Section 8: More Bufferbloat

CSE 461 21sp

Administrative Things

- Homework 4 due Tuesday, May 25th
- Quiz 4 next Wednesday, May 26th
- Project 3 due Wednesday, June 2nd

Agenda

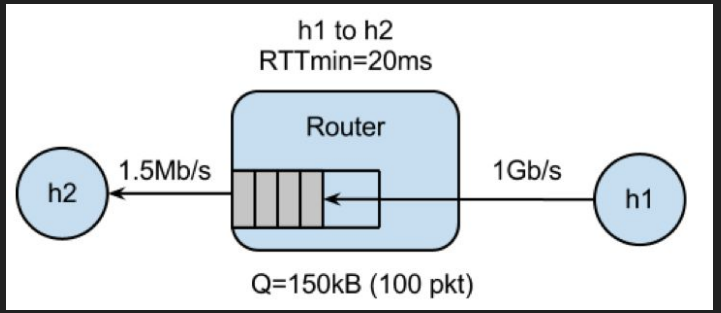
Based on the poll…

- More Project 3, Bufferbloat Content
  - Last week - high level introduction, project setup
  - This week - more detail on the experiments, what you need to do

- Office Hours with time left
  - Lecture questions, project 3, homework 4 all fair game

Topology Setup

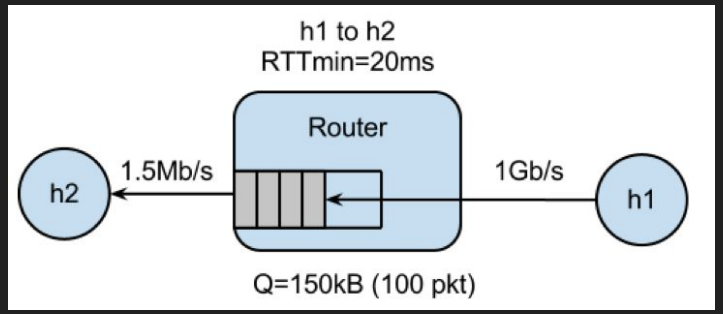
- Very similar to Project 2, Part 1
  - Except need to specify link characteristics (bandwidth, minimum RTT, max queue size)
  - Look into Mininet documentation!



The Experiment

Run ***in parallel***

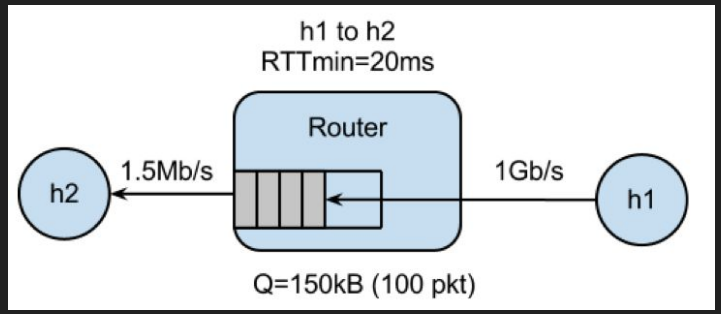
- Long-lived TCP flow between h1 and h2 (iperf/iperf3)
  - Fills bottleneck router
- Ping train between h1 and h2
  - Measure latency between hosts
- Measure time to `curl` down webpage from h1
- Monitor queue occupancy at bottleneck router
  - (done for you)



Goal: See how queue size behaves under congestion, and how that affects latency/download times

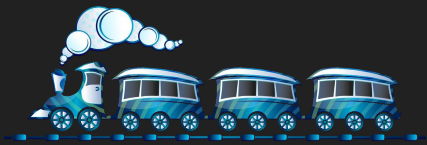
Long-lived TCP Flow

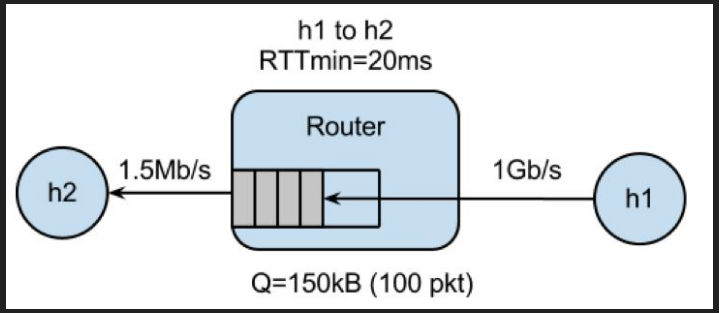
- Starter code sets up iperf server on h2
- Goal: start iperf client on h1, connect to h2
  - Should be “long-lasting”, i.e. for time specified by --time parameter
- How do I connect to a certain IP or make the connection long-lasting?
  - man pages are your friend!
  - type `man iperf` in a Linux terminal



Ping Train

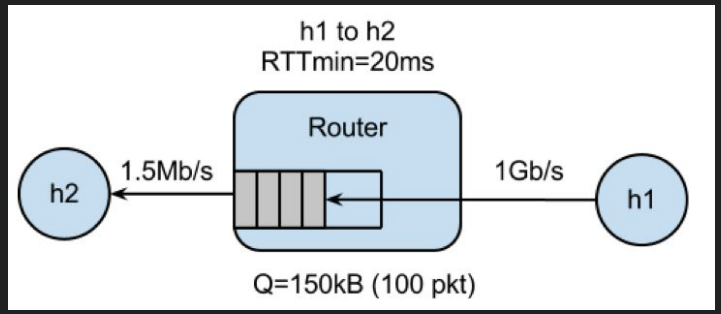
- Goal: Start “ping train” between h1 and h2
  - Pings should occur at 10 per second interval
  - Should run for entire experiment
- How do I specify the ping interval and how long the ping train runs?
  - man pages are your friend!
  - type `man ping` in a Linux terminal
- Write the RTTs recorded from `ping` to {args.dir}/ping.txt
  - See starter code comments for more detail





Download Webpage with curl

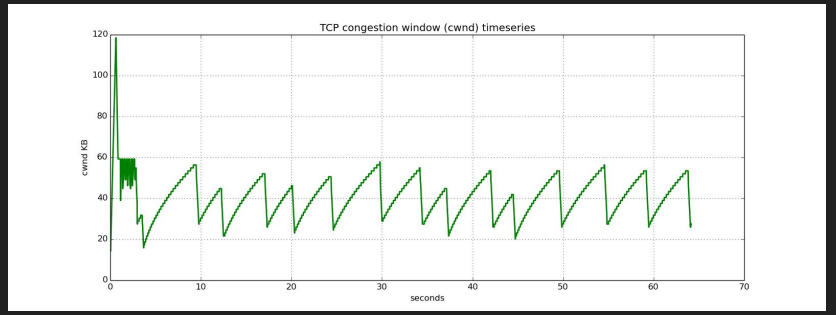
- Starter code spawns webserver on h1
- Goal: Use `curl` to measure fetch time to download webpage from h1
  - Starter code has hint on formatting curl command
  - Make sure `curl` doesn't output an error
    - Errors report very small latency
- No need to plot fetch times



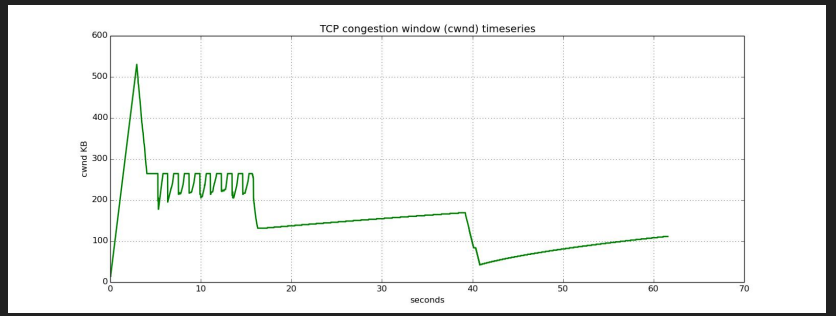
Plotting

- Starter code contains scripts for plotting, `plot_queue.py`, `plot_ping.py`
  - Expects queue occupancy in $dir/q.txt, ping latency in $dir/ping.txt
  - Plots are useful for debugging!
- Part 3, run same experiments with TCP BBR instead of TCP Reno
  - How do you expect the graph outputs to differ?

Q = 20



Q = 100



Any questions?

What is Bufferbloat?

From Wikipedia, “bufferbloat is a cause of high latency in packet-switched networks caused by excess buffering of packets”