Bufferbloat Assignment Report

Hokyun Park

Fall 2021

Introduction

The bufferbloat problem is caused by excessively large buffers available in the network. The large buffers with its lower egress bandwidth than its ingress bandwidth causes packets to build up in its queue. If these large buffers are filled, it can lead to even higher packet loss and induce devastating latencies for everyone in the network.[1]

The latency is mainly due to the queuing delay. It is the time a packet resides in the queue until it takes its turn to be transmitted out to a next hop. The experiment was performed to simulate bufferbloat effect and demonstrate how it leads to increase in latency.

Experimental Set Up

The simulation was implemented using mininet. Through mininet, a small network consist of $h1$, $h2$, and $s0$ was established. $h1$ mainly functioned as a server, $h2$ functioned as a client host and $s0$ functioned as a router that navigates packets between $h1$ and $h2$.

The link between $h1$ and $s0$ had bandwidth of 1Gbps and the link between $s0$ and $h2$ had a bandwidth of 10Mbps. To simulate bufferbloat effect, three different queue size - 5 packets, 20 packets, 100 packets - were imposed on the buffer of $s0$ and analyzed how its size impacts the transmission of data between $h1$ and $h2$.

Three types of transmissions measured is:

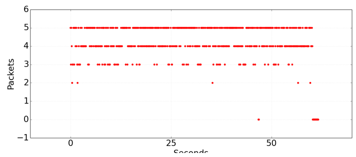
- Long-lived TCP sending data from $h1$ to $h2$ using $iperf$
- Round Trip Time (RTT) of 10 pings per second from $h1$ to $h2$
- Webpage download time of size 174KB every 2 seconds. The webpage was downloaded from $h1$ to $h2$.

Furthermore, using a packet size, one can easily calculate the maximum queueing delay. For example, the maximum transmit queue length of `eth0` interface of virtual machine used in experiment is 1000 packets and Maximum Transmission Unit of this interface is 1500 bytes = 12000bits. If the queue drains at 100Mb/s, the maximum queue time a packet might wait in the queue is:

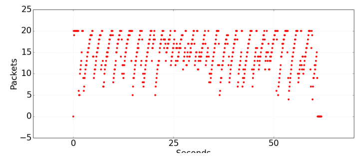
$$\frac{1000 \text{ packets} \times 12000 \text{ bits}}{100 \text{ Mb/s}} \times \frac{1000 \text{ ms}}{1 \text{ s}} = 120 \text{ ms}$$

Results

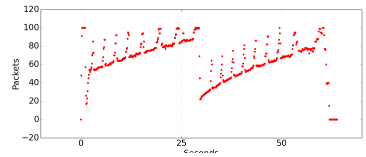
Figure 1 shows varying queue size indicating the queue is being filled. The "sawtooth" pattern of congestion window is apparent for size of 20 and 100 (figure 2).



(a) Q=5

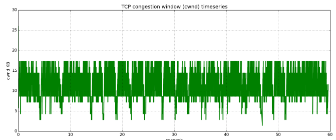


(b) Q=20

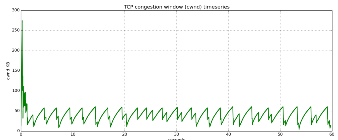


(c) Q=100

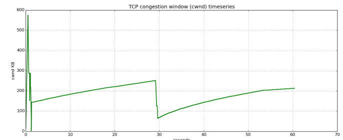
Figure 1: Queue Size



(a) Q=5

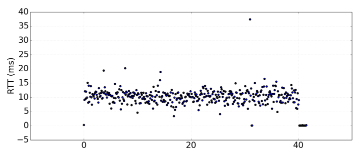


(b) Q=20

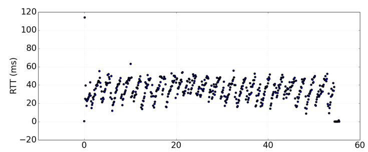


(c) Q=100

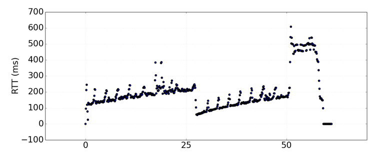
Figure 2: Congestion Window Size



(a) Q=5

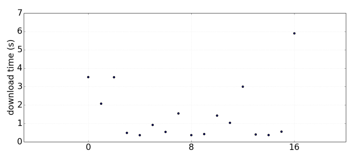


(b) Q=20

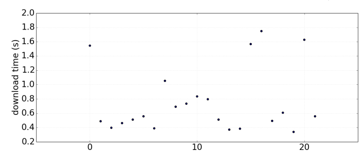


(c) Q=100

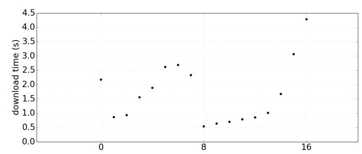
Figure 3: Round Trip Time (RTT)



(a) Q=5



(b) Q=20



(c) Q=100

Figure 4: Webpage Download Time

For the queue size Q=5, the average RTT is around 10ms. For the queue size Q=20, the average RTT is approximately, 30ms. For the queue size Q=100, the average RTT is around 200ms. Thus, it is clearly seen from figure 3 that RTT is linearly correlated with the queue size.

An approximate symbolic equation can be derived from above reports and by observing features of links and router. The total RTT is sum of transmission delay, propagation delay, and queuing delay without all overheads in the server execution. Let Q represent size of the buffer queue of $s0$, then,

$$\begin{aligned}\text{RTT} &= \frac{Q \times \text{packet size}}{s0, h2 \text{ link bandwidth}} + \text{propagation delay} + 2\Big(\frac{\text{packet size}}{h1, s0 \text{ link bandwidth}}\Big) + 2\Big(\frac{\text{packet size}}{s0, h2 \text{ link bandwidth}}\Big) \\ &= \frac{Q \times \text{packet size}}{10Mbps} + 4ms + 2\Big(\frac{\text{packet size}}{1Gbps}\Big) + 2\Big(\frac{\text{packet size}}{10Mbps}\Big)\end{aligned}$$

Furthermore, the effect of bufferbloat is clearly demonstrated in figure 4. The webpage download time is on average 2.0s, 1.0s, and 2.0s for queue sizes Q=5, Q=20, Q=100 respectively. This shows that out of these three possible sizes (5, 20, 100), queue size of 20 is the best option and having larger buffer does not necessarily relieve packet congestion.

The result shows that it is NOT desirable to have buffer size to be too small or too large. There is an optimal size for the buffer for the efficient and effective delivery of packets from one end to another.

Solutions and Mitigations

There are two main categories of methods that can mitigate the effect of bufferbloat effect.

One category of methods operates at the level of buffer of routers. It manages the buffer queue so that it is not filled completely to mitigate packet loss and dramatic increase in latency. One example could be Proportional Integral controller Enhanced (PIE).[2] This works by randomly dropping a packet at the onset of congestion. The probability of dropping a packet at the onset of congestion is determined by latency periods.

The other category of methods implemented to relieve bufferbloat works at the end hosts of network. Instead of relying on TCP packet loss, the end hosts communicates each other to decide the optimal sending rate. One example of this is Bottlenet Bandwidth and Round-trip Time Congestion Control (BBR).[3] BBR works by searching for the optimal sending rate such that the rate of packet delivered to the other end just plateaus - this is called bottleneck delay. Then, the BBR implemented end host ensures the packets are sent at the optimal rate.

References

1. J Gettys, "Bufferbloat: Dark Buffers in the Internet" IEEE Computer Society, pp 95-96, May-June 2011.

2. R. Pan, P. Natarajan, C. Piglione, M. S. Prabhu, V. Subramanian, F. Baker and B. VerSteeg, "PIE: A Lightweight Control Scheme to Address the Bufferbloat Problem" Advanced Architecture Research Group. pp. 148 - 153, 2013.

3. N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "Bbr: Congestion- based congestion control," ACM Queue, vol. 14, pp. 20 - 53, September-October, 2016.