TCP Lab

Readings:

http://linux.die.net/man/1/ttcp
http://linux.die.net/man/7/tcp
http://linux.die.net/man/8/tc
http://linux.die.net/man/8/sysctl
http://en.wikipedia.org/wiki/TCP_window_scale_option

Theory Questions:

1. [5] What would increasing the MTU do for throughput on a lossless link?
2. [5] How would decreasing the TCP Window size affect throughput?
3. [10] What would happen to your application latency if you had an infinite size buffer, and you were receiving more data than you could process?
4. [10] What assumption does TCP make about packet loss?
5. [10] Briefly explain the following mechanisms of TCP:
    a. Slow Start
    b. Congestion Avoidance
    c. Fast Recovery/Fast Retransmit

Lab Questions:

For this section you will be using ttcp to send TCP data segments of the network. Use the –l parameter to change the length, and –n to change the number of packets to send.

1. [5] Using ttcp, send 1000 packets of size 1400. **Submit a screenshot of the Tcptrace Time Sequence Graph.**
2. [35] Modify TCP's send and receive memory buffers. Set the memory values for both buffers to be min = 10000, default = 75000, max = 12500000. (note: depending on you modify these values, reloading the configuration from the file may be required). **Submit this screenshot of the Tcptrace Time Sequence Graph as well.** Describe the differences between part (1)'s graph and this graph.
3. [45] Change the memory buffer values back to default. Set eth0 device to have a 5%-15% loss. You can do this prior to sending traffic. Your objective is to create a Tcptrace Time Sequence Graph for which you can both show and explain where

Slow Start and Congestion Avoidance phases begin and end. **Submit both graphs.** You may modify –n and –l arguments as you see fit.

4. [15] For this section, make sure the commands are pre-typed, and that the –n value is large enough to make sure you execute another command (in another terminal) simultaneously. You will need to, during execution of the ttcp command, set the link loss to 100%. After 10 seconds, set the loss back 0%. **Submit the Tcptrace Time Sequence Graph.**

Clarification of deliverables:

1. Tcptrace Time Sequence Graph
2. Tcptrace Time Sequence Graph and a description of the graph
3. Tcptrace Time Sequence Graph that shows SS + CA phases
4. Tcptrace Time Sequence Graph that shows 100% loss for 10 seconds

Hints: For link loss, the command *tc qdisc* will be needed. When you first use the command you should use *add dev* for the device you plan on changing. It only needs to be set on the sender's side. After adding the device use *change dev*.

This is not meant to be a straightforward lab. You will need to look into the man pages and Linux configuration settings.