Cpr E 489: Lab Experiment #8: TCP Congestion Control Basics
(Total Points: 100)

Objective

This lab is designed to allow you to develop an experiment testing the basic functions of TCP congestion control.

Pre-Lab

Create a slice in the GENI portal,

1) Go to https://portal.geni.net/ press the **Use GENI** button and from the **Drop-Down** menu select your institution
2) Create a slice in the previously joined Cpr E 489 Project
3) Generate a private SSH key

Lab Expectations

Please carefully read the following important information, regarding the new format of Lab 8 in Week 15.

1) All students will **work on Lab 8 on their own** instead of as a team.
2) You may work on the lab **ANY time at your convenience.**
3) **The TAs will be online during the regular lab hours** to help you and for Q&A:
    a. TA – Ethan Shoemaker will be available on Zoom (meeting ID: 978 354 397) on 4/21 and 4/28 from 2:10 PM to 6:00 PM.
    b. TA – James Bonner will be available on Zoom (meeting ID: 127 304 414) on 4/22 and 4/29 from 1:10 to 3:00 PM, and 4/24 and 5/1 from 1:10 to 3:00 PM.
4) Outside the regular lab hours, if you have any questions, you may contact the TAs via email or schedule an individual online meeting with the TAs.
5) After the lab, write up a lab report. Be sure to:
    - **Summarize** what you learned in a few paragraphs **(20 points)**.
    - Include the required **screenshots (20 points)** and your **answers** for the exercise (**60 points)**.
6) **Lab 8 report is due on 5/5 Tuesday 11 AM CST for all sections. Each student please submit a copy of your own lab report on Canvas.**

Procedure:

a. In the GENI Portal, create a slice and name it: **LastName-Lab8**

b. Then click **Add Resources**. Scroll down to where it says **Choose RSpec**, select the "**URL**" option, and load the RSpec from the URL: **https://git.io/vSioM**

c. Click on **Site 1** and choose an **InstaGENI** site to bind to, then **reserve** your resources. Wait for your nodes to boot up (they will turn green in the canvas display on your slice page in the GENI portal when they are ready). Then, **SSH into each node** (see notes section at the end for how to SSH on a Windows machine). Notice: if you cannot SSH into your GENI nodes after they have a status of ready, delete the resources from your slice, load the RSpec and choose a different aggregate for **Site 1.**

d. On the end hosts (**sender** and **receiver**), install the iperf network testing tool, with the command

```
sudo apt-get update
sudo apt-get install iperf
```

e. Also set TCP Reno as the default TCP congestion control algorithm on both, with

```
sudo sysctl -w net.ipv4.tcp_congestion_control=reno
```

f. On the router, turn on packet forwarding with the command

```
sudo sysctl -w net.ipv4.ip_forward=1
```

g. Also, set each experiment interface on the **router** so that the router is a 1 Mbps bottleneck, and buffers up to 0.1 MB, using a token bucket queue:

```
sudo tc qdisc replace dev eth1 root tbf rate 1mbit limit 0.1mb burst
32kB peakrate 1.01mbit mtu 1600

sudo tc qdisc replace dev eth2 root tbf rate 1mbit limit 0.1mb burst
32kB peakrate 1.01mbit mtu 1600
```

h. Finally, on the **sender** host, load the `tcp_probe` kernel module, and tell it to monitor traffic to/from TCP port 5001:

```
sudo modprobe tcp_probe port=5001 full=1
sudo chmod 444 /proc/net/tcpprobe
```

i. We will use this TCP probe module to monitor the behavior of the TCP congestion control algorithm on the sender.

Generate Data

j. Next, we will generate some TCP flows between the two end hosts and use it to observe the behavior of the TCP congestion control algorithm.

k. On the **receiver**, run

```
iperf -s
```

l. On the **sender**, run

```
dd if=/proc/net/tcpprobe ibs=128 obs=128 | tee /tmp/tcpprobe.dat
```

m. Open another SSH session to the **sender** and run (for PuTTY users: right click the top bar of the sender client window and select **Duplicate Session**)

```
iperf -t 60 -c receiver -P 3
```

You will see a final status report in the `iperf` window after the `iperf` sender finishes, which should take about a minute. In the window where the TCP probe is running, you should see a line of output for each TCP packet.
Take two screenshots: one is for the output of iperf which runs on the receiver and will be complete after the send operation finishes, and another is for the output of the tcpprobe which runs on the sender.
(10 points per screenshot, 20 points in total)

n. The output of the TCP probe will be saved in `/tmp/tcpprobe.dat` on the "sender". Exit the SSH and use `scp` to transfer this file to your own machine for processing:

```
scp -i <private_key> -P <port> username@hostaddress:file dest_location
```

On Windows, this can be done by opening a Command Prompt by hitting the Windows key and typing "cmd <enter>". The private key here must be the actual private key, not the PuTTY key. The command may look like:

```
scp -i Downloads\id_geni_ssh_rsa -P 29412 <netid>@<hostname>:/tmp/tcpprobe.dat ./
```

The TCP module probe records a line of output every time a packet is sent, if either the destination or source port number in the TCP packet header is 5001 (since we loaded it with the *port=5001* option).

Each line of output will include the following fields, in order:

Field	**Explanation**
Time	Time (in seconds) since beginning of probe output
Sender	Source address and port of the packet, as IP:port
Receiver	Destination address and port of the packet, as IP:port
Bytes	Bytes in packet
Next	Next send sequence number, in hex format
Unacknowledged	Smallest sequence number of packets send but unacknowledged, in hex format
Send CWND	Size of send congestion window for this connection (in MSS)
Slow start threshold	Size of send congestion window for this connection (in MSS)
Send window	Send window size (in MSS). Set to the minimum of send CWND and receive window size
Smoothed RTT	Smoothed estimated RTT for this connection (in ms)
Receive window	Receiver window size (in MSS), received in the lack ACK. This limit prevents the receiver buffer from overflowing, i.e. prevents the sender from sending at a rate that is faster than the receiver can process the data.

By looking at your TCP probe data, you can observe the behavior of the TCP congestion control algorithm.

Notes for SSHing on a Windows Machine:

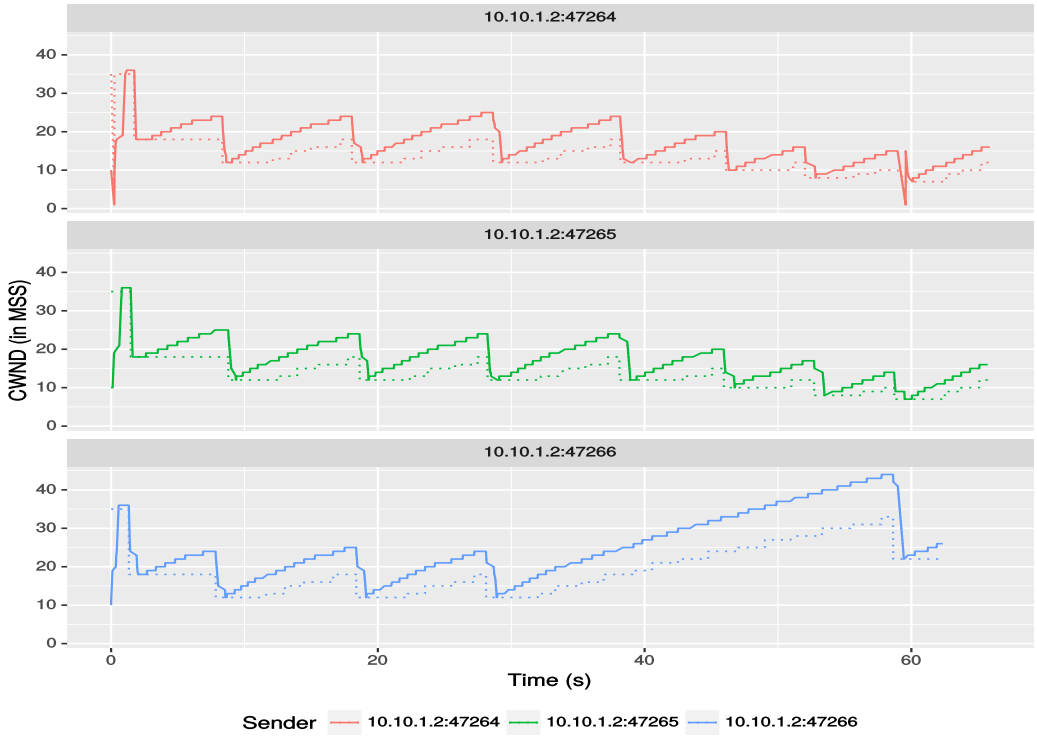
Instead of downloading the private key from GENI as you normally would, instead download the PuTTY key as well as the PuTTY client. You can set up a session for easy rejoining by doing the following:

1. Open PuTTY and make sure you are in the **Session** options by clicking **Session** in the left sidebar of PuTTY. In the **Host Name** box, put <netid>@<host_name_from_GENI>.
2. Change the port to the appropriate port number GENI tells you (note this may be different for each node and if no port is listed, then either leave it blank or select port 22).
3. In the **Saved Sessions** box, put "sender"/"receiver"/"router" so you know which node it is that this saved session is for.
4. Under **Connection** in the left sidebar, expand SSH by clicking the + next to it and select **Auth**.
5. In the **Private key file for authentication:** box, click **Browse** and find your downloaded PuTTY key, it should be something like "id_geni_ssh_rsa.ppk". Hit **Open**.
6. Now return to the **Session** options and verify all the settings are correct. Hit **Save** on the right side of the window.

7. Repeat for all nodes in this lab (this can be simplified by loading the first session you made and changing only the Host Name and/or port number, and the name for the new session before saving).
8. To open a connection to a saved session, simply check the **Saved Sessions** box, select the node name you would like to connect to, click **Load**, and then hit **Open** on the bottom.

Exercise

1. Create a plot of the congestion window size and slow start threshold for each TCP flow over the duration of the experiment, similar to the figure to the right **(10 points)**.



2. For each flow, annotate your plot to show the following: **(10 points per flow, 30 points in total)**

- Periods of **Slow Start**
- Periods of **Congestion Avoidance**
- Instances where 3 duplicate ACKs were received (which will trigger **Fast Recovery**)
- Instances of **Timeout**

3. Using your plot and/or experiment data, explain how the behavior of TCP is different in the **Slow Start** and **Congestion Avoidance** phases **(10 points)**. Also, using your plot, explain what happens to both the congestion window and the slow start threshold when 3 duplicate ACKs are received **(10 points)**.

Cleanup Experiment

After you are done with your experiment, you should always release your resources so that other experimenters can use the resources. In order to cleanup your slice:

Press the **Delete** button in the bottom of the Manage Resources panel on the Slice page.

Wait a few moments for all the resources to be released and you will have an empty canvas again. Notice that your slice is still there. There is no way to delete a slice. It will be removed automatically after its expiration date but remember that a slice is just an empty container, which doesn't take up any resources.