GENI Lab 6

Due: 3:30pm October 11, 2018

0. Introduction

0.1. Overview

We stay on the transport layer in this lab.

This lab allows students to experiment with **how RTT (Round Trip Time) and TCP window size affect TCP throughput**. You will learn how to adjust the RTT of a TCP connection by adding delay, as well as how to adjust the window size of an `iperf` TCP flow.

When not limited by the transmission speed of a link, TCP's theoretical maximum throughput is:

$$Theoretical\ Maximum\ Throughput = \frac{TCP\ window\ size}{RTT}$$

From the formula above, you can clearly see that `TCP window size` determines how much data are transmitted whereas `RTT` determines how much time has to be spent for transmitting them and for the Acknowledgment segment to come back.

0.2. Objectives

Upon completing this module you will:

- Be able to adjust the delay (and consequently RTT) on an interface
- Be able to start flows with custom window sizes
- Have a feel for how window size and RTT can affect TCP throughput

0.3. Key Commands

`ifconfig`, `ping`, `tc`, and `iperf`

1. Lab Configurations

Create a new slice (please name the new slice "**lab6-your-full-name**") and request resources that include two connected nodes like what we had in Lab 0.

> If you don't remember how, check back the instructions for lab 0.

Be sure you have one ssh terminal to each of the two nodes.

2. Lab Details, Part I: Adding and Removing Delay to Affect RTT

- First, determine the approximate Round Trip Time (RTT) between your nodes using the `ping` tool. In the `server` SSH terminal, type:

  ```
  ping client
  ```

  Note the "**time=**" value at the end of each line, which is the approximate RTT.

- Type Ctrl-c on the server SSH terminal to stop the pinging.

- Determine the name of the data plane interface on the server node using:

  ```
  ifconfig
  ```

  You will use this interface name (e.g. "eth1") in the commands below.

- Linux offers a rich set of tools for controlling traffic. We will utilize the `tc` (Traffic Control) tool here to add a fixed amount of delay to all packets leaving the given local interface. Issue the following command on the `server` :

  ```
  sudo tc qdisc add dev <interface name> root netem delay 100ms
  ```

  where <interface name> needs to be replaced with the interface name determined in the step above. This command adjusts the traffic control (`tc`) queuing discipline (qdisc) by adding (add) a network emulation (netem) delay of 100 milliseconds to the given device at it's root.

- Run ping again:

  ```
  ping client
  ```

  The RTT ("time=") should now be 100ms longer. Type Ctrl-c to stop the pinging.

- Now, remove the delay with the following command:

  ```
  sudo tc qdisc del dev <interface name> root
  ```

  where <interface name>, again, needs to be replaced with the interface name. This deletes (del) the current queuing discipline from the given device's root.

  Note: **To change the delay on an interface, you must always first delete the queuing discipline and then add a new one**.

- Run ping yet again.

  ```
  ping client
  ```

  The RTT ("time=") should return to approximately what it was before adding delay. Type Ctrl-c to stop the pinging.

3. Lab Details, Part II: Adjusting the TCP Window Size

> `iperf` is a tool for conducting network throughput tests and measuring TCP and UDP bandwidth performance. Run `iperf` on both the server and client nodes to check its availability. If no, install it by running the following command:
>
> ```
> sudo apt-get install iperf
> ```

- In the server SSH terminal, start the iperf server by typing:

  ```
  iperf -s
  ```

- In the client SSH terminal, start the iperf client by typing:

  ```
  iperf -c server -t 10
  ```

  This command starts `iperf` in client mode ("-c"), opens a TCP connection to the specified hostname ("server"), and begins sending packets using the default window sizes. The client will stop after 10 seconds ("-t 10").

- Once the client is done, type "Ctl-c" in the server SSH terminal to kill the server. Note the average

bandwidth value, which is printed in the client terminal. It is also considered as the **actual throughput value**.

- We will now repeat the procedure but we will limit the window size on both the client and server. Note that Linux **automatically doubles the window size** set by `iperf`, so the commands below request a 2KB (2 kilobytes) window to obtain a 4KB (4 kilobytes) window size. Also, notice that the unit of the window size here is **`KB` or kilobytes**, which might need to be converted to **`kb` or kilobits** for further comparisons (1 byte = 8 bits). In the server SSH terminal, type:

```
iperf -s -w 2KB
```

- In the client SSH terminal, type:

```
iperf -c server -t 10 -w 2KB
```

- Once the client is done, type "Ctrl-c" in the server SSH terminal to kill the server. Note the average bandwidth value in the client window. The value should be significantly less than it when the default window size was used.

Now, open **lab6-practice-worksheet**, do the experiments, and answer the questions there. Notice this is **not** the work that I want you to submit though.

Finally, open **lab6-worksheet**, do the experiments, answer the questions, and submit the completed worksheet on BlazeVIEW.

4. What to Turn in?

Submit the following file:

- lab6-worksheet.docx