COL 724 - Advanced Computer Networks
Assignment 1

Notes:

- This assignment has two parts - Bufferbloat Implementation and Network Modifications.
- Your answers have to be a PDF submission and a tar file containing the mininet code. Do **not** submit handwritten or scanned sheets.
- Due date: 11:59 PM, 24^{th} February, 2021

Mininet Introduction:
The "Introduction to Mininet" tutorial on github (available here) has a number of examples to help you with common tasks you will encounter in this assignment: creating topologies, spawning programs (ping, iperf, etc.) and collecting output.

1. **Buffer-bloat Implementation**
   In this exercise we will study the dynamics of TCP in home networks. Take a look at the Figure 1 which shows a typical home network with a Home Router connected to an end host. The Home Router is connected via Fibre (GPON) or DSL to a Headend router at the Internet access providers office. We are going to study what happens when we download data from a remote server to the End Host in this home network In a real network its hard to measure cwnd (because its private to the Server) and the buffer occupancy (because its private to the router). To make our measurement job easier, we are going to emulate the network in Mininet.

   Goals:

   (a) Learn first-hand the dynamics of TCP sawtooth and router buffer occupancy in a network.
   (b) Learn why large router buffers can lead to poor performance. This is often called "bufferbloat."
   (c) Learn how to use Mininet to run traffic generators, collect statistics and plot them.
   (d) Learn how to package your experiments so its easy for others to run your code.

   Tasks: Within Mininet, create the following topology. Here h1 is your home computer that has a fast connection (1Gb/s) to your home router with a slow uplink connection (10Mb/s). The round-trip propagation delay, or the minimum RTT between h1 and h2 is 4ms. The router buffer size can hold 100 full sized ethernet frames (about 150kB with an MTU of 1500 bytes). Then do the following:

   (a) Start a long lived TCP flow sending data from h1 to h2. Use iperf.
   (b) Send pings from h1 to h2 10 times a second and record the RTTs.
   (c) Plot the time series of the following:

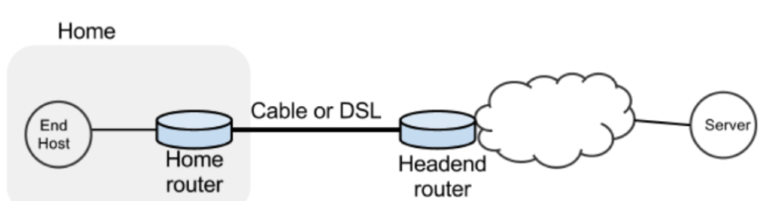

Figure 1: Bufferbloat

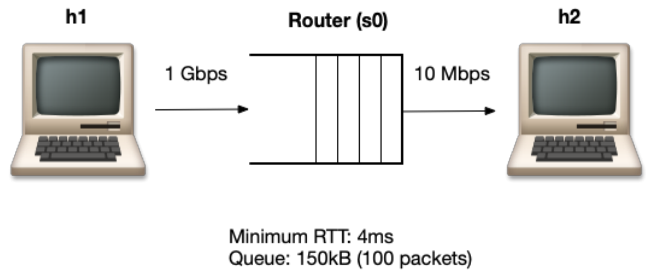

Figure 2: Topology for Bufferbloat

- The long lived TCP flows cwnd.
- The RTT reported by ping.
- Queue size at the bottleneck.

(d) Spawn a webserver on h2. Periodically download the index.html web page using h1 and measure how long it takes to fetch it (on average).

(e) Repeat the above experiment and replot all three graphs after enabling the PIE AQM algorithm on the switch interface attached to the bottleneck link. Set the target delay for PIE to 20 ms.

(f) Repeat the above experiment with PIE enabled, but this time, start 10 parallel TCP flows with iperf.

Framework Code: To help you get started, a basic skeleton code available available here . The following are important files you will find in the repository. Ignore other files.

(a) **bufferbloat.py**: Creates the topology, measures cwnd, queue sizes and RTTs and spawns a webserver.

(b) **plot_queue.py**: Plots the queue occupancy at the bottleneck router.

(c) **plot_ping.py**: Parses and plots the RTT reported by ping.

(d) **plot_tcpprobe.py**: Plots the cwnd time-series for a flow specified by its destination port.

(e) **run.sh**: Runs the experiment and generates all graphs in one go.

Submission/Deliverables: Please submit your codes, README and an assignment report before the scheduled deadline.

2. **Wireless Buffer-bloat**
Broadband and wireless bufferbloat are also the root causes of most of the poor Internet performance seen at many hotels and conferences. In this question, we want you to evaluate the performance gap due to a noisy wireless network. Given the network in Q1, add a WiFi hop in the home network. Specifically, your home network is h1 is now connected to your router via WiFi, which in-turn is *noisy*. As a result, the performance of the network varies depending on the WiFi quality. You must model the impact of noisy WiFi with loss percentages of 0.5, 1, 2, and 4% and also compare with the scenario without WiFi. Repeat the above set of experiments for this scenario as well, you must plot the time series of the following (for all loss %):

   - The long lived TCP flows cwnd.
   - The RTT reported by ping.
   - Queue size at the bottleneck.

   You can make the necessary assumptions but please mention them in the assignment report.

3. **TCP and UDP**
With TCP/IP, during network congestion bufferbloat causes extra delays, limiting the speed of internet connections. Other network protocols are also affected, including UDP-based protocols, partly because they share buffers in the router with TCP/IP connections. In this question we want you to compare the performance of these two protocols. Specifically, in the network in Q1, consider a video stream of UDP and the file transfer over TCP competing over these links and report the values for TCP (as above) an for video jitter, latency, and packet loss. You are allowed to make the necessary assumptions and identify the parameter values but please mention them in the assignment report.