TCP Fairness among Modern TCP Congestion Control Algorithms including TCP BBR

Kanon Sasaki
Department of Information and Communications Engineering
Kogakuin University
Tokyo, Japan
c515046@ns.kogakuin.ac.jp

Kouto Miyazawa
Department of Information and Communications Engineering
Kogakuin University
Tokyo, Japan
c515102@ns.kogakuin.ac.jp

Naoki Oda
Electrical Engineering and Electronics
Kogakuin University Graduate School
Tokyo Japan
cm17012@ns.kogakuin.ac.jp

Masato Hanai
Electrical Engineering and Electronics
Kogakuin University Graduate School
Tokyo Japan
cm16036@ns.kogakuin.ac.jp

Aki Kobayashi
Electrical Engineering and Electronics
Kogakuin University Graduate School
Tokyo Japan
aki@cc.kogakuin.ac.jp

Saneyasu Yamaguchi
Electrical Engineering and Electronics
Kogakuin University Graduate School
Tokyo Japan
sane@cc.kogakuin.ac.jp

***Abstract*—For improving communication performance, many fast TCP algorithms, e.g. CUBIC TCP and Compound TCP, have been proposed. These proposals have raised another issue that is performance fairness among TCP congestion control algorithms. Several papers were published for discussing the fairness and some of them revealed their unfairness. In addition, some fairness improving methods based on packet dropping, for example applying CoDel, have been proposed. However, these existing works were based on the existing TCP algorithms, which were loss-based, delay-based, or hybrid type TCP congestion control algorithm. In 2016, another TCP congestion control algorithm, called TCP BBR, was proposed. The algorithm is based on the congestion model by Kleinrock and not loss-based or delay-based. In this paper, we investigate the performance fairness between CUBIC TCP and TCP BBR. We present performance evaluation in conditions wherein connections of TCP BBR and CUBIC TCP are concurrently established. We then demonstrate that the performance fairness between TCP BBR and CUBIC TCP is very low, especially with high latency. In the case of 64 ms RTT, TCP BBR obtained about 45 times of performance of CUBIC TCP. We applied CoDel for improving TCP fairness and evaluated the fairness. Our evaluation showed that it did not work well in case of TCP BBR consumes much bandwidth. On the other hand, it is effective in case of TCP BBR cannot obtain enough throughput.**

Keywords—TCP, congestion control algorithm, TCP BBR, CUBIC TCP, TCP fairness

I. Introduction

Several fast TCP congestion control algorithms, such as CUBIC TCP [1] and Compound TCP [2], have been proposed. Obtained throughput depends on TCP congestion control algorithm. Thus, these proposals raised an issue of performance fairness among TCP congestion control algorithms, which is called *TCP fairness* [2].

Several works were studied for discussing TCP fairness, for example, fairness evaluation [3] on TCP Vegas [4] and those [5][6][7][8] on CUBIC TCP. These works then pointed unfairness out. In addition, some methods, such as applying RED [9], applying CoDel [7][10][11][12], and Active Packet Dropping [13], were proposed for improving the fairness by dropping packets of the bandwidth-consuming flow [5]. These improving methods effectively utilize AIMD (additive increase multiplicative decrease), with which dropping a single packet can remarkably decrease the throughput of the target connection.

However, these existing works were based on existing TCP congestion control algorithms, which are loss-based [1][14], delay-based [4], or hybrid [2] algorithms. In 2016, a new TCP congestion control algorithm called TCP BBR [15] was proposed. That is based on not these model but Kleinrock's congestion model. Therefore, a new profound study on the performance fairness between TCP BBR and other TCP algorithms and effect of methods for improving fairness based on dropping packets is required. In this paper, we investigate the TCP fairness among TCP BBR and CUBIC TCP. We then discuss a method for improving the fairness.

II. Related Work

A. CUBIC TCP

CUBIC TCP [1] is an algorithm based on BIC TCP [16], so it has high scalability. Moreover, it has better TCP fairness and RTT fairness [17][18] than BIC TCP. RTT fairness is performance fairness among connections with different RTTs. CUBIC TCP uses the following cubic function, while BIC TCP uses binary search.

$$cwnd = C(t-K)^3+W_{max} \quad (1)$$

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

cwnd is congestion window size, *t* is the time from the last packet loss, *Wmax* is the congestion window size at the last loss, *C* and *β* are parameters for speed adjustment in the usual state and dropping ratio at packet losses, respectively. The larger *C* results in the faster increase. In most cases of recent Linux operating systems, *C* is 0.4 and *β* is 0.3.

B. TCP BBR

TCP BBR, which stands for bottleneck bandwidth and round-trip propagation time, is a TCP algorithm that was newly proposed by Cardwell et al. in 2016 [15].

978-1-5386-6831-3/18/$31.00 ©2018 IEEE

It has been widely recognized that bandwidth-delay product (BDP) is the ideal value of TCP window size. However, many TCP algorithms, including popularly used CUBIC TCP and TCP Reno, do not aim BDP. They increase their congestion window size until packet losses and then often exceed the BDP on purpose. The authors focused on two physical constraints that bound transport performance. The two constraints are round-trip propagation (*RTprop*) and bottleneck bandwidth (*BtlBw*). The TCP BBR estimates these two values during its communication. It then calculates the BDP, which is *RTprop***BtlBw*, according to these estimated values and set its congestion window size to this estimated BDP.

On the other hand, popular loss-based TCP algorithm such as TCP Reno and CUBIC TCP increase its congestion window size, more than the optimal size ,i.e. BDP. Such algorithms then cause network congestion and packet losses on purpose. This causes also an unnecessary increase of RTT. This unnecessarily large RTT is pointed out by Cerf et al. [19][20] as *Bufferfloat*.

In the paper, they compared performances of CUBIC TCP and TCP BBR, and stated that TCP BBR can obtain performance of 2 to 10 times higher than CUBIC TCP. In addition, they also evaluate performance fairness among multiple TCP BBR connections sharing a bottleneck link and stated that the fairness of communication will improve as RTprop and BtlBw are investigated. However, fairness between TCP BBR and other TCP algorithms, e.g. TCP Reno and CUBIC TCP, are nod discussed.

C. CoDel (Controlling Queue Delay)

Nichols et al. proposed a new packet queue scheduling algorithm, called CoDel [10]. This drops packets when queue delay time, which is time length between time at enqueue and dequeue of a packet, exceeds the threshold time, which is called *target*. The threshold is 5 ms with the default setup.

Unlike RED, CoDel has only one parameter, i.e. target, and dropping is determined only by the queueing delay time. CoDel is expected to be popular because it is easy to tune.

D. TCP Fairness

The following works are on TCP fairness without TCP BBR. Mo et al. and Hasegawa et al. showed the performance unfairness between TCP Reno and TCP Vegas [21][22]. The fairness between Compound TCP and CUBIC TCP are described in several works of [5][6][8][23]. These works showed that the performance fairness is severaly low.

The following works are on TCP fairness with TCP BBR. Hock et al. evaluated the fairness between CUBIC TCP and TCP BBR [24]. They then showed that CUBIC TCP and TCP BBR outperform with large and small buffers, respectively. They pointed it out that CUBIC increases the queue length longer than the optimal point. Li et al. compared the performances of CUBIC TCP and TCP BBR over 4G Long Term Evolution (LTE) networks [25]. They then concluded that TCP BBR significantly reduced the delays than CUBIC. However, these works did not mention the methods for improving fairness by dropping packets.

The following are works on improvement of TCP fairness. Itsumi et al. proposed a method for improving TCP fairness by dropping packets [5]. The method monitors queue length in the bottleneck router and drops packets at detection of sharp grows of the queue length. The work of [13] evinced that the

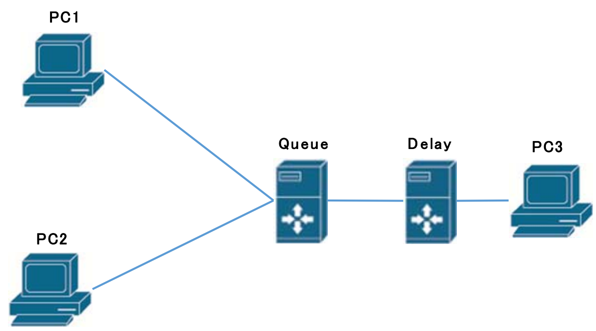

Fig. 1. Experimental network

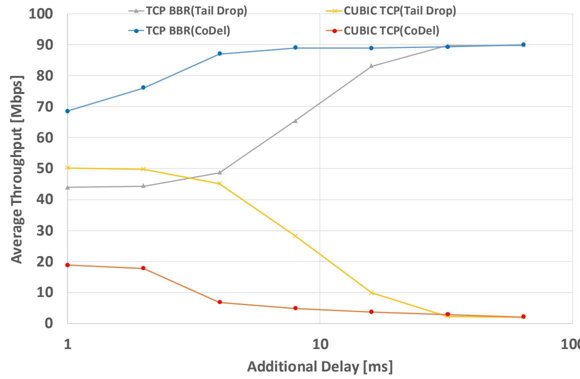

Fig. 2. BBR and CUBIC THroughput

TCP fairness is increased by applying RED. The work of [7][26] does that the fairness is increased by applying CoDel. These works effectively utilized the feature of AIMD. Namely, the most of existing TCP algorithms significantly decline their congestion window size at a detection of a packet loss. Therefore, these methods can largely decrease the throughput of the most bandwidth-consuming flow by dropping a packet. For advancing these methods, Active packet dropping [13] and dynamically Controlling queue delay [7][11] were proposed. These advanced methods discard packets in bandwidth-consuming flows more aggressively.

III. TCP Fairness Evaluation

In this section, we evaluate the performance fairness between TCP BBR and CUBIC TCP with and without packet droping by CoDel using actual TCP implementations and actual network elements.

A. Experimental setup

We constructed the experimental network illustrated in Fig. 1 and evaluated the TCP fairness using iperf [27] in environments wherein connections of TCP BBR and CUBIC TCP coexist. PC1 and PC2 in the figure are sending machines. PC3 is the receiving machine. Communications of PC1-PC3 and PC2-PC3 are performed simultaneously. Both of the communications share the network elements between Queue and PC3. Queue is a queueing element for the bottleneck link. Queue manages its packet queue with TailDrop or CoDel. Delay emulates a network latency using Linux netem. All the elements support Gigabit Ethernet. The advertised window sizes of sending and receiving machines are 32 MB. Ten connections are established in each path. The total number of connections is 20. The buffer size, which is the parameter limit in netem, in the Queue is 1000, which is the default size.

B. Experimental Results

In this subsection, we evaluate the performance fairness in an environment wherein communications with TCP BBR and CUBIC TCP competitively coexist. The operating systems of The operating systems of PC1 and PC2 are Fedora21 with Linux 4.13.2. PC1 and PC2 send data using TCP BBR and CUBIC TCP, respectively. The operating system of PC3 is Fedora24 with Linux 3.17.4. Ten TCP BBR connections between PC1 and PC3 were established. Ten CUBIC TCP connections also were established simultaneously between PC2-PC3.

Figure 2 shows the experimental results with Tail Drop and CoDel in the Queue machine. The horizontal axis in the figure represents the delay time in the round-trip artificially added on the Delay machine. We refer it to *additional delay*. The vertical axis of the graph shows the average throughputs of connections of each TCP algorithm.

The results show that the throughput of TCP BBR increases and that of CUBIC TCP decreases as the additional delay increases regardless of the algorithm of queue management in the Queue machine. In the case of the additional delay is 64 ms, TCP BBR consumes almost all the bandwidth and the throughput of CUBIC TCP is remarkably less, which is less than one-tenth, with either queue management algorithm. Comparing the results of Tail Drop and CoDel, we can see the similar results and trends.

In spite of the fairness between CUBIC TCP and Compound TCP was improved by applying CoDel [7], the fairness was not improved in this experiment. CoDel improved the TCP BBR performance significantly and declined the fairness more.

From these results, we can conclude that the fairness is severe with high latency and the existing fairness improving method by dropping packets does not work well in case of TCP BBR outperforms.

IV. Discussion

We discuss the effect of methods to improve fairness by dropping packets. The traditional congestion control algorithms largely decrease their congestion window sizes at a detection of packet loss. This is called AIMD as mentioned above. Based on this behavior, several methods to improve fairness by dropping a small number of packets [5], such as applying CoDel [7][26], have been proposed. However, TCP BBR does not decrease its congestion window size at the detection of packet losses. Thus, these methods decrease the throughput of TCP except for TCP BBR. These methods do not work well in case of TCP BBR significantly outperforms. We think a method that directly effects the parameter of TCP BBR, such as BtlBw and RTprop, is effective for improving fairness between TCP BBR and another TCP algorithm. We think also that utilizing TCP BBR instead of other TCP algorithms as possible is one of the promising solutions.

V. Conclusion

In this paper, we evaluated the fairness between TCP BBR, which is newly proposed and increasing its importance, and CUBIC TCP using practical network elements and practical TCP implementations. We then demonstrated that the fairness was low with high latency and that traditional methods for improving TCP fairness by dropping packets do not work well in case of TCP BBR consumed much bandwidth.

For future work, we plan to evaluate fairness with other queue management algorithm such as RED, evaluate fairness with several TCP algorithms, and discuss a method for improving TCP BBR.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Numbers 15H02696, 17K00109, 18K11277.

This work was supported by JST CREST Grant Number JPMJCR1503, Japan.

I would like to thank Utsumi Satoshi for useful discussions.

References

[1] Injong Rhee and Lisong Xu “CUBIC: A New TCP-Friendly High-Speed TCP Variant,” Proc. Workshop on Protocols for Fast Long Distance Networks, 2005, 2005.

[2] Kun Tan, Jingmin Song, Qian Zhang, and Murari Sridharan, “A Compound TCP Approach for High-speed and Long Distance Networks,” Proc. of IEEE Info COM 2005, July 2005

[3] J. Mo, R. J. La, V. Anantharam and J. Walrand, “Analysis and comparison of TCP Reno and Vegas,” INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proc. IEEE, 1999, pp. 1556-1563 vol.3. doi: 10.1109/INFCOM.1999.752178

[4] L. S. Brakmo and L. L. Peterson, “TCP Vegas: End to End Congestion Avoidance on a Global Internet”, IEEE Journal on Selected Areas in Communication, Vol.13, No.8, pp.1465-1480, October 1995.

[5] ITSUMI Hayato, YAMAMOTO Miki, "Improving Fairness between CUBIC and Compound TCP," IEICE Technical Report, vol. 110, no. 372, NS2010-160, pp. 103-108, 2010 (in Japanese)

[6] Ryo Oura Saneyasu Yamaguchi, “An Experimental Analysis of Fairness between Modern TCP Congestion Avoidance Algorithms,” FIT2011 vol. 10 issue 4 (2011) pp: 11-14 Published by Forum on Information Technology, Sep. 2011. (in Japanese)

[7] Masato Hanai, Saneyasu Yamaguchi, and Aki Kobayashi, “Modified Controlling Queue Delay for TCP Fairness Improvement,” The 18th Asia-Pacific Network Operations and Management Symposium (APNOMS 2016), October 2016.

[8] R. Oura and S. Yamaguchi, "Fairness Comparisons among Modern TCP Implementations," 2012 26th International Conference on Advanced Information Networking and Applications Workshops, Fukuoka, 2012, pp. 909-914. doi: 10.1109/WAINA.2012.167.

[9] S. Floyd and V. Jacobson, “Random early detection gateways for congestion avoidance,” IEEE/ACM Transactions on Networking, vol. 1, pp. 397.413, Aug. 1993.

[10] Kathleen Nichols and Van Jacobson. 2012. Controlling queue delay.Commun. ACM 55, 7 (July 2012), 42-50.

[11] Masato Hanai, Saneyasu Yamaguchi, Aki Kobayashi, “TCP Fairness Improvement by Modifying Packet Dropping of Controlled Delay,” IPSJ Journal, Vol.58 No.12, pp. 1993 - 2005, Dec. 2017

[12] Masato Hanai, Saneyasu Yamaguchi, and Aki Kobayashi. 2018. TCP Fairness Evaluation with Modified Controlled Delay in the Practical Networks. In Proceedings of the 12th International Conference on Ubiquitous Information Management and Communication (IMCOM '18). ACM, New York, NY, USA, Article 77, 7 pages. DOI: https://doi.org/10.1145/3164541.3164558

[13] Yuria Akiyama, Tomoki Kozu, Saneyasu Yamaguchi, "Active packet dropping for improving performance fairness among modern TCPs," The 16th Asia-Pacific Network Operations and Management Symposium (APNOMS 2014), 2014

[14] W. Richard Stevens, “TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms,” IETF RFC 2001, 1997.

[15] Neal Cardwell, Yuchung Cheng, C. Stephen Gunn, Soheil Hassas Yeganeh, and Van Jacobson, “BBR: Congestion-Based Congestion Control,” Queue 14, 5, pages 50 (October 2016), 34 pages, 2016. DOI: https://doi.org/10.1145/3012426.3022184

[16] L. Xu, K. Harfoush and I. Rhee, "Binary Increase Congestion Control for Fast Long-Distance Networks," Proc. IEEE Info COM 2004, March 2004

[17] T. Kozu, Y. Akiyama and S. Yamaguchi, "Improving RTT Fairness on CUBIC TCP," 2013 First International Symposium on Computing and Networking, Matsuyama, 2013, pp. 162-167. doi: 10.1109/CANDAR.2013.30

[18] Tomoki Kozu, Yuria Akiyama, Saneyasu Yamaguchi, "Improving RTT Fairness on CUBIC TCP," International Journal of Networking and Computing, Vol 4, No. 2, pp.291-306, 2014. http://doi.org/10.15803/ijnc.4.2_291

[19] Vint Cerf, Van Jacobson, Nick Weaver, and Jim Gettys. 2011. BufferBloat: What's Wrong with the Internet?. Queue 9, 12, Pages 10 (December 2011), 11 pages. DOI: https://doi.org/10.1145/2076796.2076798

[20] J. Gettys, "Bufferbloat: Dark Buffers in the Internet," in IEEE Internet Computing, vol. 15, no. 3, pp. 96-96, May-June 2011.

[21] J. Mo, R. J. La, V. Anantharam and J. Walrand, "Analysis and comparison of TCP Reno and Vegas," *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, New York, NY, 1999, pp. 1556-1563 vol.3. doi: 10.1109/INFCOM.1999.752178

[22] G. Hasegawa, K. Kurata and M. Murata, "Analysis and improvement of fairness between TCP Reno and Vegas for deployment of TCP Vegas to the Internet," *Proceedings 2000 International Conference on Network Protocols*, Osaka, 2000, pp. 177-186. doi: 10.1109/ICNP.2000.896302

[23] R. Oura and S. Yamaguchi, "Fairness Analysis among Modern TCP Congestion Avoidance Algorithms Using Actual TCP Implementation and Actual Network Equipments," *2011 Second International Conference on Networking and Computing*, Osaka, 2011, pp. 297-299. doi: 10.1109/ICNC.2011.56.

[24] M. Hock, R. Bless and M. Zitterbart, "Experimental evaluation of BBR congestion control," *2017 IEEE 25th International Conference on Network Protocols (ICNP)*, Toronto, ON, 2017, pp. 1-10. doi: 10.1109/ICNP.2017.8117540

[25] Feng Li, Jae Won Chung, Xiaoxiao Jiang, and Mark Claypool,”TCP CUBIC versus BBR on the Highway,” In Proceedings of the Passive and Active Measurement Conference (PAM), March 26-27, 2018. doi:10.1007/978-3-319-76481-8_20

[26] Masato Hanai, Saneyasu Yamaguchi, and Aki Kobayashi, "Improving TCP Fairness Between Modern TCP Algorithms Based on Controlling Delay," Workshop of 2016 IEEE 17th International Conference on High Performance Switching and Routing, Jun. 2016.

[27] iperf homepage, https://iperf.fr/