Measurement and Redesign of BBR-based MPTCP

Jiangping Han, Kaiping Xue, Yitao Xing, Peilin Hong
University of Science and Technology of China
kpxue@ustc.edu.cn

David S.L. Wei
Fordham University

ABSTRACT

BBR is a congestion-based congestion control algorithm proposed to promote the performance of TCP. The interesting and vital question is thus: How would MPTCP perform on BBR? We first conduct extensive experimental evaluation of BBR-based MPTCP, and show that it provides several times throughput higher and more stable sending rate than that of others. We also observe that to replace the congestion control in MPTCP, some important algorithms in MPTCP need to be re-designed: 1) To achieve fairness with other flows, we propose a BBR based coupled congestion control algorithm, called Coupled BBR; 2) To further prevent performance degradation in highly dynamic nature, we also propose a scheduler to take advantage of dynamic redundancy.

CCS CONCEPTS

• **Networks** → **Transport protocols**; *Network algorithms.*

ACM Reference Format:
Jiangping Han, Kaiping Xue, Yitao Xing,, Peilin Hong, and David S.L. Wei. 2019. Measurement and Redesign of BBR-based MPTCP. In *SIGCOMM '19: ACM SIGCOMM 2019 Conference (SIGCOMM Posters and Demos '19), August 19–23, 2019, Beijing, China.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3342280.3342312

1 INTRODUCTION

MPTCP (multipath TCP) enables the full use of the device's multiple interfaces and transmit data via multiple paths concurrently [5]. Unfortunately, MPTCP inherits the problems from traditional TCP, such as performing badly in lossy network environments. BBR [3] is a congestion-based congestion control algorithm emerging in the past two years, which aims at achieving acceleration by proactively measuring bandwidth [2] for TCP. BBR measures and estimates the available bandwidth of path without taking reaction to losses, thus performs butter in lossy scenarios. Therefore, an interesting question is thus: **Can BBR also help promote the performance of MPTCP?** To answer this question, we conduct some tests on BBR-based MPTCP in both testbed and real network scenarios. BBR provides better tolerance of loss for MPTCP, which can increase the throughput performance to be several times higher than that of other algorithms.

SIGCOMM Posters and Demos '19, August 19–23, 2019, Beijing, China

ACM ISBN 978-1-4503-6886-5/19/08…$15.00
https://doi.org/10.1145/3342280.3342312

Also, should we redesign the existing algorithms for BBR-based MPTCP? Simply deploying original BBR into MPTCP, is not enough. Still, network fairness remains a problem for MPTCP [4]. However, BBR does not include an AIMD (Additive Increase Multiplicative Decrease) scheme, i.e. we could no longer adjust the increase or decrease speed to form coupled congestion control for BBR-based MPTCP as original MPTCP does. Therefore, we propose a novel coupled congestion control algorithm for BBR-based MPTCP (Coupled BBR), which uses the measurement bandwidth to adjusts each path's sending rate directly. Moreover, the redundancy of MPTCP may be additionally helpful in dynamic nature, while BBR offers an opportunity for adjusting redundancy according to the real-time measurement. Thus, we propose a novel scheduler to send redundant packets dynamically based on the BBR's proactive bandwidth measurement.

We deploy our scheme in MPTCP v0.94 to Linux kernel [1]. Our system does not require interaction between receiver and sender and only needs to modify the servers, which makes it easier to deploy.

2 MEASUREMENT

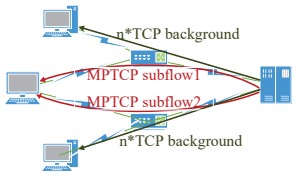


Figure 1: Testbed topology

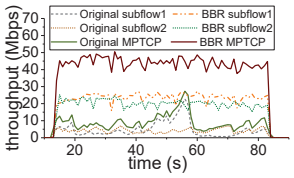


Figure 2: Test in testbed

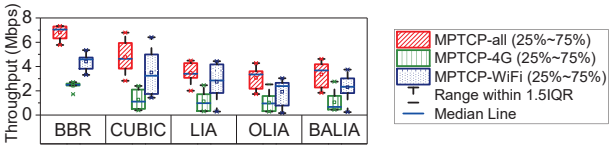


Figure 3: Test in real network

We first show the comparison of BBR-based MPTCP and original MPTCP in lossy environments. As shown in Fig. 1, a MPTCP connection has two subflows through different paths. Each path has 100Mbps bandwidth with random packet loss rate of 0.1%. There are two TCP background flows on each path using the same congestion control algorithm as in MPTCP. We run the BBR-based MPTCP and original MPTCP respectively and show the results in Fig. 2. Although 0.1% is not a large packet loss rate in the actual network, the throughput of original MPTCP drops dramatically, which is only 20% of BBR-based MPTCP. Because BBR-based MPTCP can almost reach the upper bound of the available bandwidth, while original MPTCP is limited by the packet loss. Moreover, the real-time

throughput of original MPTCP fluctuates very much, while that of BBR-based MPTCP remains stable without significant fluctuations.

To show the performance in the real network, we deploy MPTCP server on the cloud and download data using 4G and Wi-Fi. Fig. 3 shows the overall throughput of MPTCP and each subflow. BBR improves the overall throughput of MPTCP, and the improvement is more significant on 4G link, because packet loss is more serious on 4G link. In addition, it can also be seen that BBR-based MPTCP has better stability, in specific, the throughput fluctuation of both subflows and the whole connection are far less than that of other algorithms. CUBIC also performs better than LIA/OLIA/BALIA. However, it has the highest fluctuation because it increases the step size of window adjustment.

3 OUR DESIGN

3.1 Coupled BBR

Coupled BBR aims at network fairness and balanced congestion. To achieve this, it modifies the PROBE_BW phase of original BBR, in which BBR spends most of time (over 98%) at a steady sending rate. The PROBE_BW phase keeps 8 RTTs as a large cycle with sending rate of (1.25, 0.75, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0) $*BW_i$, where BW_i is the maximum bandwidth detected on subflow$_i$. Coupled BBR does not change the sending rate of the first two RTT to ensure the ability for each subflow measuring available bandwidth BW_i. For the next 6 RTTs, Coupled BBR replaces the sending rate with a smaller $\alpha_i * BW_i$. Where α_i is related to the bandwidth of each subflow: $\alpha_i = (4\beta_i - 1)/3$, where $\beta_i = \frac{BW_i \cdot \max\{BW_i\}}{\sum BW_i^2}$. If the calculated α_i is less than zero, Coupled BBR just sets the congestion window of subflow$_i$ to 4 packets.

Coupled BBR maintains a cycle rate of (1.25, 0.75, α_i, α_i, α_i, α_i, α_i, α_i) $*BW_i$ to send data for each subflow$_i$. In fact, each subflow$_i$ of Coupled BBR will reach a average throughput of $\frac{BW_i^2}{\sum BW_i^2} \cdot \max\{BW_i\}$, and overall MPTCP will reach a throughput of $\max\{BW_i\}$. For now, Coupled BBR satisfies the goals we set: it achieves fairness with other flows, and it has the ability of balanced congestion.

3.2 Our Scheduler

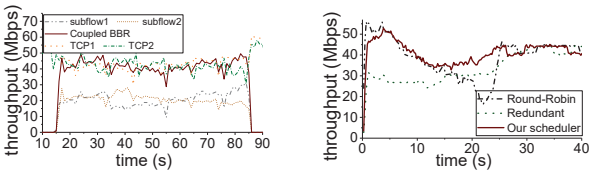
Besides fairness, MPTCP should also provide more robustness than TCP in highly dynamic scenarios. Based on this consideration, our scheduler adaptively sends redundant packets to avoid performance degradation caused by packet loss on "bad" subflows. It detects the subflow's performance in real-time and sends redundant packets when the following conditions hold:

1) When $cwnd_i <= 4$. The subflow may not be able to discover the packet loss in time until the timer times out. When multiple subflows send different data at the same time, this packet loss on the single subflow may even affect other subflows.

2) When $\alpha_i < RTT_i/(N * (RTT_i + \min RTT_i))$. It means that the sending rate of subflow$_i$ is too small, the scheduler schedules redundant data on subflow$_i$. The scheduler tends to schedule redundant packets on the subflow with low bandwidth and high RTT.

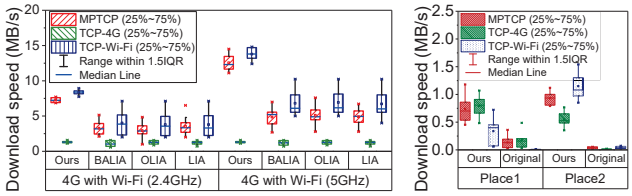
4 PERFORMANCE EVALUATION

We first test our scheme under the same test topology as shown in Fig. 1. The two paths both have 100Mbps bandwidth and the packet loss rates are 0 and 1% respectively. The curve of "TCPi" in Fig. 4(a) shows the throughput of one TCP background flow on pathi. Coupled BBR accurately achieves the principle of fairness and keeps less fluctuation of sending rate. Fig. 4(b) shows the performance of our scheduler. We download files using coupled BBR with our scheduler and Round-Robin/Redundant, respectively. During the experiment, two paths both perform well at beginning. After 5s, one path's loss rate keeps growing up to 20%. Round-robin performs better than Redundant within the first 15s, and Redundant performs better after 15s. Moreover, within the whole process, our scheduler always performs the best.



(a) Performance of Coupled BBR. (b) Performance of our scheduler.

Figure 4: Performance of our scheme in testbed.



(a) Different path conditions. (b) different places.

Figure 5: Download speed in real network.

We also deploy MPTCP server with our scheme on the cloud and conduct some tests with different scenarios. We download files using our scheme through 4G and Wi-Fi, and using TCP through 4G and Wi-Fi respectively. When MPTCP uses our Coupled BBR and scheduler, TCP flows use BBR . When MPTCP uses traditional algorithms, TCP flows also use traditional newReno. In Fig. 5, with the different path conditions, BBR provides more performance gain in the second scenario, whose Wi-Fi link has larger bandwidth and higher loss rate. Meanwhile, our scheme works well at each scenario that can accurately achieve the principle of fairness. Moreover, the performance of our scheme is still much better than the original MPTCP, and the overall throughput of our scheme is two times higher than that of original MPTCP. Fig. 5(b) shows the performance when we place our server in some other places (in foreign countries), which suffer much higher level of delay and loss rate. In this scenario, original MPTCP is hard to work. However, our scheme still works well in these places, which gains more than 10 times better performance than that of the original MPTCP.

5 ACKNOWLEDGMENTS

This work is supported in part by the National Key R&D Program of China under Grant No. 2017YFB0801702, the National Natural Science Foundation of China under Grant No. 61671420, and Youth Innovation Promotion Association of the Chinese Academy of Sciences (CAS) under Grant No. 2016394.

REFERENCES

[1] [n.d.]. Linux MPTCP kernel. http://www.multipath-tcp.org/. Accessed on: 2019.

[2] Eneko Atxutegi, Fidel Liberal, Habtegebreil Kassaye Haile, Karl-Johan Grinnemo, Anna Brunstrom, and Ake Arvidsson. 2018. On the Use of TCP BBR in Cellular Networks. *IEEE Communications Magazine* 56, 3 (2018), 172–179.

[3] Neal Cardwell, Yuchung Cheng, S Hassas Yeganeh, and Van Jacobson. 2017. BBR congestion control. *Working Draft, IETF Secretariat, Internet-Draft draft-card-well-iccrg-bbr-congestion-control-00* (2017).

[4] Simone Ferlin, Özgü Alay, Thomas Dreibholz, David A Hayes, and Michael Welzl. 2016. Revisiting congestion control for multipath TCP with shared bottleneck detection. In *Proceedings of IEEE INFOCOM*. IEEE, 1–9.

[5] Li Li, Ke Xu, Tong Li, Kai Zheng, Chunyi Peng, Dan Wang, Xiangxiang Wang, Meng Shen, and Rashid Mijumbi. 2018. A measurement study on multi-path TCP with multiple cellular carriers on high speed rails. In *Proceedings of ACM SIGCOMM*. ACM, 161–175.