Austin Lolli

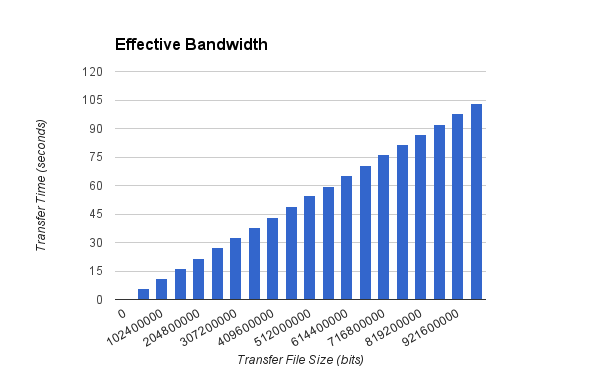
COEN 146

**Networks Lab 2 Writeup**

**The Script:**

The purpose of the script (script.sh, given to us by our TA) is to go through a sequence of transmitting random bits as a file of increasing size over an ssh connection, and then report out the time for the entire process. The outermost while loop specifies the entire process will happen 10 times, which is done to give us a big enough sample size for a reliable average when calculating our effective bandwidth. Each run inside of this while, the file size being sent is increased, starting at zero and increasing in size by the blockSize (1024) constant multiplied by the count variable, resulting in file sizes incrementing by 51200000 bits (6.4 Megabytes), ranging from size of 0 bits up to 972800000 bits (121.6 Megabytes). For each file size and trial, the script outputs the times for user, sys, and real to the log.txt file, times related to the completion of the file transfer over ssh connection. For the purpose of calculating effective bandwidth, we took the sys and user time and added them together. Why this is done is explained below with the graph.

**The Graph:**



The graph shows the average sys + user time for each run for the given size of file. This is effectively the entire time the file transfer took. These two times are added together because user is the amount of time the process spends outside of kernel mode, and sys is the amount of time spent in kernel, or privileged mode. Added together we get the total time the events inside the time command occurred (the transferring of data over the ssh connection). By calculating the effective bandwidth (the time it takes to completely transfer the data of the file) to be the size of the file divided by the time taken, most of the bandwidths for the varying file sizes hovered around 9.3-9.45 Megabits per second, with an overall average of 9.39 Megabits per second. For the estimate of time taken on things other than data, the test runs of file sizes of zero become helpful here. Aside from a couple outliers, most all runs were within .03 of the average of .043 seconds for the empty file runs. This leads me to believe that establishing a connection is very fast, especially relative to the transferring of data.