Mutual Influence of Opposite TCP Flows in a Congested Network

Nikolai Mareev, Dmytro Syzov, Dmytry Kachan, Kirill Karpov, Maksim Iushchenko and Eduard Siemens

Future Internet Lab Anhalt, Anhalt University of Applied Sciences, Bernburger Str. 57, Köthen, Germany
{*nikolai.mareev, dmytro.syzov, dmitry.kachan, kirill.karpov, maksim.iushchenko, eduard.siemens*}@*hs-anhalt.de*

Keywords: TCP, IP, Highspeed, Congestion Control Algorithm, Internet, Performance, Bidirectional, Two-Way, 10G Networks, BBR, CUBIC, Coexistence.

Abstract: With the rapid growth of the Internet community, some of the simple and familiar tasks related to the field of data transfer are becoming increasingly complex. A modern worldwide network can offer high-speed channels and many opportunities for IT companies that provide high load through the Internet. This creates a bunch of new problems for software solutions and algorithms in the field of high-speed digital c ommunications. This article observes one of these problems: the mutual influence between two mutually opposite single-threaded TCP flows with the various congestion control algorithms. In this paper, some of the most efficient congestion control algorithms were tested on a real network using channel emulation equipment. The test results presented in the article show that two-way TCP data transfer with modern congestion control algorithms can lead to a significant performance drop.

1 INTRODUCTION

Transmission Control Protocol (TCP) provides a set of functions for automatically controlling sender parameters during data transfer in TCP/IP networks. One of these functions is the congestion control algorithm, that addresses three features:

- Prevent network devices from overloading.
- Achieve high bottleneck bandwidth utilization.
- Share the network resources with other flows.

The network congestion is a situation when a network node receives more data than it can handle or forward. Network congestion results in an overloaded transmission buffer on network devices, additional network delay, and packet drops. Congestion control algorithms (CCA) can be divided into groups according to the main indicator of congestion - the data transfer parameter, which corresponds to network congestion. Key congestion indicators are network delay, packet loss, and available bandwidth. Delay-based congestion control algorithms (VENO [1], VE-GAS [2]) are designed to proactively detect network congestion - before packet loss occurs. Common issues of such algorithms are unfair resource sharing and low bottleneck bandwidth utilization. Loss-based and loss-delay-based algorithms (CUBIC [3], YEAH [4]) treat packet loss as network congestion. Achieving high bottleneck bandwidth utilization is another important challenge for congestion control algorithm. Different types of CCAs use different data rate control schemes and require different depths of the bottleneck queue buffers to fully utilize the bottleneck bandwidth. The third challenge for congestion control algorithms is resource sharing. Network resources, such as bottleneck bandwidth or port queue depth, are limited. Sharing network resources require additional methods in the algorithm and rely on the congestion indicators dynamics. BBR [5] is a congestion-based congestion control algorithm developed by Google past few years. This algorithm uses the bottleneck bandwidth estimation as the primary indicator and the round trip time as the secondary indicator of congestion. BBR can achieve relatively high data transfer performance in cases where packet loss can occur on a non-congested link.

The main purpose of this article is to present a study of the mutual influence of two mutually opposite TCP data streams in a congested network. Particular attention was paid to eliminating hardware, cross-traffic, and other possible impacts on the results. Work has been performed in Future Internet Lab Anhalt [6].

The rest of this document is organized as follows: The second Section provides a brief overview of TCP coexistence issues. Section 3 describes the experimental setup and properties of the experiment. Test results and evaluation are presented in Section 4. Section 5 contains a discussion of the results provided, and Section 6 contains a conclusion.

2 TCP COEXISTENCE

The simultaneous coexistence of different TCP data streams in the same channel requires special behavior of the congestion control algorithms for the fair sharing of network resources. Essentially, during data transfer, the congestion control algorithm probes the bandwidth by changing the data transfer rate and measures the parameters of the connection i.e. congestion indication. In the case of loss-based congestion control algorithms, packet losses considered as a sign of congestion. This leads to a certain behavior during data transfer: the amount of data increases until the bottleneck of the port buffer is overloaded and some amount of data packets are dropped. Such algorithms relatively fairly share network resources among themselves and can provide high data transfer performance. The modern trend is to increase the depth of the queue buffers over the network. In cases with fat network buffers, loss-based congestion control algorithms have a strong negative effect on network delay [7]. However, most TCP connections are controlled by congestion control algorithms based on loss or loss-delay congestion indication.

Delay-based congestion control algorithms use changes in the network delay as an indication of network congestion. This allows keeping the load level of the bottleneck queue buffers at some lower level than loss-based algorithms do. Such algorithms have less aggressive behavior compared to loss- or loss-delay based algorithms, it leads to unfair sharing of the network resources. However, there are several different strategies for achieving fairness between loss- and delay-based congestion control algorithms [8].

A relatively new solution, the BBR congestion control algorithm, uses probing cycles to estimate available bandwidth, network delay, and channel state. BBR tends to keep low bottleneck queue buffer load level and achieve high bandwidth utilization. Another important feature of BBR is packet losses tolerance and high performance in lossy networks. This strategy allows in most cases to nearly fairly share network resources during coexistence with loss-based TCP flows. However, BBR is still under development and has several performance issues [9, 10, 11].

Congestion control algorithms use the dynamics of congestion indicators to mutually influence each other during coexistence and change the data transfer rate for the main purpose of sharing network resources. In case of one-way congestion, the dynamic behavior of congestion indicators is expected in network latency and available bandwidth. In case of two-way network congestion, main congestion indicators may have unexpected behavior due to the influence of the two-way data stream, and lead to performance issues.

3 EXPERIMENTAL SETUP

Testbed network is presented in Figure 1. Core elements in the network are Netropy 10G and Netropy 10G2 - WAN emulators from Apposite Technologies [12]. These devices allow to emulate various network conditions by setting the properties of the channel (see Table 1) and saving per second statistics of the forwarded data stream, such as data transfer rate, queue buffer load level, packet loss, etc. All data flow statistics in this work are collected by Netropy devices.

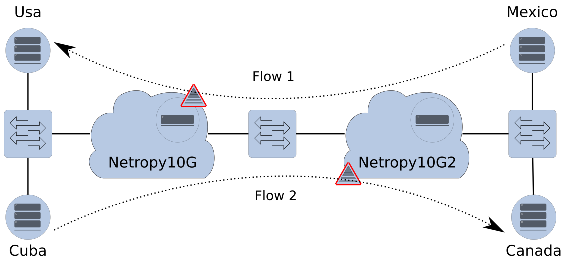

Figure 1: Experimental network.

Table 1: Netropy WAN emulators description.

Label	Netropy10G	Netropy10G2
Max. Agg. Throughput	20Gbps	40Gbps
Max. Packet Rate	29Mpps	59.5Mpps
Bandwidth	from 100 bps to 10 Gbps	
Queuing	RED or tail drop queue management; priority or round robin queuing;	
Queue depth	up to 100MB	
Latency	0 ms – 10000 ms or greater in each direction in 0.01 ms increments; constant, uniform, exponential, normal distributions with or without reordering; accumulate and burst delay;	
Packet loss	random, burst, periodic, BER, Gilbert-Elliott, or recorded packet loss; data corruption; network outage	

The second important element in the testbed is a network switch - Extreme Networks Summit x650-24x [13]. It has 24 10GBASE-X SFP+ interfaces, 488 Gbps maximum aggregated bandwidth and 363 Mpps maximum packet throughput. It is an edge-level network switch with tiny shared queue port buffer. The last elements on the scheme are servers (named as follows: Usa, Mexico, Cuba and Canada) with common specifications:

- 64GB DDR4 of RAM.
- Intel Corporation 82599ES 10-Gigabit SFI/SFP+ NIC.
- Linux 5.3.0-24-generic x86 64 Kernel.
- Intel(R) Xeon(R) CPU E5-2̄643 v4 3.40GHz CPU.

Provided tests require the exclusion of a possiblenegative impact from the OS and hardware on the process of transferring data. Each test case includes the following features.

- To exclude OS-level resource sharing / competition / queuing, a separate pair of servers were used for each TCP data stream.
- Bottleneck queues in both directions were configured separately on different WAN emulators in order to eliminate possible specific queue management problems in cases of two-way congestion.
- The emulated bottleneck bandwidth in all tests was configured at a level that is significantly lower than the maximum bandwidth of network devices in the testbed.
- The maximum data transfer rate was significantly lower than the maximum aggregated throughput of the tested devices.
- The emulated network delay was configured on 20ms to exclude possible overreact issues on the TCP congestion control side (TCP congestion control can show unexpected behavior in cases of LAN network delay)
- Bottleneck buffer queue depth has been set as 2.5 MB (tail drop queuing algorithm) according to the rule-of-thumb recommendations mentioned in [14, 15].

All tests have been performed with iperf3 ver. 3.6 TCP traffic generation utility [16].

4 EXPERIMENTAL RESULTS

To observe the behavior of data transfer of both streams separately and in coexistence TCP flows were started with a time interval of 50 seconds between each other. The interaction period of oncoming traffic is 150 seconds and shows the mutual influence of TCP data streams in case of two-way network congestion.

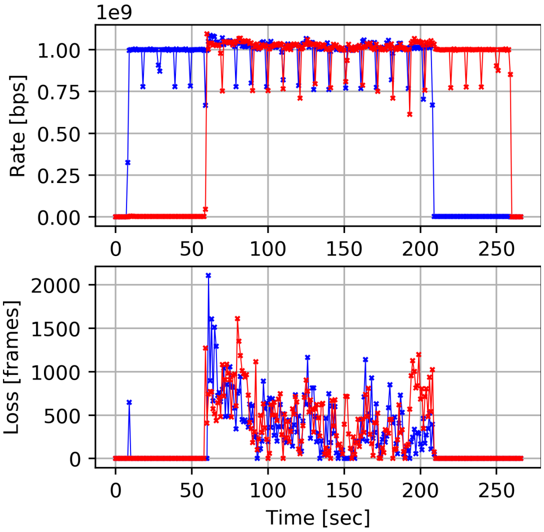

Figure 2: Two TCP BBR mutually reverse data flows.

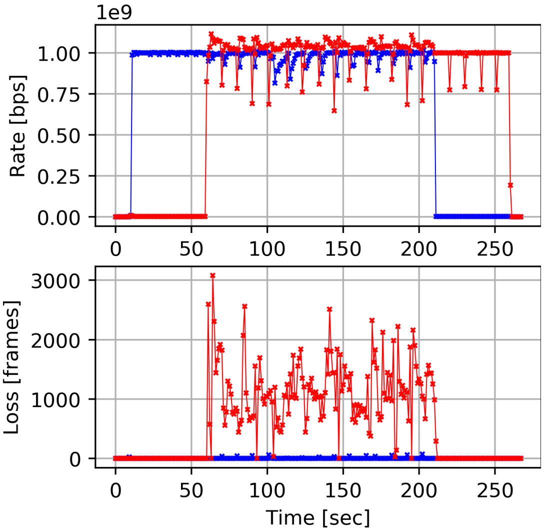

Figure 3: TCP CUBIC (blue) and TCP BBR (red) mutually reverse data flows.

On the Figure 2, an example of the mutual influence of two counter TCP BBR data flows is presented. TCP BBR requires relatively low bottleneck queue buffer during data transmission and it perfectly fits in the given test environment. Bottleneck bandwidth is fully utilized and no packet losses detected until the second TCP BBR flow appears in the link. The interaction of two data flows on a this link leads to overloaded bottleneck queue buffers and massive packet drops in both directions. It breaks the resource sharing ability of an algorithm and excludes any additional loss- or loss-delay based congestion control TCP flow in this link. However, the bottleneck bandwidth is utilized fully during the coexistence period.

The mutual influence of TCP counter BBR and

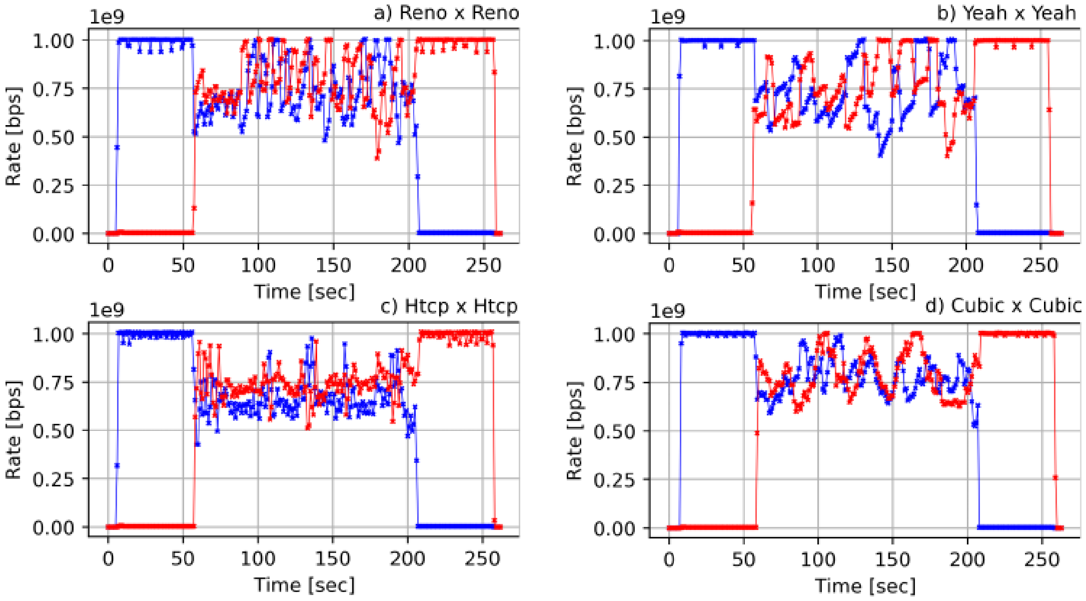

Figure 4: Different loss-based TCP congestion control mutually reverse data flows.

TCP CUBIC traffic is shown in Figure 3. The TCP BBR stream loses twice as many packets as the previous case. The TCP flow in the opposite direction to the BBR is controlled by the TCP CUBIC congestion control and shows a slight decrease in data rate during coexistence.

Highly efficient congestion control algorithms are observed in the [17] by Lukaseder T. et al., these CCAs was decided to test in the proposed case. Figure 4 shows the inter-protocol mutual influence of TCP streams with various loss and loss-delay congestion control algorithms: TCP RENO, TCP YEAH, HIGH-SPEED TCP, and TCP CUBIC. Each tested congestion control algorithm fits in the channel and utilizes full available bandwidth until another data flow started. Compared to TCP BBR, loss-based algorithms show a much higher influence on each other during coexistence. Performance degradation during this type of coexistence can be described by reduction of bottleneck bandwidth utilization by up to 25 %.

5 DISCUSSION

Delay-based congestion control algorithms have well-known issues of resource sharing during coexistence [18] and were not included in the article. The main goal of the article is to observe the behavior of the most popular congestion control algorithms. The issue of the performance drop during the two-way network congestion is the influence on the congestion indicators in both directions. In such a case the round-trip-time delay (RTT) measured by first flow would be influenced by queuing delay load in the opposite direction caused by the second data flow. Loss-based congestion control algorithms treat changes in the network delay and packet losses as the signals to release the bandwidth, like in one-way coexistence. This behavior leads to a drop in the bottleneck bandwidth utilization. Another influence is caused by packets in the feedback channel of the flows. A lot of service packets from the downstream flow are including in the data packets of the upstream data flow disturbing a bottleneck queue and provide an additional network delay and packet losses.

A possible solution for this issue could be the usage of one-way network delay (OWD) as the congestion indication instead of a round-trip-time delay. This would exclude the influence of a feedback channel on the congestion indication. It would also exclude additional network delay jitter in the feedback channel and, probably, increase the data transmission performance. Nevertheless, clock drift is a serious problem, and such a strategy requires additional algorithms for proper operation. Low priority TCP congestion control algorithms like TCP LP [19] or TCP LEDBAT [20] also shows performance drop in case of bidirectional network congestion. It is confusing because these algorithms use one-way delay instead of RTT for the congestion indication. Probably the implementation of these algorithms in the Linux kernel is actually using RTT instead of OWD.

6 CONCLUSION

In this article a mutual influence of counter TCP data flows in the case of bidirectional network congestion was observed. loss- and delay-based congestion control algorithm demonstrates significant performance degradation during such a test case, up to 25% data rate drop. TCP BBR, a congestion based congestion control algorithm demonstrates still high bottleneck bandwidth utilization, however, two-way network congestion leads to massive packet losses and impossibility of share the bandwidth with other loss-based flows. Future work including OWD-based congestion indication implementation in RMDT [21] protocol or/and research of the congestion indication in the TCP low priority congestion control solutions in the Linux kernel.

ACKNOWLEDGEMENTS

This work has been funded by Volkswagen Foundation for trilateral partnership between scholars and scientists from Ukraine, Russia and Germany within the project CloudBDT: Algorithms and Methods for Big Data Transport in Cloud Environments.

REFERENCES

[1] Ch. Peng Fu and S. Liew, “TCP veno: TCP enhancement for transmission over wireless access networks,” IEEE Journal on Selected Areas in Communications, vol. 21, no. 2, pp. 216–228, February 2003.

[2] L.S. Brakmo and L.L. Peterson, “TCP vegas: end to end congestion avoidance on a global internet,” IEEE Journal on Selected Areas in Communications, vol. 13, no. 8, pp. 1465-1480, October 1995.

[3] S. Ha, I. Rhee and L. Xu, “CUBIC: a new TCP-friendly high-speed TCP variant,” ACM SIGOPS Op-erating Systems Review, vol. 42, no. 5, pp. 64-74, July 2008.

[4] A. Baiocchi, A. P. Castellani, and F. Vacirca, “YeAH-TCP: Yet another highspeed TCP,” In proc. The fifth PFLDNET workshop, February 2007.

[5] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and Van Jacobson, “BBR: congestion-based congestion control,” vol. 60, no. 2, pp. 58-66, January 2017.

[6] Future Internet Lab Anhalt. [Online]. Available: https://fila-lab.de/

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

[8] M. Hock, R. Bless and M. Zitterbart, “Toward coexistence of different congestion control mechanisms,” IEEE 41st Conference on Local Computer Net-works (LCN), pp. 567-570, November 2016.

[9] K. Miyazawa, K. Sasaki, N. Oda and S. Yamaguchi, “Cycle and divergence of performance on TCP BBR,” IEEE 7th International Conference on Cloud Networking (CloudNet), October 2018.

[10] N. Mareev, D. Kachan, K. Karpov, D. Syzov and Siemens, “Efficiency of BQL Congestion Control under High Bandwidth - Delay Product Network Conditions,” Proc. of the 7th International Conference on Applied Innovations in IT, (ICAIIT), pp. 19–22, March 2019.

[11] K. Sasaki, M. Hanai, K. Miyazawa, A. Kobayashi, N. Oda and S. Yamaguchi, “TCP Fairness Among Modern TCP Congestion Control Algorithms Including TCP BBR,” IEEE 7th International Conference on Cloud Networking (CloudNet), October 2018.

[12] Leaders in network emulation and testing. [Online]. Available: https://www.apposite-tech.com/

[13] End-to-end cloud driven networking solutions. [Online]. Available: https://www.extremenetworks.com/

[14] A. Dhamdhere, Hao Jiang and C. Dovrolis, “Buffer sizing for congested internet links,” Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 2, 2005, pp. 1072-1083.

[15] C. S. Curtis Villamizar, “High performance TCP in ANSNET,” ACM Computer Communications Review, pp. 45-60, September 1994.

[16] iPerf - the TCP, UDP and SCTP network bandwidth measurement tool. [Online]. Available: https://iperf.fr/

[17] T. Lukaseder, L. Bradatsch, B. Erb, R. W. Heijden and F. Kargl, “A Comparison of TCP Congestion Control Algorithms in 10G Networks,” IEEE 41st Conference on Local Computer Networks (LCN), pp. 706-714, November 2016.

[18] R. Al-Saadi, G. Armitage, J. But and P. Branch, “A survey of delay-based and hybrid TCP congestion control algorithms,” IEEE Communications Surveys & Tutorials, vol. 21, no. 4, pp. 3609-3638, 2019.

[19] A. Kuzmanovic and E. Knightly, “TCP-LP: a dis-tributed algorithm for low priority data transfer,” IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 3, pp. 1691-1701, 2003.

[20] D. Rossi, C. Testa, S. Valenti and L. Muscariello, “LEDBAT: The new BitTorrent congestion control protocol,” in Proc. of 19th International Conference on Computer Communications and Networks (IC-CCN 2010), August 2010.

[21] D. Syzov, D. Kachan and E. Siemens, “High-speed UDP data transmission with multithreading and automatic resource allocation,” Proc. of the 4th International Conference on Applied Innovations in IT,(ICAIIT), pp. 51-55, March 2016.