Bufferbloat Problem

CSC458 Computer Networks Programming Assignment 2

Author: Tingfeng Xia, University of Toronto, `mailto:tingfeng.xia@mail.utoronto.ca`
Date: Tuesday 24th November, 2020

Abstract Bufferbloat refers to the existence of excessively large and frequently full buffers inside a network. [Sivov and Sokolov, 2012]

Most TCP congestion control algorithms rely on packet drops to determine the available bandwidth between two ends of a connection.[Allman and Paxson, 2009, El-Sayed et al., 2011] In general, TCP congestion control algorithms speed up the data transfer until packets start to drop, then slow down the transmission rate. Under ideal conditions, we expect an equilibrium speed to be reached after a period of time of adjustments.

To illustrate the problem of Bufferbloat, let's consider an internet topology illustrated in Figure 1. From our perspective, h_1 to s_0 via ℓ_1 should be sending at maximum 10Mbps to avoid congestion, since ℓ_2 is the bottleneck in our network with a link speed of 10Mbps. However, from h_1's perspective, it only knows to decrease data transfer speed when it receives signal of packet drop, which will happen after the buffer of s_0 is full. Importantly, h_1 will not slow down its data transfer speed until s_0 buffer is saturated and s_0 begins to drop packets from h_1. In other words, the buffer within s_0 caused packet drop signals cease to be a timely indication of congestion, which TCP congestion avoidance algorithms rely heavily on.

Bufferbloat causes increase in queueing delay, and thus causes end users to experience increase in latency, which is the sum of transmission delay, processing delay, and queueing delay. [Sivov and Sokolov, 2012] Bufferbloat also causes jitters and decreases the overall throughput of the network.

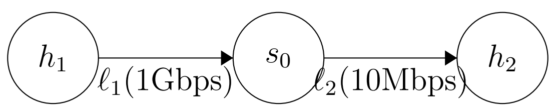


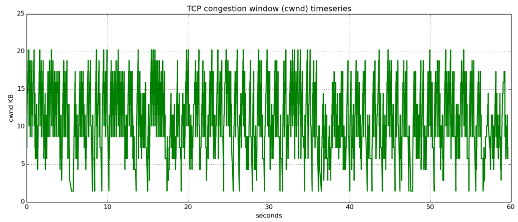
Figure 1: Assignment Topology. Link ℓ_1 and ℓ_2 has bandwidths 1Gbps and 10Mbps respectively; Router s_0 has buffer size 150kB, and assuming MTU of 1500 bytes it can buffer approximately 100 packets.

Methods

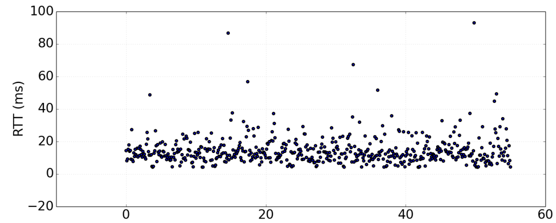
We emulate our Assignment topology, as illustrated in Figure 1, internet topology using mininet. Then, we simultaneously perform the following three tasks

- start a long lived TCP flow sending data from h_1 to h_2, and
- send 1 ping per 0.1 second from h_1 to h_2, and
- spawn an web server on h_1, and download the webpage from h_1 once every two seconds.

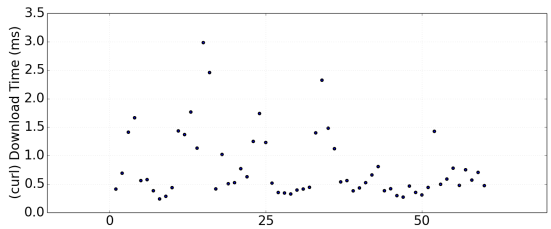
This simulation is repeated for three different queue sizes, $Q = 5/20/100$ pkts. Then, for each max queue size, we plot the time-series of the long-lived TCP flow's cwrd, the RTT reported by ping, the webpage download time, and the queue size at the router s_0.



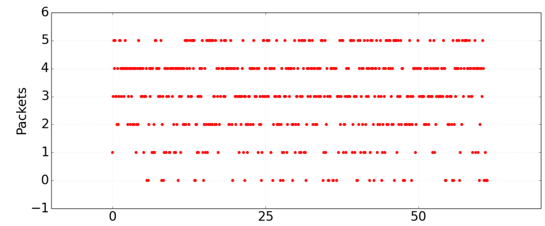
(a) Long-lived TCP flow's cwrd



(b) RTT repoted by ping

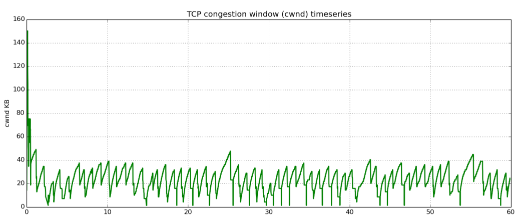


(c) Webpage download time

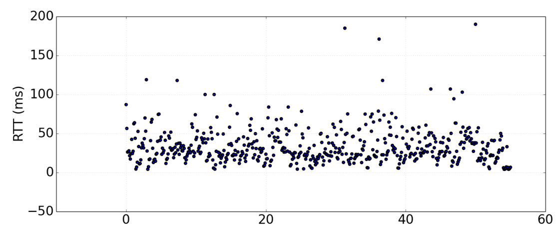


(d) Queue size at the router

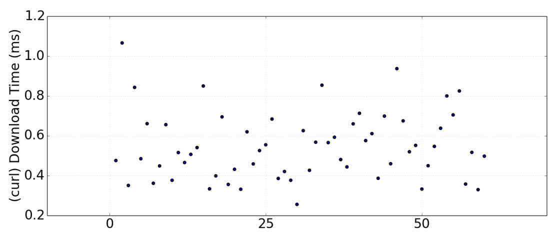
Figure 2: Long-lived TCP flow's cwrd, RTT reported by ping, webpage download time, and queue size at the router with max buffer size of 5.



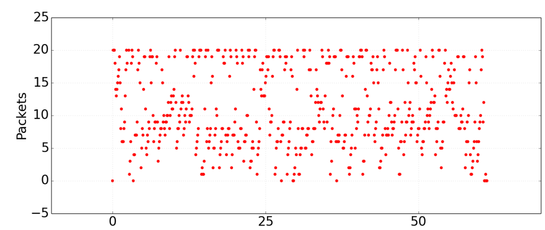
(a) Long-lived TCP flow's cwrd



(b) RTT repoted by ping

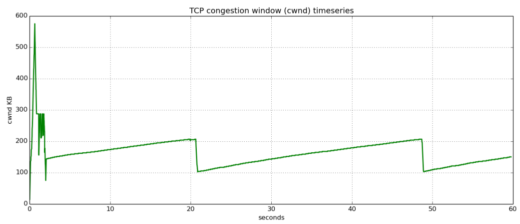


(c) Webpage download time

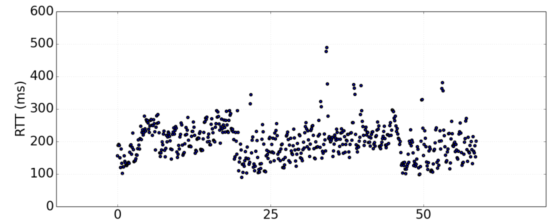


(d) Queue size at the router

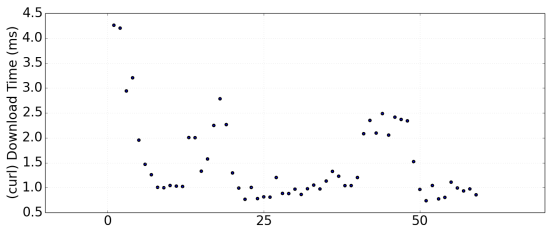
Figure 3: Long-lived TCP flow's cwrd, RTT reported by ping, webpage download time, and queue size at the router with max buffer size of 20.



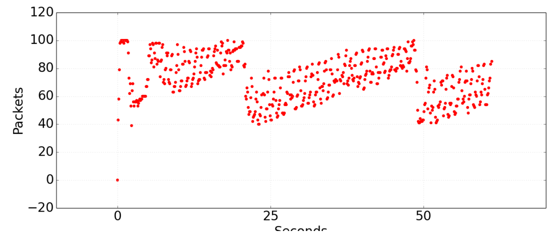
(a) Long-lived TCP flow's cwrd



(b) RTT repoted by ping



(c) Webpage download time



(d) Queue size at the router

Figure 4: Long-lived TCP flow's cwrd, RTT reported by ping, webpage download time, and queue size at the router with max buffer size of 100.

Results

Figure 2a is consists

Discussion

References

[Allman and Paxson, 2009] Allman, M. and Paxson, V. (2009). Tcp congestion control. Rfc, RFC Editor.

[El-Sayed et al., 2011] El-Sayed, A., HAGGAG, S., and EL-FESHAWY, N. (2011). A survey for mechanisms for tcp congestion control. *International Journal of Research and Reviews in Computer Science (IJRRCS)*, 02:676–682.

[Sivov and Sokolov, 2012] Sivov, A. and Sokolov, V. (2012). The bufferbloat problem and tcp: Fighting with congestion and latency.