(b), Fig. 5(c), and Fig. 5(d). That is because CUBIC with ECN receives ECN and acts as duplicate ACKs in Fig. 5(b) and Fig. 5(d), whereas BBR with ECN multiplies its gain by 1.25 based on RTT and the number of ECN packets in probeBW in Fig. 5(c) and Fig. 5(d).

5.2 Four flow performance

Figure 6(a) and Fig. 6(b) show the flow throughput performance of CUBIC and BBR for the elapsed time when the buffer size of the bottleneck link is 1 BDP, in the existing and proposed methods, respectively. We verify that BBR consumes more bandwidth than CUBIC in the existing method, while the flow throughput of BBR with ECN decreases and that of CUBIC with ECN increases in the proposed method. That is because BBR with ECN multiplies its gain by 0.75 based on RTT and the number of ECN packets in probeBW.

Figure 6(c) and Fig. 6(d) show the flow throughput of CUBIC and that of BBR for the elapsed time when the buffer size of the bottleneck link is 6 BDP in the existing and proposed methods respectively. In this scenario, the existing method and the proposed method are both fair.

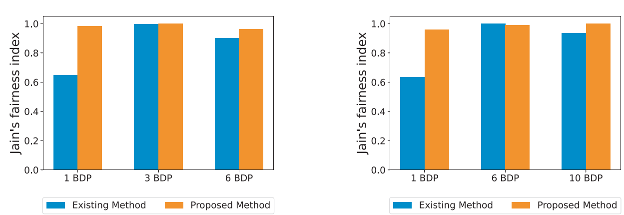
Figure 6(e) and Fig. 6(f) show the flow throughput of CUBIC and that of BBR for the elapsed time when the buffer size of the bottleneck link is 10 BDP, in the existing and proposed methods, respectively. We verify that CUBIC consumes more bandwidth than BBR in the existing method, while the flow throughput of CUBIC with ECN decreases, and that of BBR with ECN increases in the proposed method. That is because CUBIC with ECN receives ECN and acts as duplicate ACKs, whereas BBR with ECN multiplies its gain by 1.25 based on RTT and the number of ECN packets in probeBW.

6. Discussion

To precisely quantify fairness improvements of our proposed method across different experiments, we use Jain's fairness index [14] as a measure of fairness between CUBIC and BBR.

$$Fairness\ Index = \frac{(\sum_{i=1}^{n} x_i)^2}{(n \times \sum_{i=1}^{n} x_i^2)}, \qquad (4)$$

where n is the number of senders, and x_i is the throughput of the i-th flow. We focus on the throughput from 20 s to 70 s to evaluate fairness in steady-state conditions. Figure 7 shows the fairness of CUBIC and BBR for two and four flow scenarios. Fairness improves consistently across all two flow and four flow scenarios, demonstrating the robustness of our proposed method.



(a) Two flow

(b) Four flow

Fig. 7 Jain's fairness index

7. Conclusion

In this work, we have proposed new ECN based methods for controlling congestion window of CUBIC and BBR aiming at improving fairness between them within QUIC. We have verified that our proposed methods are effective or do not impact throughput in many cases, even when the existing methods and the proposed methods compete among themselves. As future work, we plan to demonstrate fairness improvements using BBR version 2 (BBRv2) or other congestion control algorithms within QUIC.

Acknowledgments

This work was supported in part by the Strategic Information and Communications R&D Promotion Programme (SCOPE) funded by Ministry of Internal Affairs and Communication (MIC), Japan, and JSPS KAKENHI Grant Number 21H03430.

References

[1] J. Iyengar and M. Thomson, "QUIC: A UDP-based multiplexed and secure transport," IETF RFC 9000, May 2021. DOI: 10.17487/rfc9000

[2] R. Gail, et al., "An invarient property of computer network power," IEEE International Conference on Communications, 63.1.1-63.1.5, 1981.

[3] N. Uchida, D. Nobayashi, T. Ikenaga, and D. Cavendish, "Performance evaluation on the impact of bottleneck link buffer size under differnt congestion control algorithm competition in QUIC," Proc. IEEE CCNC, pp. 714–715, Jan. 2023. DOI: 10.1109/ccnc51644.2023.10060682

[4] Y.-J. Song, G.-H. Kim, and Y.-Z. Cho, "Improvement of cyclic performance variation between TCP BBR and CUBIC," IEEE Asia-Pacific Conference on Communications (APCC), pp. 1–6, Nov. 2019. DOI: 10.1109/apcc47188.2019.9026471

[5] N. Uchida, D. Nobayashi, D. Cavendish, and T. Ikenaga, "Poster: Fairness improvement method using ECNs with different congestion control algorithms within QUIC," Proc. IEEE ICNP, pp. 1–2, Oct. 2023. DOI: 10.1109/icnp59255.2023.10355640

[6] Y. Han, M. Zuo, H. Yuan, Y. Zhong, Z. Yuan, and T. Bi, "A Qos-based fairness-aware BBR congestion control algorithm using QUIC," Wireless Communications and Mobile Computing, vol. 2022, pp. 1–16, April 2022. DOI: 10.1155/2022/7222030

[7] N. Katumata, et al., "TCP fairness improvement by congestion control from user space," International Conference on Engineering Technology for Communications, Dec. 2021.

[8] Y.-J. Song, G.-H. Kim, and Y.-Z. Cho, "Congestion window scaling method for inter-protocol fairness of BBR," IEEE Anual Consumer Communications and Networking Conference (CCNC), pp. 1–4, March 2020. DOI: 10.1109/ccnc46108.2020.9045307

[9] R. Ware, M.K. Mukerjee, S. Seshan, and J. Sherry, "Modeling BBR's interactioins with loss-based congestion control," IMC Proceedings of the Internet Measurment Conference, pp. 137–143, Oct. 2019. DOI: 10.1145/3355369.3355604

[10] S. Utsumi and G. Hasegawa, "Improving inter-protocol fairness based on estimated behavior of competing flows," IFIP Networking Conference, pp. 1–9, 2022. DOI: 10.23919/ifipnetworking55013.2022.9829780

[11] I. Rhee, L. Xu, S. Ha, A. Zimmermann, L. Eggert, and R. Scheffenegger, "CUBIC for fast long-distance networks," IETF RFC 8312, Feb. 2018. DOI: 10.17487/rfc8312

[12] N. Cardwell, et al., "BBR congestion control," IETF draft draft-

cardwell-iccrg-bbr-congestion-control-00, July 2017.
[13] PICOQUIC, https://github.com/private-octopus/picoquic
[14] K.R. Jain, et al., “a quantitative measure of fairness and discrimination,” Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA, Tech. Req., vol. 21, 1984.