Bufferbloat Report

CSC458 Fall 2021

Introduction

Brief description of the bufferbloat problem and its impact on the users **(a)**

Bufferbloat is a term used to describe a scenario in which excessive buffering in a network can result in long delay periods and irregular packet delivery.

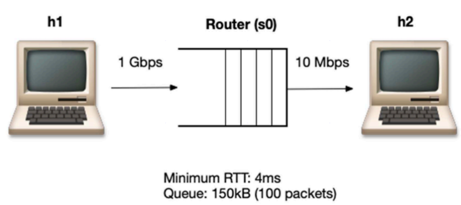
When the rate at which hosts are delivering data momentarily exceeds the network's capacity, a buffering method is utilized to keep the network from being overburdened and requiring packets to be discarded. It lets additional packets simply linger in network buffers to be sent when the network becomes less congested and freer. But it also means that packets must wait longer before being sent.

Additionally, hosts detect network capacity by increasing their transmission rate until packets start getting lost in the network, at which point they recognize they are sending too much and drastically decrease their transmission rate. When a host does not know exactly how much data it may send across the network at any particular moment, it uses a technique called "TCP Sawtooth." Packets are not instantly rejected when using large buffers, however; instead, they are stored there for subsequent transmission, until router transmitting more than the network can manage. After a while, the network's buffers full up and the hosts recognize that they can no longer deliver as much data. As a result, the network becomes ineffective. Users get a worse experience when using the Internet.

Simulation Setup

In order to demonstrate the incidence of bufferbloat, a simulation was developed in Mininet to simulate a simple network architecture including a client, a server, and a router between them. There is a 10 Mb/s connection between the router and the client node, and a 1 Gb/s connection between the router and the server node. We also set the propagation delay of each link to 1ms so that the minimum RTT is 4ms.

With the iperf command, a long-lived TCP flow from h1 to h2 is created for this experiment.



Measure the round-trip time of 10 pings from the client to the server. Measure the time it takes for every two seconds a webpage is downloaded from the server. Choose three level of maximum buffer size: 5, 20, and 100. The simulation is conducted for 60 seconds each time. Result shown below.

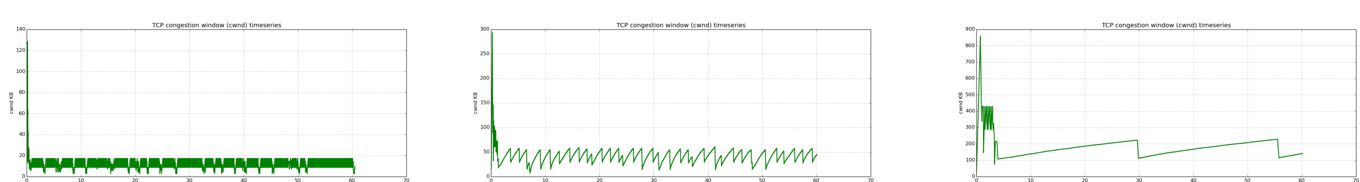
To see the result graph, run “sudo ./run.sh” in the folder.

Results

Though the investigation, basically, buffer float occur when buffer size rise. And as far as the buffer size increase, congestion window drop slower, pinging the server took significantly longer time and downloading webpages to take significantly longer time.

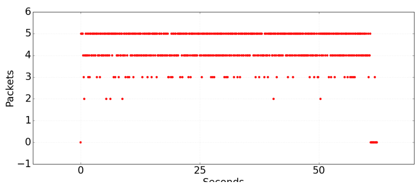
More specific investigation result shown below.

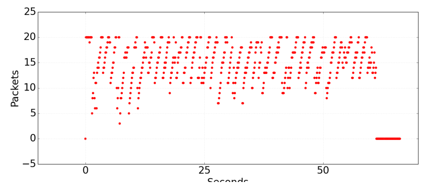
Congestion window size (1)

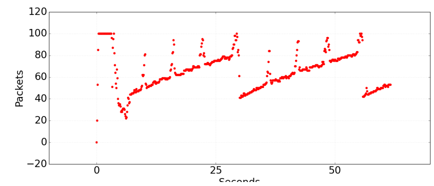


q = 5 q = 20 q = 100

Queue length (2)

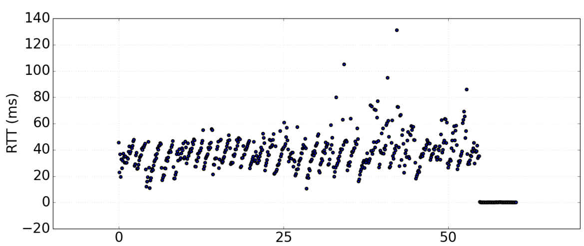


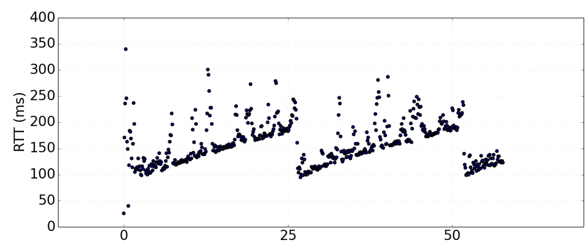




q = 5 q = 20 q = 100

Ping RTT (3)





q = 20 q = 100

As for the **Congestion window size (1)** shown above, for buffer size 5 and 20, graphs show same sawtooth like pattern; while for buffer size 100, the congestion graph show a smoother look. That is because when buffer size is small, the network starts dropping packets sooner than the one with large buffer size, so the graph with smaller buffer size (5 or 20) changing actively and regularly.

As for the **Ping RTT (3)** shown above, the RTT of ping is linearly related to the queue size. Overall, time for ping increases as the maximum queue size increases. Ignoring the overhead in the server execution, RTT is consisted of queueing delay, propagation delay and transmission delay. When buffer size is 20, most part in ping graph changing very low, while when buffer size is 100, we see that the ping times are much higher and changing more actively compared to with a buffer size of 5. That is possibly because of the bufferbloat problem, which causes packet delivery slower and less efficient.

Queue Size	**Average Download Time**	**Standard Deviation Download**
20	0.69	0.24
100	1.21	0.56

As for downloading time, we could see that with smaller queue size, the downloading speed comparatively quicker. That is possible because of the bufferbloat problem, it occurred with the webpage with the larger queue size, causing packets related to webpages get stuck in the large buffer, waiting for a long time instead of being dropped quickly by the network and being resent. From the user’s point of view, the page loading more slowly.

Solution to the bufferbloat problem

1. Why do you see a difference in webpage fetch times with short and large router buffers?

The average fetch times for the short router buffer is shorter than the long router buffer. This is possibly because the short router buffer will have a shorter queue, the number of packets queued is less, as a result, they have a shorter wait time. which means when a new fetching comes into the router buffer with larger buffer size, it has to wait a longer time.

2. Bufferbloat can occur in other places such as your network interface card (NIC). Check the output of ifconfig eth0 on your VirtualBox VM. What is the (maximum) transmit queue length on the network interface reported by ifconfig? For this queue size, if you assume the queue drains at 100Mb/s, what is the maximum time a packet might wait in the queue before it leaves the NIC?

max transmit queue length is 1000

and MTU is 1500 bytes

1000 * 1500 * 8 / (100 * 10 / 6 b/s) = 0.12s.

3. How does the RTT reported by ping vary with the queue size? Write a symbolic equation to describe the relation between the two (ignore computation overheads in ping that might affect the final result).

RTT is proportional to the queue size accroding to the graph. So I'd say RTT = k * quene_size

4. Identify and describe two ways to mitigate the bufferbloat problem.

1. Reduce queue size, have shorter router buffer. which could effectively reduce the wait time and handle TCP congestion control better.

2. Decreasing the timeout value for TCP, which would result in the decrease of the cwnd. since packets stuck in longer queue would be considered dropped.

Reference

(a) F. Casey, "Beginner's Guide to Bufferbloat - How to fix network lag," *Netduma.com*, 18-Jul-2018.[Online]