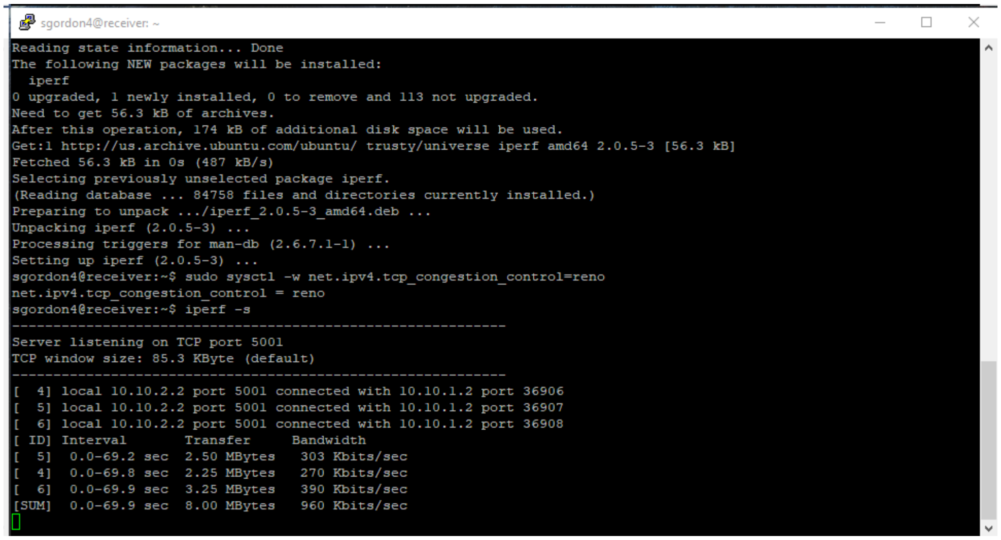
CprE 489, Section 4
Lab Experiment #8: TCP Congestion Control Basics

Sean Gordon

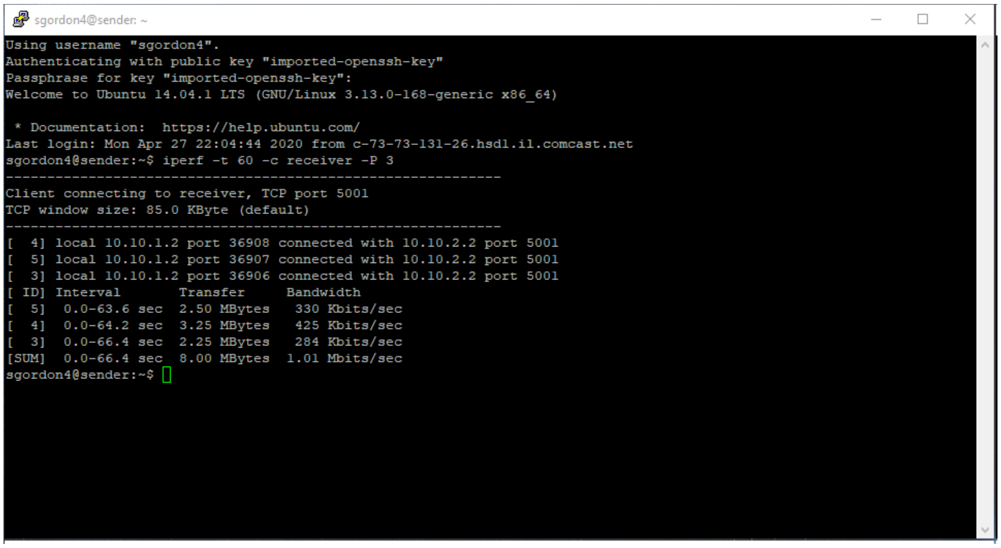
Experience:

This lab built off of what we have recently learned about tcp protocols over the last few days. Using GENI, we simulated sending many packets and saved the data on them to analyze. The analysis focused on looking for the intervals of slow start up, fast recovery, and congestion avoidance.

Data Generation:



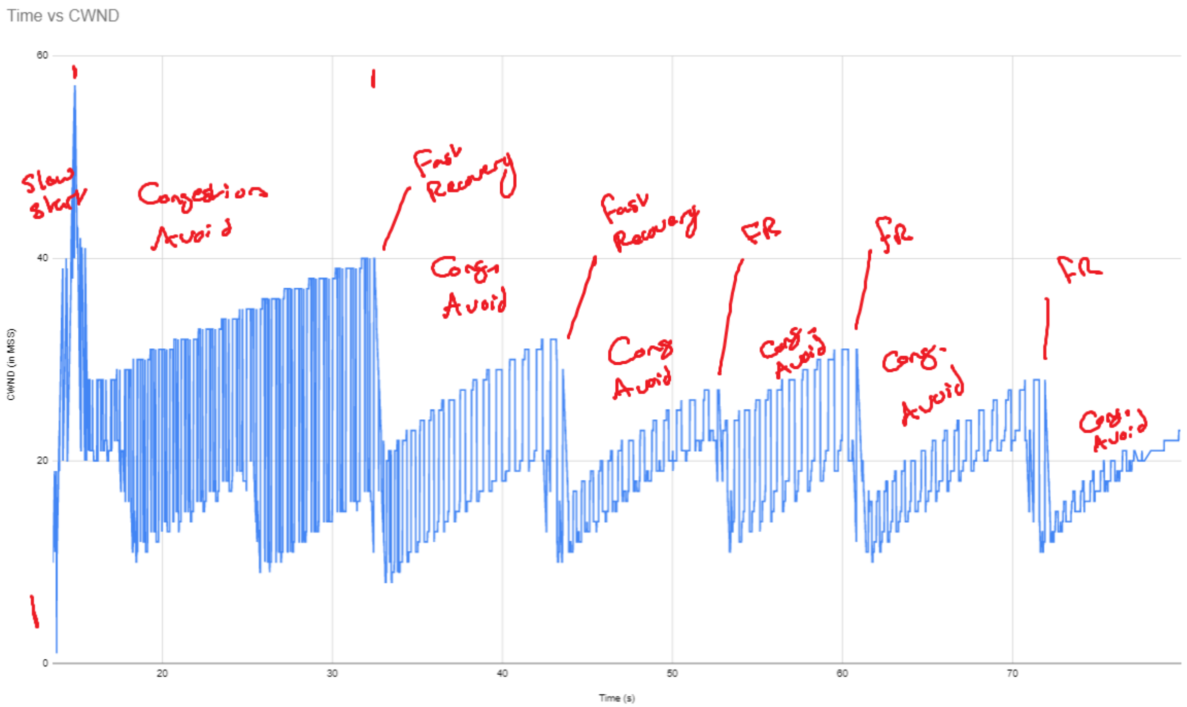
```
Reading state information... Done
The following NEW packages will be installed:
  iperf
0 upgraded, 1 newly installed, 0 to remove and 113 not upgraded.
Need to get 56.3 kB of archives.
After this operation, 174 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu/ trusty/universe iperf amd64 2.0.5-3 [56.3 kB]
Fetched 56.3 kB in 0s (487 kB/s)
Selecting previously unselected package iperf.
(Reading database ... 84758 files and directories currently installed.)
Preparing to unpack .../iperf_2.0.5-3_amd64.deb ...
Unpacking iperf (2.0.5-3) ...
Processing triggers for man-db (2.6.7.1-1) ...
Setting up iperf (2.0.5-3) ...
sgordon4@receiver:~$ sudo sysctl -w net.ipv4.tcp_congestion_control=reno
net.ipv4.tcp_congestion_control = reno
sgordon4@receiver:~$ iperf -s
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  4] local 10.10.2.2 port 5001 connected with 10.10.1.2 port 36906
[  5] local 10.10.2.2 port 5001 connected with 10.10.1.2 port 36907
[  6] local 10.10.2.2 port 5001 connected with 10.10.1.2 port 36908
[ ID] Interval       Transfer     Bandwidth
[  5]  0.0-69.2 sec  2.50 MBytes   303 Kbits/sec
[  4]  0.0-69.8 sec  2.25 MBytes   270 Kbits/sec
[  6]  0.0-69.9 sec  3.25 MBytes   390 Kbits/sec
[SUM]  0.0-69.9 sec  8.00 MBytes   960 Kbits/sec
```

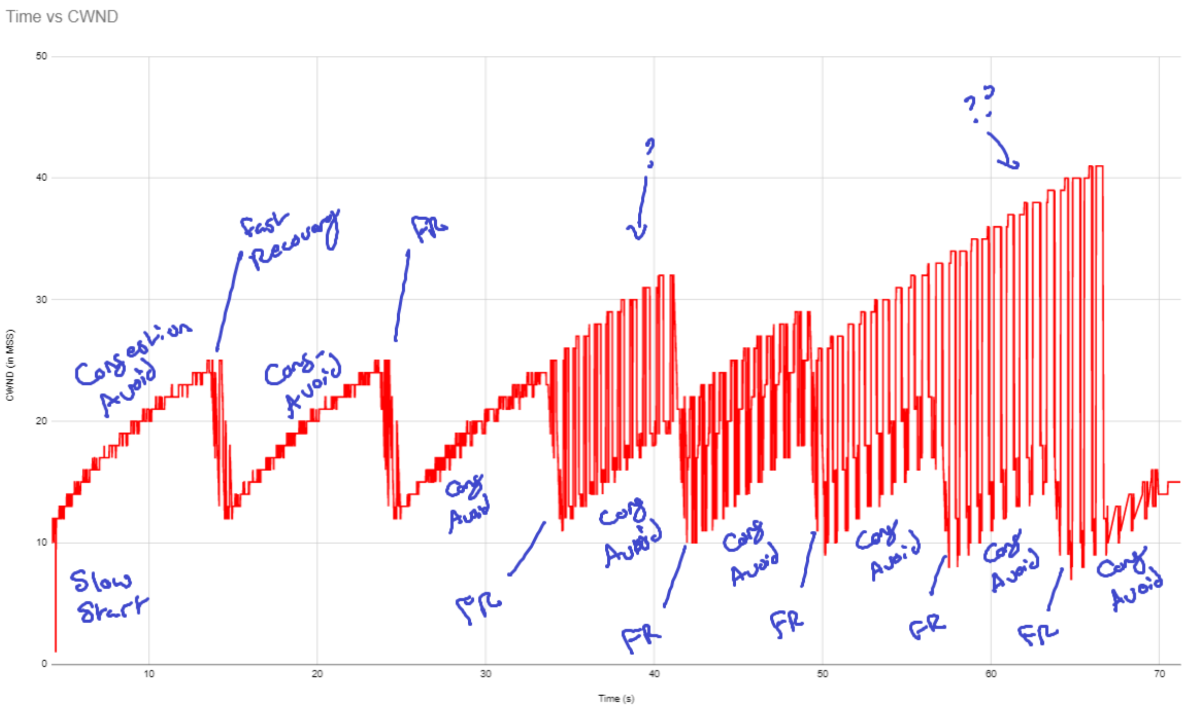


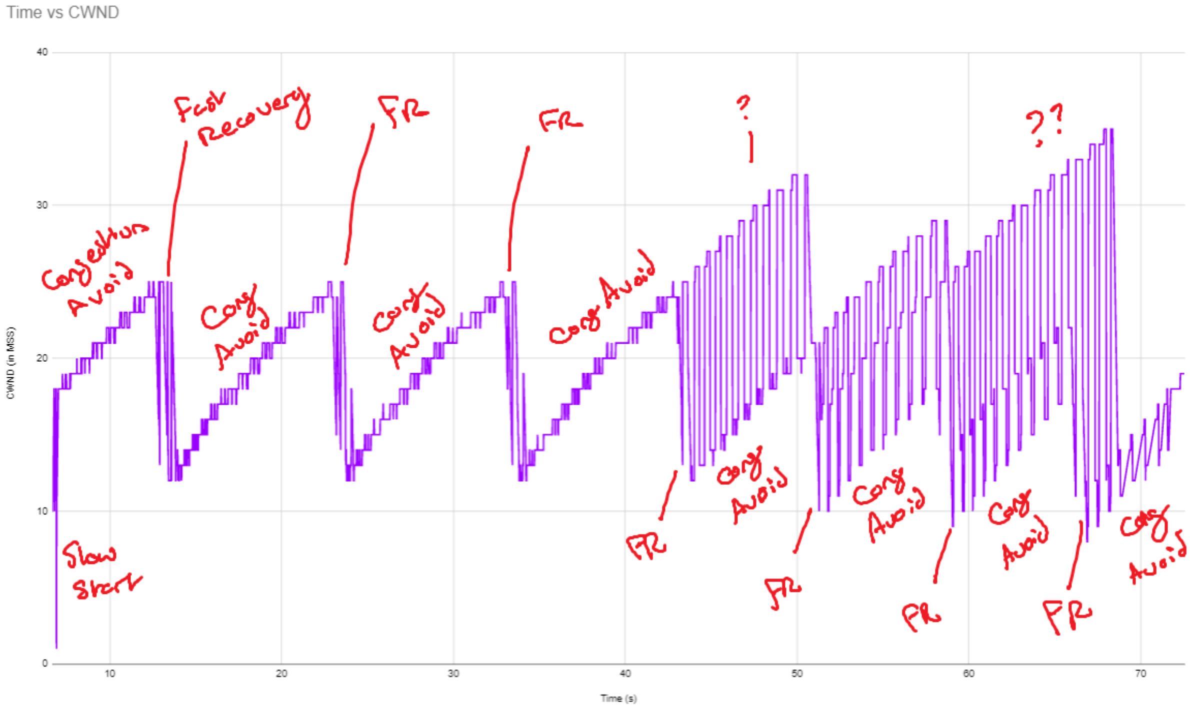
```
Using username "sgordon4".
Authenticating with public key "imported-openssh-key"
Passphrase for key "imported-openssh-key":
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-168-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
Last login: Mon Apr 27 22:04:44 2020 from c-73-73-131-26.hsdl.il.comcast.net
sgordon4@sender:~$ iperf -t 60 -c receiver -P 3
------------------------------------------------------------
Client connecting to receiver, TCP port 5001
TCP window size: 85.0 KByte (default)
------------------------------------------------------------
[  4] local 10.10.1.2 port 36908 connected with 10.10.2.2 port 5001
[  5] local 10.10.1.2 port 36907 connected with 10.10.2.2 port 5001
[  3] local 10.10.1.2 port 36906 connected with 10.10.2.2 port 5001
[ ID] Interval       Transfer     Bandwidth
[  5]  0.0-63.6 sec  2.50 MBytes   330 Kbits/sec
[  4]  0.0-64.2 sec  3.25 MBytes   425 Kbits/sec
[  3]  0.0-66.4 sec  2.25 MBytes   284 Kbits/sec
[SUM]  0.0-66.4 sec  8.00 MBytes  1.01 Mbits/sec
sgordon4@sender:~$
```

Congestion Plots:







TCP Reno begins in slow start, exponentially increasing CWND until the threshold (ssthresh) is reached, after which it enters congestion avoidance, linearly increasing CWND until 3 duplicate ACKs are received. This switch can be observed at the start of the plot, as the line quickly increases from 1, then changes to a stair shape.

When 3 duplicate ACKs are received in Congestion Avoidance, the slow start threshold (ssthresh) and the CWND value are set to half the current CWND value. This can be observed at any point the graph suddenly drops.