

CS3210 Computer Networks Lab  
Even Sem. 2017, Prof. Krishna Sivalingam  
Lab 8: TCP Protocol based Experiments  
Due date: April 1, 2017, 11PM, On Moodle  
Individual or Two-Member Group

March 21, 2017

## 1 Description

*Note: Viva voce exam and demo will be conducted in the presence of the instructor.*

The objective of this project is to evaluate the performance of TCP protocol variants, as detailed below.

1. Two machines (laptops or even virtual machines) will be connected to each other. You will require *sudo* access on these machines.
2. The parameters to be varied are:

- SACK option, to enable Selective Acknowledgements: ON/OFF  
# sysctl -w net.ipv4.tcp\_sack="1"  
# sysctl -w net.ipv4.tcp\_sack="0"
- TCP Window Scaling option: ON/OFF  
# sudo sysctl -w net.ipv4.tcp\_window\_scaling="1"  
# sudo sysctl -w net.ipv4.tcp\_window\_scaling="0"
- The receiver and sender window sizes can be adjusted using similar commands. This can also be handled by directly editing the configuration file as shown below:

```
#sudo gedit /etc/sysctl.conf

net.core.rmem_max = 16777216
net.ipv4.tcp_rmem = 4096 87380 16777216
net.ipv4.tcp_wmem = 4096 16384 16777216

net.ipv4.tcp_sack=1
net.ipv4.tcp_window_scaling = 1

# sysctl -p /etc/sysctl.conf

# sysctl -a
```

All units are in bytes. In cases where there are 3 values, the first is the Minimum Guaranteed Buffer Size by the kernel, second is Default Buffer Size, and third is the maximum buffersize.

Thus, the value of 16777216 for `net.ipv4.tcp_rmem` implies 16KB. You can set this to be larger than 64KB, if TCP window scaling option is turned ON.

- Congestion Window Schemes:
  - To obtain the set of available TCP Congestion Control schemes, type:
 

```
# sudo sysctl net.ipv4.tcp_available_congestion_control
```
  - To print the currently enabled Congestion Control protocol:
 

```
# sysctl /proc/sys/net/ipv4/tcp_congestion_control
```
  - To select a specific Congestion Control protocol:
 

```
# sudo echo reno > /proc/sys/net/ipv4/tcp_congestion_control
```
  - To add a new TCP Congestion Control scheme to the kernel, type:
 

```
# sudo /sbin/modprobe tcp_htcp # sudo /sbin/modprobe tcp_reno etc.
```
- Using *netem* and *tc* to change the link parameters:

```
# tc -s qdisc ls dev eth0

# tc qdisc add dev eth0 root netem delay 100ms 10ms 25%

# tc qdisc change dev eth0 root netem loss 0.3% 25%
```

Explanation: Sets link delay to 100ms delay with 10ms variation. This causes the added delay to be 100ms +/- 10ms with the next random element depending 25% on the last one. This will cause 0.3% of packets to be lost, and each successive probability depends by a quarter on the last one.

## 2 What to do?

1. Run *httpserver* (python provides SimpleHTTPServer) on one machine and a *wget* client on the other.  
(OR)  
Run *iperf* server on one machine and *iperf* client on the other machine (the client can use the “-f filename” option, to specify a file name to be transferred). Report the server side bandwidth.  
Transfer files from one system to the other and measure the throughput and time taken. File Sizes: 512KB, 1MB, 2MB. (Larger file sizes can also be attempted).  
For each file size (and parameter combinations as below), five repeated experiments must be conducted to measure the average throughput and latency values.
2. The experiments will be conducted for a total of 32 combinations (and three file sizes in each case):  
SACK ON/OFF; Window Size of 16KB and 256KB; TCP Reno and TCP Cubic; Link Delay of 2ms and 50ms; Link Drop Percentage (at Receiver) of 0.5% and 5%.
3. The results will be presented using tables and graphs. You can choose how you wish to organize the tables.  
The results have to be analyzed in detail. Present your conclusions in terms of which of these parameters have major impact on system performance.

## 3 Help

1. Ask questions EARLY and start your work NOW (really, no choice). Take advantage of the help of the TAs and the instructor.
2. Some URLs that can provide additional information:
 

```
http://www.slashroot.in/linux-network-tcp-performance-tuning-sysctl
```

```
http://www.drdobbs.com/linux-kernel-tuning-using-system-control/199101660
```

<https://www.mjmwired.net/kernel/Documentation/networking/ip-sysctl.txt>

<https://wiki.linuxfoundation.org/networking/netem>

<https://www.cyberciti.biz/faq/linux-traffic-shaping-using-tc-to-control-http-traffic/>

3. Submissions PAST the extended deadline SHOULD NOT be mailed to the TAs. Only submissions approved by the instructor or uploaded to Moodle within the deadline will be graded.
4. Demonstration of code execution to the TAs MUST be done using the student's code uploaded on Moodle.
5. NO sharing of code between students, submission of downloaded code (from the Internet, Campus LAN, or anywhere else) is allowed. The first instance of code copying will result in ZERO marks for the Lab component of the Course Grade. The second instance of code copying will result in a 'U' Course Grade. Students may also be reported to the Campus Disciplinary Committee, which can impose additional penalties.
6. Please protect your Moodle account password. Do not share it with ANYONE. Do not share your academic disk drive space on the Campus LAN.

## 4 Grading

- Experiments demonstrated correctly: 60 points
- Report: 30 points
- Viva Voce: 10 points