Electronic Preprint for Journal of Information Processing Vol.29

Technical Note

Fairness Improvement of Congestion Control with Reinforcement Learning

Meguru Yamazaki[1,a)] Miki Yamamoto[1,b)]

Received: March 6, 2021, Accepted: May 11, 2021

Abstract: With fast deployment of high speed wireless access networks, communication environments for internet access have been changing drastically. According to these wide range of network environments, a lot of TCP variants have been proposed. Each of these algorithms focuses on the specific environment and is designed with hardwired logic. This means there is no one-size-fits-all congestion control which can adapt to all environments. To resolve this problem, reinforcement learning based congestion control which learns operation suitable for each environment has been proposed. QTCP (Q-learning Based TCP) is one of the promising learning based TCPs. In this paper, we first reveal that a QTCP flow only behaves in the selfish manner of just increasing its own utility function, which causes unfairness between resource sharing flows. We propose a new QTCP congestion window control mechanism which is based on AIMD. Performance evaluation results show our proposal improves fairness without degrading high throughput and low latency characteristics of QTCP.

Keywords: congestion control, fairness, reinforcement learning

1. Introduction

Network applications consuming high bandwidth, e.g., video streaming, cloud gaming and AR/VR applications, are continuously requiring higher network bandwidth. In addition, with fast deployment of high speed wireless access networks, communication environments for internet access have been changing drastically. According to these wide range of network environments, a lot of TCP variants have been proposed [1].

These TCP variants include widely used approaches such as CUBIC [2] and Compound TCP [3], and approaches suitable for the specific environment such as DCTCP [4] and TCP Westwood [5]. The existence of a lot of TCP variants implies that there is no one-size-fits-all congestion control applicable to all the current internet environments. The limitation of existing TCP variants is their hardwired mapping nature [6].

Hardwired mapping means that the action for each of the specific events is strictly predefined. For example, TCP NewReno [7] halves the congestion window when a packet loss that implies network congestion is detected. Therefore, in the wireless environment where a packet loss might occur without correlation with congestion, NewReno has a chance to degrade its performance due to unnecessary window halving. The reason for this problem is the hardwired mapping that supposes a wired environment.

New kinds of TCP variants that can resolve the problem of hardwired mapping by adapting its performance to many kinds of network environments with machine learning have been proposed. Q-learning based TCP (QTCP) [8] has been proposed as one of these variants. QTCP applies reinforcement learning to Congestion Avoidance Phase of NewReno. In reinforcement learning, a learning agent learns the optimal action through trial and error. With this reinforcement learning, QTCP can learn how to adaptively change the congestion window and run without hardwired mapping. In addition, QTCP can achieve high throughput and low latency by considering the network utility (throughput and latency) as reward of reinforcement learning metrics.

When TCP variants with machine learning compete in the network, they may cause unfairness because they take a selfish action to increase their individual network utility. For example, in QTCP, each sender only chooses actions to increase its own throughput or decrease its own delay, and does not change in a direction of improving fairness. There are two ways to solve this problem. The first one is to change the reinforcement learning model to one that considers fairness. The second one is to adopt window control mechanism considering fairness, e.g., AIMD (Additive Increase Multiplicative Decrease).

In this paper, we take the latter approach to improve fairness of QTCP. At first, we show that QTCP converges unfair condition because it takes selfish actions and also its window control mechanism is MIMD (Multiplicative Increase Multiplicative Decrease). After that, we propose a new window control mechanism of QTCP based on AIMD. This paper is an extended version of our technical report [9], in which we only evaluated in a simple model. In this paper, we extend our work by evaluating our proposal in a more complicated model and show that our approach can improve fairness while keeping high throughput and low latency characteristics of QTCP.

[1] Graduate School of Science and Engineering, Kansai University, Suita, Osaka 564–8680, Japan
[a)] k563987@kansai-u.ac.jp
[b)] yama-m@kansai-u.ac.jp

2. QTCP

QTCP is the reinforcement learning congestion control based on TCP NewReno. Reinforcement learning is applied to Congestion Avoidance Phase of NewReno. Reinforcement learning is modeled with state space, action space and reward. In QTCP, state space is represented by network states and an action space is formed with the ways of how to change congestion window. And reward is degree of improvement of network utility [10] represented by throughput and delay. Namely, QTCP learns the ways of how to change congestion window that can achieve high throughput and low latency for each network state.

Network states are represented by three metrics: average inter packet sending time, average inter ACK arrival time and average RTT. They are calculated at each Time Interval (TI). Action is the variation of congestion window when ACK is received and is selected for each TI. The amount of each window change is selected from 3 options, +10 [byte], 0 [byte] (no change) and −1 [byte]. Reward is calculated by using network utility defined by the following equation:

$$Utility = \log(throughput) - \delta \cdot \log(\mathrm{RTT}), \tag{1}$$

where δ is a parameter. If the utility in the current TI is higher than the previous one, reward has a positive value, otherwise, a negative value. Throughput is calculated by the number of received ACKs in each TI. RTT is the exponential weighted average of measured RTT.

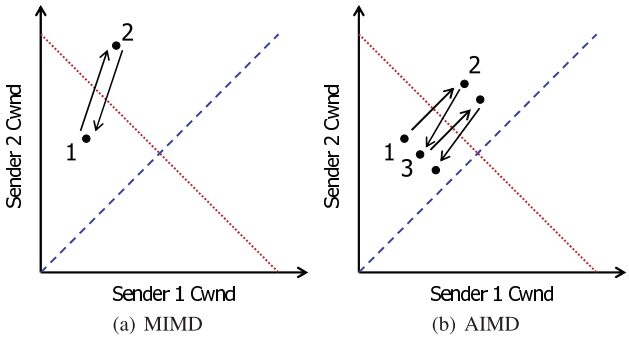
3. Fairness improvement for QTCP

In this section, we discuss fairness issues of QTCP and propose a new QTCP algorithm which resolves unfairness between bottleneck sharing flows.

When network utilization is less than 1, RTT is unresponsive to variation of congestion window because queue length is held 0 even with slight change of window size. On the other hand, throughput is responsive to variation of congestion window. Therefore, according to Eq. (1), the action of increasing window brings positive reward and the action of decreasing window brings negative reward. In this case, QTCP increases congestion window to increase throughput.

When network utilization is 1, throughput is unresponsive to the change of the congestion window size because increase of window size just leads to queue length growth. And RTT is responsive to variation of congestion window. Therefore, according to Eq. (1), the action of decreasing window brings positive reward under this assumption and accordingly QTCP decreases congestion window.

As described above, QTCP increases congestion window when network utilization is less than 1, and decreases congestion window when network utilization is 1. QTCP behavior is decided by its basic policy of maximizing utility function. This means when occasionally multiple flows sharing the same bottleneck link fall into unfair condition, e.g., one flow whose initial window size is small joins, the flow(s) obtaining high throughput will not decrease its window size because decrease of window size causes degradation of utility function. In this sense, QTCP behaves in a selfish manner.



(a) MIMD (b) AIMD

Fig. 1 Congestion window behavior in MIMD and AIMD.

Next, we would like to discuss QTCP window control mechanism from the perspective of fairness. Assume that TI is an integral multiple of RTT and the selected action does not change during one RTT. In this situation, QTCP window size after one RTT can be formulated as follows:

$$cwnd \leftarrow \begin{cases} cwnd * \left(1 + \dfrac{10}{\mathrm{MSS}}\right) & \text{(increase)}, \\ cwnd * \left(1 - \dfrac{1}{\mathrm{MSS}}\right) & \text{(decrease)}, \end{cases} \tag{2}$$

where MSS is maximum segment size. According to Eq. (2), QTCP multiplies congestion window by 1 + (10/MSS) [pkt] and 1 − (1/MSS) [pkt] in 1 RTT when QTCP selects action of increasing and decreasing window, respectively. This means QTCP window control mechanism is MIMD.

Figure 1 (a) shows congestion window sizes of two flows sharing the same bottleneck link. The red line indicates that the sum of throughput of two flows is equal to the link capacity. The blue line shows equal bandwidth share of the two flows. When congestion window sizes of two flows are given by point 1, link utilization is below 1. According to the above discussion, two flows behave in selfish manner and increase their congestion window in MI manner. The sum of the throughput increases along a vector connecting the origin and point 1. When it exceeds the red line, link utilization is 1. And at point 2, occasionally two flows decrease their window sizes in MI manner. In this decreasing phase, the sum of the throughput decreases also along a vector connecting point 2 and the origin. QTCP iteratively increases and decreases along this line, which means QTCP cannot converge to fair share of the blue line.

One of the most important issues for machine learning congestion control which utilizing utility function is fairness among flows. This is because of its selfish behavior. There can be two approaches for resolving this technical issue, one is significant modification of utility function and the other is change of MIMD to AIMD mechanism. In this paper, we try to resolve the QTCP fairness issue by the latter approach. Figure 1 (b) shows AIMD behavior. As shown in this figure, AIMD can improve fairness. So, we would like to integrate QTCP and AIMD approaches.

We set the amount of the congestion window variation when receiving an ACK to MSS/*cwnd* [byte] (Additive Increase) and −5 [byte] (Multiplicative Decrease). The amount of change of congestion window in 1 RTT is as follows:

$$cwnd \leftarrow \begin{cases} cwnd + 1 & \text{(increase)} \\ cwnd * \left(1 - \dfrac{5}{\text{MSS}}\right) & \text{(decrease)} \end{cases} \quad (3)$$

According to Eq. (3), the amount of variation in 1 RTT is 1 [pkt] in increase phase, and $(1-5/\text{MSS})$ [pkt] in decrease phase, which is totally an AIMD mechanism.

4. Evaluation

In this section, we evaluate default QTCP and QTCP-AIMD (QTCP with our proposal method) with computer simulation. In this simulation, we use ns-3 [11] as a network simulator. Simulation parameters are as follows: bottleneck link = 40 [Mbps], other links = 100 [Mbps], RTT = 120 [ms], Buffer size = 100 [pkt], Learning rate = 0.1, Discount rate = 0.9, Exploration rate = 0.1, $\delta = 10$. **Figure 2** shows simulation topology and each of the four senders starts its transmission every 500 seconds and leaves every 500 seconds after 2,000 seconds.

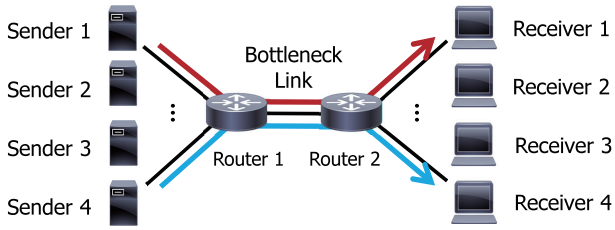


Fig. 2 Simulation topology.

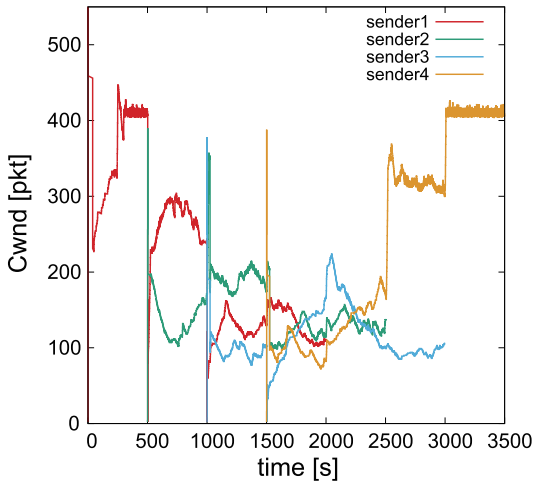


Fig. 3 Cwnd of QTCP.

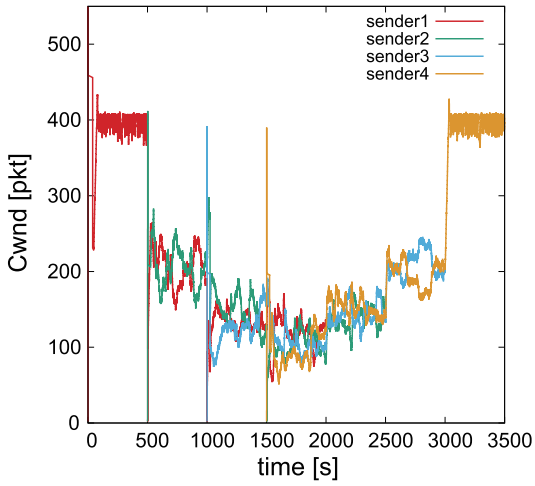


Fig. 4 Cwnd of QTCP-AIMD.

Figures 3 and **4** show characteristics of Cwnd in QTCP and QTCP-AIMD, respectively. QTCP-AIMD converges to fair Cwnd, while QTCP cannot converge to fair Cwnd. **Figures 5** and **6** show characteristics of bottleneck link throughput in QTCP and QTCP-AIMD, respectively. As shown in Fig. 6, QTCP-AIMD achieves high throughput as well as QTCP. **Figures 7** and **8**

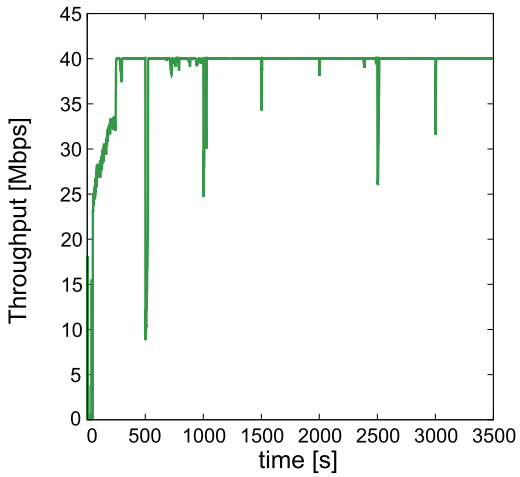


Fig. 5 Throughtput of QTCP.

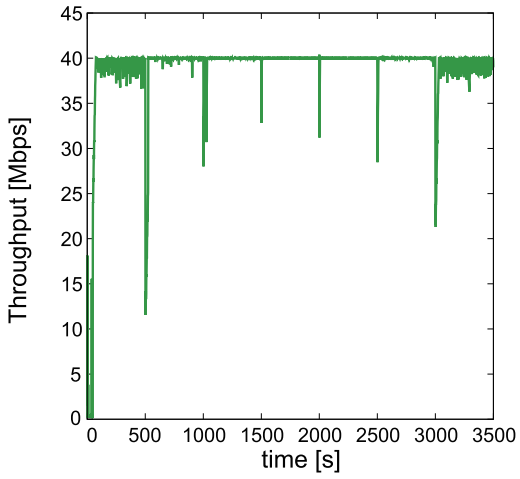


Fig. 6 Throughput of QTCP-AIMD.

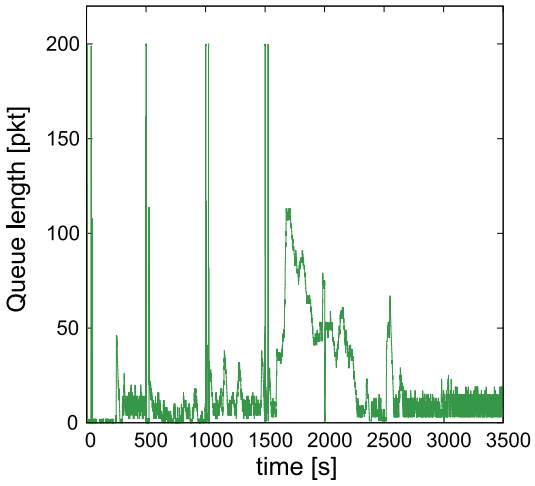


Fig. 7 Queue length of QTCP.

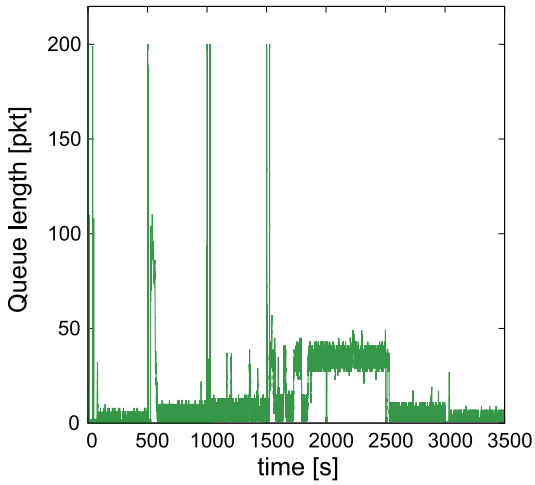


Fig. 8 Queue length of QTCP-AIMD.

show characteristics of the queue length of the bottleneck link in QTCP and QTCP-AIMD, respectively. As shown in Fig. 8, QTCP-AIMD achieves low queue length as well.

5. Conclusion

In this paper, we introduce QTCP which can perform without hardwired mapping as adaptive congestion control for diverse communication environments. However, QTCP has a technical problem of throughput fairness between senders. We propose a new QTCP window control mechanism based on AIMD. Despite QTCP's selfish behavior due to machine learning approach, our proposal (QTCP-AIMD) based on simple AIMD mechanism greatly improves fairness without degradation of throughput and queue length characteristics.

References

[1] Afanasyev, A., Tilley, N., Reiher, P. and Kleinrock, L.: Host-to-Host Congestion Control for TCP, *IEEE Communications Surveys & Tutorials*, Vol.12, No.3, pp.304–342 (online), DOI: 10.1109/SURV.2010.042710.00114 (2010).

[2] Ha, S., Rhee, I. and Xu, L.: CUBIC: A New TCP-Friendly High-Speed TCP Variant, *Proc. ACM SIGOPS Operating Systems Review*, pp.64–74 (online), DOI: 10.1145/1400097.1400105 (2008).

[3] Tan, K., Song, J. and Zhang, Q.: A Compound TCP Approach for High-Speed and Long Distance Networks, *Proc. IEEE INFOCOM*, pp.1–12 (onlne), DOI: 10.1109/INFOCOM.2006.188 (2006).

[4] Alizadeh, M., Greenberg, A., Maltzym, D.A., Padhyey, J., Pately, P., Prabhakarz, B., Senguptay, S. and Sridharan, M.: Data Center TCP (DCTCP), *Proc. ACM SIGCOMM*, pp.63–74 (online), DOI: 10.1145/1851182.1851192 (2010).

[5] Mascolo, S., Casetti, C., Gerla, M., Sanadidi, M. and Wang, R.: TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links, *Proc. ACM Mobicom*, pp.287–297 (online), DOI: 10.1145/381677.381704 (2001).

[6] Dong, M., Li, Q., Zarchy, D., Godfrey, P.B. and Schapira, M.: PCC: Re-architecting Congestion Control for Consistent High Performance, *Proc. USENIX NSDI*, pp.395–408 (2015).

[7] Henderson, T., Floyd, S., Gurtov, A. and Nishida, Y.: The NewReno Modification to TCP's Fast Recovery Algorithm, RFC 6582 (Apr. 2012).

[8] Li, W., Zhou, F., Chowdhury, K. and Meleis, W.: QTCP: Adaptive Congestion Control with Reinforcement Learning, *IEEE Trans. Network Science and Engineering*, Vol.6, No.3, pp.445–458 (online), DOI: 10.1109/TNSE.2018.2835758 (2019).

[9] Yamazaki, M. and Yamamoto, M.: A Study on Fairness of Reinforcement Learning Based Congestion Control, (in Japanese), *IEICE Technical Report*, Vol.119, No.194, NS2019-91, pp.13–18 (2019).

[10] Winstein, K. and Balakrishnan, H.: TCP ex Machina: Computer-Generated Congestion Control, *Proc. ACM SIGCOMM*, pp.34–39 (online), DOI: 10.1145/211990.212013 (2013).

[11] ns-3, http://www.nsnam.org/

Meguru Yamazaki received his B.E. degree from Kansai University in 2019. He is currently a M.E. student in Kansai University. His research interest includes network control with machine learning.

Miki Yamamoto received his B.E., M.E. and Ph.D. degrees in communications engineering from Osaka University in 1983, 1985, and 1988. He joined the Department of Communications Engineering at Osaka University in 1988. He moved to the Department of Electrical, Electronic and Information Engineering of Kansai University in 2005, where he is a professor. He visited the University of Massachusetts at Amherst in 1995 and 1996 as a visiting professor. His research interests include content oriented networks, high-speed networks, wireless networks, and the evaluation of performance of these systems. He is a member of IEEE, ACM, and IEICE.