TCP Performance Measurements over GENI
CS 655: Introduction to Computer Networks
Fall 2020

Purpose:
The purpose of this experiment is to evaluate the performance of TCP over different network conditions using GENI.

Prerequisites:
Before beginning this experiment, you should be prepared with the following:

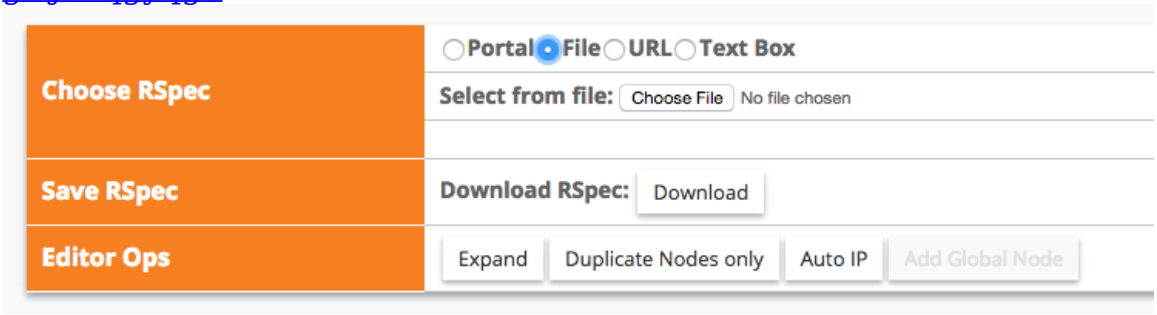
- You have GENI credentials to obtain GENI resources.
- You are able to reserve GENI resources.

- Create a new slice using the GENI Portal. Give your slice a name, e.g. GENI-TCP-*YourLastName*.

- After your slice is created, click the “Add Resources” button as shown below.

- In the “Choose RSpec” section as shown below, first check “URL”, then copy this link: https://piazza.com/class_profile/get_resource/ke6gp3q37pz50y/k

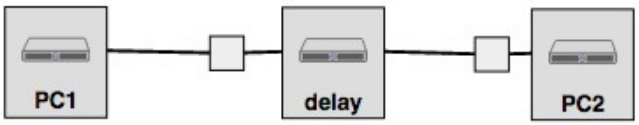
gtrjbnqgyqgb .



- Choose any InstaGENI aggregate for the “site” and click the “Reserve Resources” button at the bottom to reserve resources.

- After the resources are successfully reserved, you should be able to see a topology as shown below (at the bottom of the newly opened webpage), as well as details of these resources.

Note: It may take several minutes before the resources are ready to log in.



Install Iperf:

Iperf is a tool to measure maximum TCP/UDP bandwidth (throughput), allowing the tuning of various parameters and characteristics. Iperf reports bandwidth, delay jitter, datagram loss. Follow the following steps to install Iperf:

- Log into node PC1 and node PC2 in separate windows.

- Install Iperf on PC1 and PC2 using the following command:

  sudo apt-get install iperf

Setting the TCP version:

First we will change the TCP congestion-control algorithm to TCP Reno:

- To check the TCP version running on PC1 and PC2, type the following in the command line window for PC1 and PC2:

```
cat /proc/sys/net/ipv4/tcp_congestion_control
```

- You should see the TCP version running on your VM. If it is not 'reno', type the following to change it to Reno:

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

- Also make sure that you are using Selective Acknowledgement (sack) for TCP. To check if you are using sack, type the following on PC1 and PC2:

```
cat /proc/sys/net/ipv4/tcp_sack
```

If you get output '1', that means TCP sack is enabled. If you get output '0', that means TCP sack is not enabled. To enable TCP sack, type:

```
sudo sysctl net.ipv4.tcp_sack=1
```

Running Iperf:

We will use PC2 to run an iperf server, and PC1 to run the client.

- Run the iperf server on PC2 using the following command:

```
iperf -s
```

You should see:

```
qiaobinf@pc2:~$ iperf -s
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
```

- Run the iperf client on PC1 to connect to PC2:

```
iperf -c pc2 -t 50
```

You should see an output on the console that looks like this:

```
qiaobinf@pc1:~$ iperf -c pc2 -t 50
------------------------------------------------------------
Client connecting to pc2, TCP port 5001
TCP window size: 85.0 KByte (default)
------------------------------------------------------------
[  3] local 10.10.1.1 port 47016 connected with 10.10.2.2 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-50.0 sec  3.44 GBytes   591 Mbits/sec
```

We can see that 591 Mbits/sec is the measured bandwidth (throughput). We are interested in this value.

Adjusting link characteristics:

In this experiment, you'll be changing the characteristics of the link and measuring how they affect TCP performance. We will be manipulating two link characteristics in this experiment: delay and packet loss rate (plr).
It is possible to adjust the parameters of the two directions of your link separately, to emulate asymmetric links. In this experiment, however, we are looking at symmetric links, so we'll always change the settings in both directions together.

- Log into your 'delay' node as you do with any other node.
- We will be using Linux Traffic Control (**tc**) to manage and manipulate the transmission of packets. **tc** allows you to specify different link parameters.
- For example, to set your link parameters to 5ms delay and 0.01% packet loss (plr = 0.0001), type the following on the 'delay' node.

  ```
  sudo tc qdisc add dev eth1 root netem delay 5ms loss 0.01%
  ```

  Now specify the same link parameters in the opposite direction.

  ```
  sudo tc qdisc add dev eth2 root netem delay 5ms loss 0.01%
  ```

- To see the link parameters, type

```
sudo tc qdisc show
```

You should see an output on the console that looks like this:

```
qiaobinf@delay:~$ sudo tc qdisc show
qdisc pfifo_fast 0: dev eth0 root refcnt 2 bands 3 priomap  1 2 2 2 1 2 0 0 1 1 1 1 1 1 1 1
qdisc netem 8001: dev eth1 root refcnt 2 limit 1000 delay 5.0ms loss 0.01%
qdisc netem 8002: dev eth2 root refcnt 2 limit 1000 delay 5.0ms loss 0.01%
```

The last two lines show the link characteristics with 5ms delay and 0.01% loss (plr=0.0001) in each direction.

- To delete the **tc** rules from the interfaces, type the following commands.

```
sudo tc qdisc del dev eth1 root netem
sudo tc qdisc del dev eth2 root netem
```

Note that you need to remove previous tc rules before you can specify new rules on an interface.

Throughput:

Theoretically, the throughput of a TCP connection can be approximated by:

$$Throughput = \frac{1.22\,MSS}{RTT\sqrt{plr}}$$

where the Maximum Segment Size (MSS) = 1460 Bytes (1460x8=11680 bits) on a link with MTU of 1500 Bytes, and the **RTT value can be calculated using the Ping application**. This analytical model captures loss detection by duplicate

acknowledgments but not due to retransmission timeouts (see lecture notes).

We will use the iperf application to measure the TCP throughput for different link conditions, and compare the values of measured throughput with the analytical throughput values calculated using the equation above.

What to hand in:

- Measure the values of throughput using **iperf** for different values of packet loss and delay. **Fill the table below using the measured values as well as the calculated values.**

(Values in parenthesis are calculated values of throughput you get from the equation above.)

	loss=0.01% (plr= 0.0001)	**loss=0.1% (plr=0.001)**	**loss=1% (plr=0.01)**
Delay: 5 ms	_____ Mb/s (Mb/s)	_____Mb/s (Mb/s)	_____Mb/s (Mb/s)
Delay: 10 ms	_____Mb/s (Mb/s)	_____Mb/s (Mb/s)	_____Mb/s (Mb/s)

- Comment on how close your measured throughput values are to their analytical counterparts.

- Try at least two higher packet loss values (e.g., 2% and 5%) and comment on the validity of the above analytical model.

Submission:

Submit a PDF file with your answers on Gradescope.