Lab8: Comparing TCP variants using iperf and Wireshark

This lab should be done individually.

In this lab we will compare different TCP variants in terms of their throughput. The setup is shown in the Figure below.

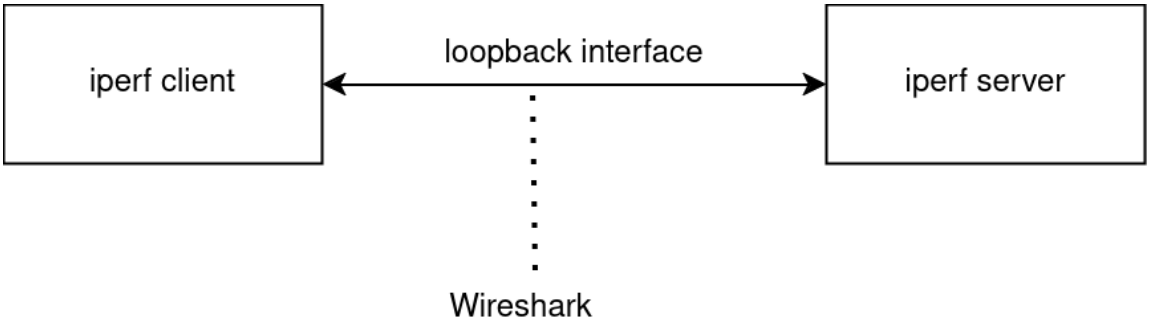

Figure: Lab 8 Setup

You only need a single machine to do this lab. The iperf client transfers a file using TCP over the loopback interface to the iperf server (recall that you have used iperf in Lab 4). We will use different variants of TCP available on your machine. The delay, maximum data rate and packet loss on the loopback interface should be set using “tc” (recall that you have used “tc” in Lab 5). Wireshark will be used to capture packets on the loopback interface and to study the variation in TCP Window over time etc.

Preliminaries

1. Install Wireshark (https://www.wireshark.org/download.html) in case it is not installed (recall that you used Wireshark in Lab 6). To run wireshark on Ubuntu, type the command “sudo wireshark” in a terminal window. It captures packets on a specified interface (you will have to choose the loopback interface for the lab).
2. Study how Wireshark can be used to study “bytes_in_flight” (unACKed bytes) of a TCP flow: (https://www.youtube.com/watch?v=9lJ0vsA40is)
3. Check which TCP protocols are available on your machine:
   `cat /proc/sys/net/ipv4/tcp_available_congestion_control`
4. Check which is the default TCP variant:
   `cat /proc/sys/net/ipv4/tcp_congestion_control`
   Note: you are likely to find Reno and Cubic on the latest Ubuntu OS. Do the experiment on a machine which has more than one TCP variant.
5. Set the default TCP variant using:
   `sudo sysctl -w net.ipv4.tcp_congestion_control=XX`
   (here XX will be `reno` or `cubic`)
6. To check if iperf is indeed using the version of TCP that you want run “`ss -tni`” while transferring a file using iperf. You will have to search for the relevant port numbers used by the iperf server and client and check the TCP variant being used.

7. How to set delay, loss, and bandwidth limit (i.e. rate) on the loopback interface using "tc":
   https://netbeez.net/blog/how-to-use-the-linux-traffic-control/
   Note: you will have to use the loopback interface in the tc command as in Lab 5. You may have to "sudo" the command. If you have set delay/loss etc. using tc once, you will first have to delete the tc rules before you can set new parameters for delay, loss etc. The above page explains how to delete tc rules.
8. To automate the large number of experimental runs you have to do, it is always a good idea to use a script. You should learn about scripting. Example:
   https://ma.ttias.be/bash-loop-first-step-automation-linux/
9. Setting Maximum Transmission Unit (MTU) on the loopback interface to 1500B (1500 bytes): The MTU may be much larger on the loopback interface than on say an Ethernet interface. To mimic the Internet, reduce the MTU to 1500B.
   To find out the MTU on all interfaces: `netstat -i`
   To set the MTU: `sudo ifconfig <loopback interface> mtu 1500`
   Run "`netstat -i`" to check that the MTU has been updated.
10. You will have to draw confidence intervals for graphs. Learn about this from:
    https://www.mathsisfun.com/data/confidence-interval.html (see Step 2 on this page)

EXPERIMENTS

In all experiments, **you will transfer a text file of size 20 MB** (20 mega bytes) from client to server. For each experiment described below, calculate the mean and standard deviation of the obtained throughput for 20 runs. Note that the iperf client and server may give slightly different throughputs for the same data transfer. You can use the throughput given by any one of them. Write scripts to automate the process. You should not have to manually run the experiment 20 times. Also, the calculated throughput information etc. should be automatically saved (or appended) to a file each time you run. You should not do this manually.

Each experiment uses a combination of Delay, Loss (set using tc and netem) and TCP variant. We will have a total of 18 experiments (3x3x2). Note that each of these must be run 20 times. The different choices for Delay, Loss, and TCP variant are given below.

Delay=10ms, 50ms, 100ms (this is a constant egress delay, not a random delay)

Loss=0.1%, 0.5%, 1% (percentage of packet loss)

TCP variant: Reno, Cubic

For all experiments, set the maximum data rate (i.e. bandwidth) on the loopback interface to 100 Mbps (use "tc" and "netem" for this) and don't forget to set the MTU on the loopback interface to 1500 bytes. The "burst" parameter can be kept to 32kbit or higher.

In your report give the results in the form of plots and comment on them. Draw error bars of 90% confidence intervals around the mean. Feel free to use any graphic tool (The online tool Desmos allows plots of error bars).

Plot 1: (Loss=0.1%) Throughput (y-axis) vs. Delay (x-axis) for both Reno and Cubic

Plot 2: (Loss=0.5%) Throughput vs. Delay for both Reno and Cubic

Plot 3: (Loss=1%) Throughput vs. Delay for both Reno and Cubic

Plot 4: (Delay=10ms) Throughput vs. Loss for both Reno and Cubic

Plot 5: (Delay = 50ms) Throughput vs. Loss for both Reno and Cubic

Plot 6: (Delay = 100ms) Throughput vs. Loss for both Reno and Cubic

Cubic is supposed to be more aggressive (has faster window increase during congestion avoidance) than Reno. **In your report** comment on whether the observed throughputs confirm this or not. Comment on how throughput changes with increasing Delay for both TCP variants. Comment on how throughput changes with increasing packet loss for both TCP variants.

In your report show the Window Scaling graphs (which shows bytes_in_flight) for both TCP variants from Wireshark for one run when Delay = 10ms, Loss=0.1%. The graph should cover a time window of at most 5 seconds. Do the same for the case Delay = 100ms, Loss=1%. Annotate the plots for Reno to show at least one region where the TCP was in Slow Start and at least one region where TCP was in Congestion avoidance. Mark any obvious locations of packet loss on the graphs. Do the same for the plots of the Cubic variant.

Upload only one tar file to Moodle. The file name should be <rollno>-lab8.tgz. The file should contain:

1. (5 marks) README file explaining how scripts are to be run

2. (10 marks) Scripts to automate running of experiments (the more automated, the better. For example, if you set the TCP variant, delay/loss/bandwidth on loopback interface in the script itself, then expect more marks)

3. (25 marks) Report