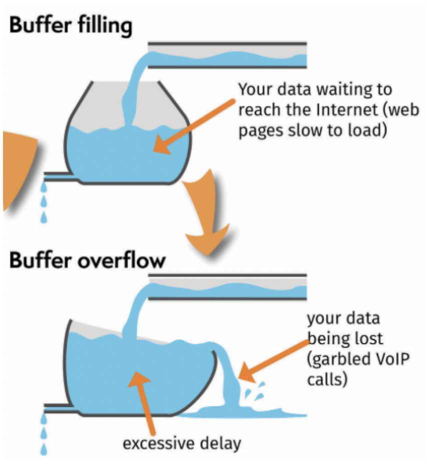
COL724 - Assignment 1

BufferBloat on Mininet

Gurarmaan S. Panjeta
2020CS50426



Report & Logistics

August 28, 2023

August 28, 2023

Report Questions

Q1. Why is there a difference between fetch times?

The fetch times for Buffer size = 20 average around 0.27 seconds, but the ones for Buffer size = 100 shoot up to around 0.73 second, which is an increase of around 3 times! The standard deviation reported in these cases was 0.117 and 0.192 seconds respectively.

This happens because the long-term TCP and the pings fill up the queues in both the cases, but since the 100 sized queue is larger, it takes longer for a packet in that queue to be serviced, and hence the fetch time increased!

0.244	1.025
0.187	0.566
0.210	0.603
0.251	0.862
0.185	0.544
0.206	0.520
0.276	0.811
0.205	0.457
0.193	0.498
0.524	0.796
0.267	0.831
0.230	0.860
0.261	1.065
0.196	0.587
0.200	0.622
0.265	0.876
0.195	0.555
0.244	0.525
0.245	0.808
0.499	0.846
0.254	0.881
0.466	0.756
0.256	0.798
0.195	0.837
0.266	1.038
0.196	0.574

A comparison between fetch times of different buffer sizes : 20 (left) vs 100 (right)

Q2. Bufferbloat in NIC and other places

Using ifconfig on Baadal VM, I get the following statistics (BaadalVM has ens3 instead of eth0):

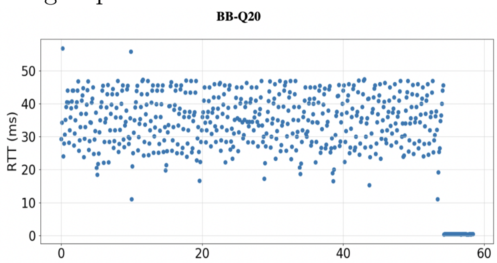
```
baadalvm@baadalvm:~/cs244-bufferbloat$ ifconfig
ens3      Link encap:Ethernet  HWaddr a2:00:00:57:0f:9d
          inet addr:10.17.50.74  Bcast:10.17.255.255  Mask:255.255.0.0
          inet6 addr: fe80::a000:ff:fe57:f9d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:99928149 errors:0 dropped:0 overruns:0 frame:0
          TX packets:63776 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:11217498952 (11.2 GB)  TX bytes:82502990 (82.5 MB)
```

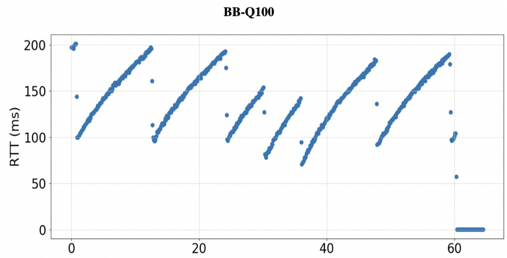
maximum transmit queue length = 1000 and packet size = 1.3Mb

Thus, the maximum transmit queue length is a 1000 and the maximum packet is 1500B (MTU). The maximum time a packet has to wait is when the queue is full, that is, there are a 1000 packets ahead. These are 1000 * 1500 B in size, and the (assumed) queue drain of 100Mb/sec means it takes $\frac{1000*1500*8b}{100Mb/sec} = 0.12 seconds$. Thus, the maximum waiting time in the queue is 0.12 seconds.

Q3. RTT vs Queue size - Observation and Explanation

While The Q = 20 RTT remains rather uniformly scattered around a lower value (around 35 ms), the Q = 100 RTT shows the classic TCP Reno type curves and higher values, periodically from 100 to 200 ms and then a steep fall. The latter happens because the delayed RTT time makes the sender believe the network is congested b y delaying acks, and hence cuts down it's window to half. Further , the RTT times are larger in the Q = 100 case because it takes a larger time for a packet to be serviced because it gets admitted to a longer queue.



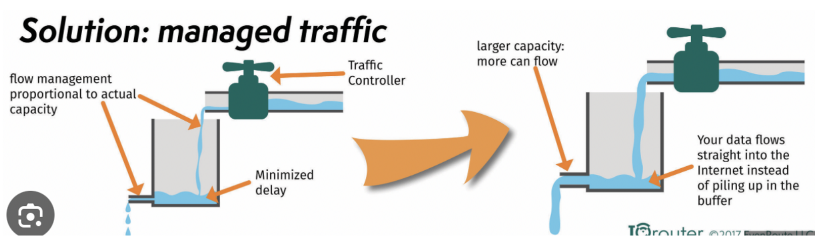


Q4. Mitigate Bufferbloat

There are various strategies we can use to tackle the buffervloat problem:

a) **Reduce Buffer sizes** :- The very origin of bufferbloat is the buffer sizes of routers and switches being too large. This leads to packets having to wait to be serviced, hence leading to delays. Thus, reducing the buffer sizes to a lower value will help in reducing latency, though care should be taken to not lower the value too much and rish under-utilisation.

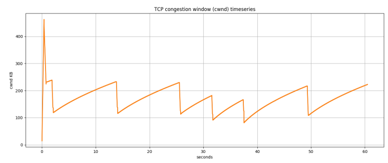
b) **Active Queue Management (AQM)** :- In order to keep queue length control, we can also actively manage the queue of packets waiting for transmission. We monitor the queue's delay and drop packets when necessary to prevent the buffer from filling up. Some popular AQM algorithms are RED(Random Early Detection), PIE (Proportional Integral Controller Enhanced) etc.

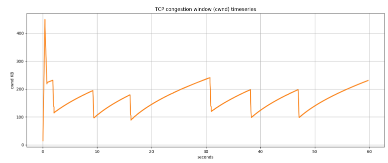


The above image shows how "regulating the traffic" by actively managing queues or tuning buffer sizes solves bufferbloat

Q5. Changes upon Re-Running

The generated graphs and data follow similar trends upon re-runs, albeit there may be a little fluctuation. These micro differences arise because the three flows are operating consecutively, and in different runs of the emulation, may send packets in different orders with respect to each other. This may create subtle differences in the flow pattern, but the overall structure and conclusions remain the same.





The above plots are the maxq = 100 cwnd plots for different runs of the emulation, showing same features with subtle differences