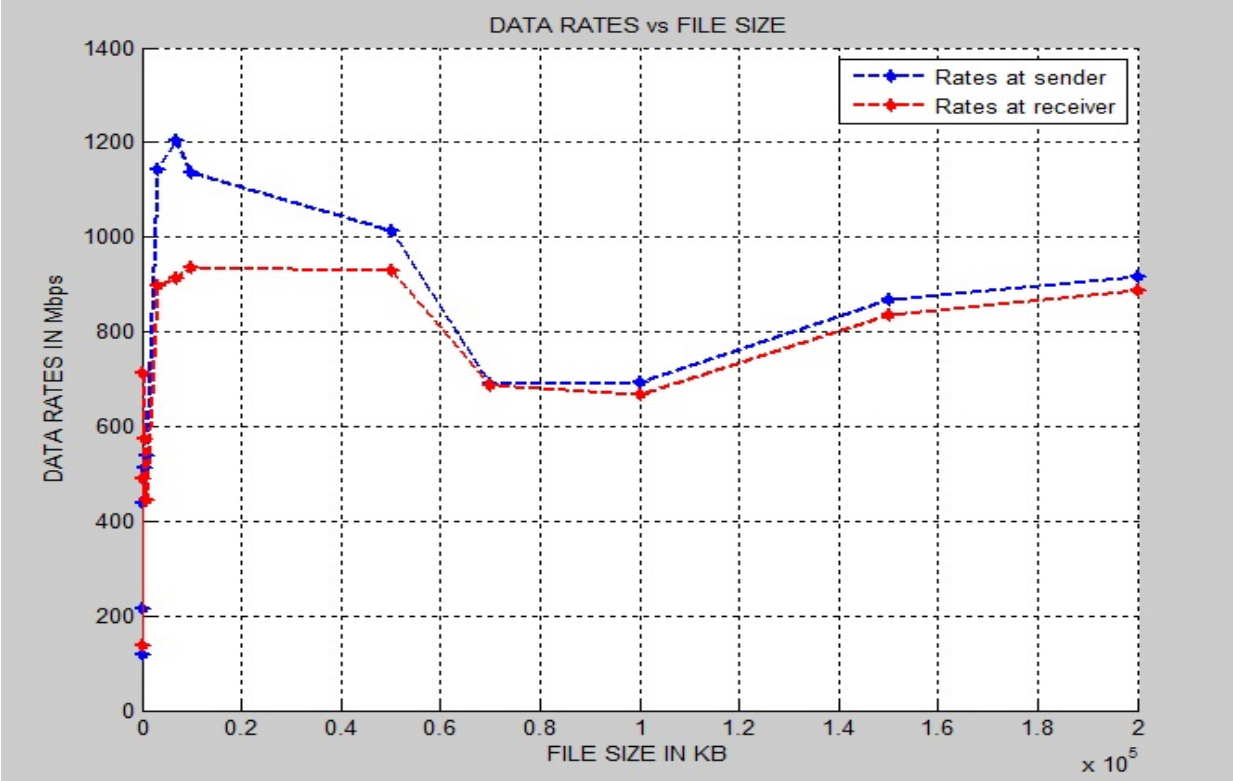
CSE-589
PA1-Remote File Sharing System
Analysis

1. Data Rates vs File Size:

Constants: Packet Size – 1000 bytes



Observations:

1) For very small files with bytes < 6000 bytes, Data rates went considerably less around 190 Mbits/sec and below.

2) Starting at 6KB to 10 MB data rates at server and receiver went as high as 1403 Mbits/sec, 1392 Mbits/ sec respectively.

3) For files bigger than 10 MB i.e 50 MB to 200 MB, sender and receiver Data rates gradually fluctuate between 700 to 900 Mbits/sec.

Analysis:

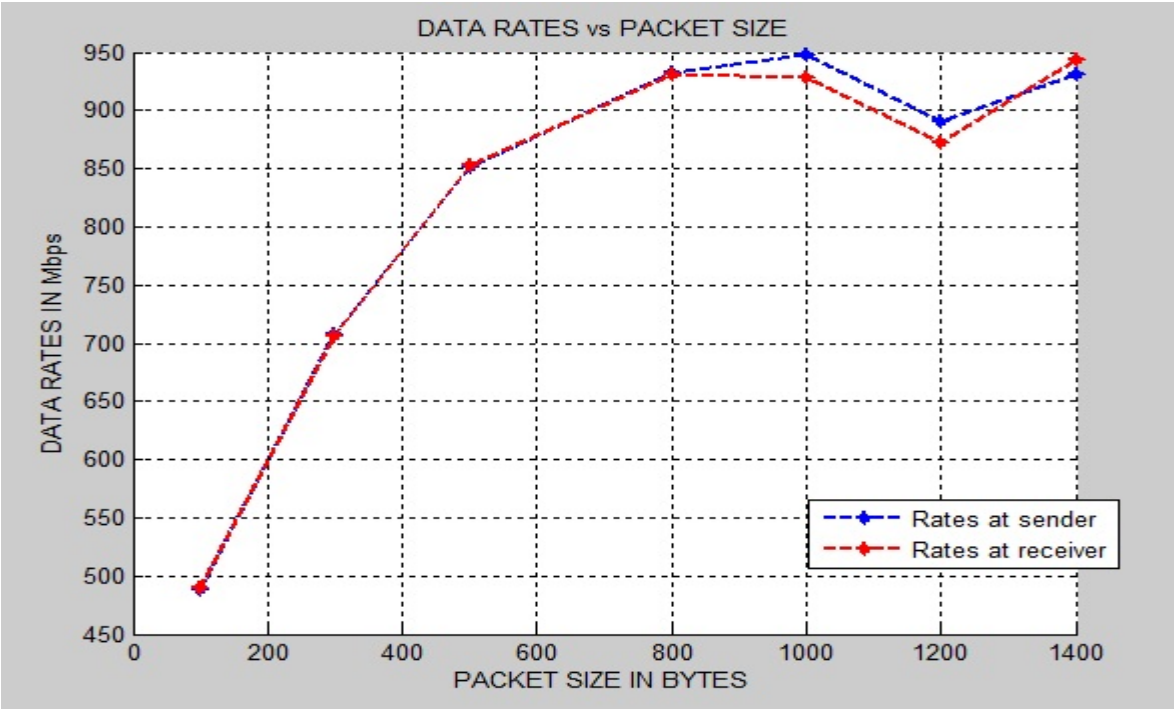
Firstly, the plotted data rates include downloading the file via network (say, Download rate) and reading from and writing into the file

Consider download rate, it depends on the bandwidth of the network rather than the file size. The maximum a download rate can go is up to the link bandwidth. Here we are downloading one file at a time via TCP, therefore we are utilizing entire available bandwidth to download. Note that the downloads are one directional i.e no cross traffic in the link. The download rates observed should be maximum when downloading in this setup.

Reading from and writing into the file at sender and receiver ends doesn't in any way effect the network download rate. If we are calling fwrite/fread with large chunks of data, it could add a small amount of overhead. Since in this implementation we are writing small chunks i.e 1000 bytes which results in timing to be close to negligible.

Observations do agree with my analysis of the system. From the graph, download rates mostly try to catch up 1000 Mbps. Hence we could assume that the bandwidth could be around 1000 Mbps.

2. Data Rates vs Packet Size:



Observations:

File Size: 200 MB

Transferring File of size 200 MB over different packet sizes ranging from 100 Bytes to 1400 Bytes resulted in,

1. Transfer time ranging from 5 sec to 1.8 sec.

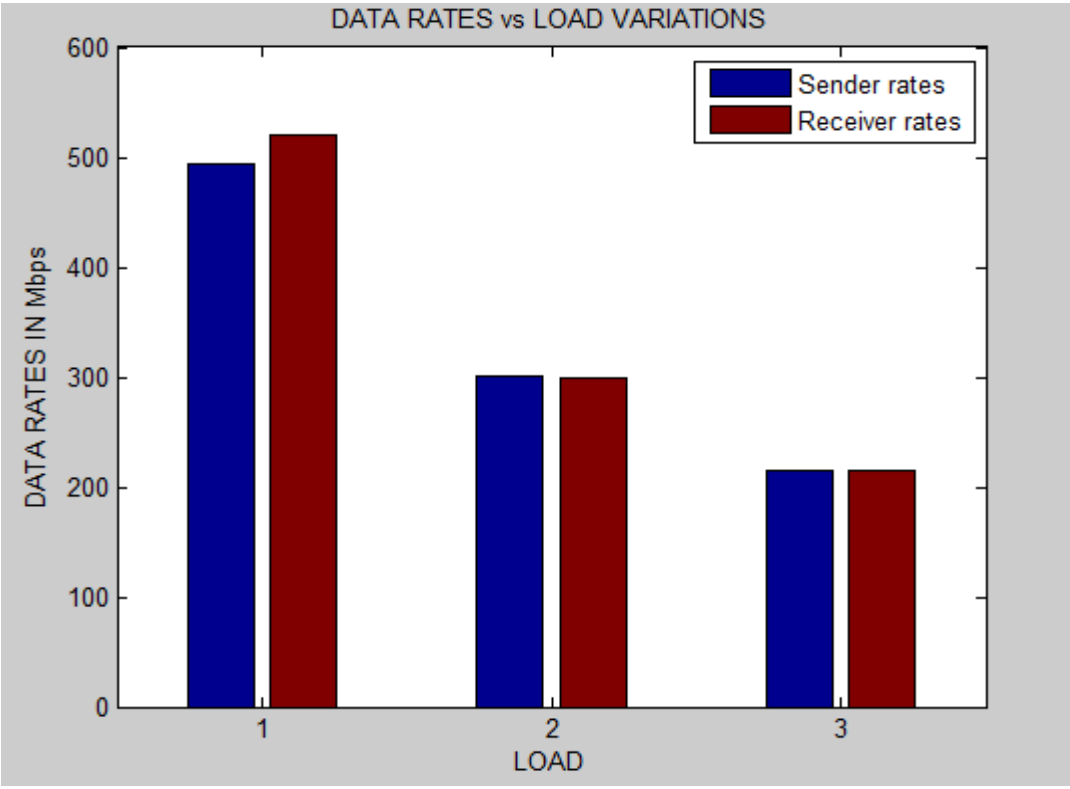
2. Data transfer rates ranging from 472 MBits/sec to 952.4 MBits/sec

Analysis:

The send and receive socket buffers should be sufficiently large so that the network link through which the transfer is being made can be optimally utilized.
Generally, If the Bandwidth delay product (BDP) is R*D(prop). If packet size is < BDP, we are under utilizing the link. If the packet size > BDP we overload the link.

Therefore with gradual increase in packet size we will utilize more link bandwidth with faster delivery of packets. Hence, we observe increase in data rates.

4.3 Data Rates vs Load Variations:



Observations:

Constants: File Size - 70 MB, Packet Size- 1000 Bytes

1. When one client downloads a file with above constants the sender(Tx) and Receiver (Rx) speeds are 510.5 Mbits/sec and 493.4 Mbits/sec respectively.

2. When two clients download files with same size in parallel, Observed sender(Tx) and Receiver (Rx) speeds are around 301.2 Mbits/sec and 298.48 Mbits/sec respectively.

3. When three clients download files with same size in parallel, Observed sender(Tx) and Receiver (Rx) speeds are around 214.2 Mbits/sec, 214.6 Mbits/sec and 212.3 Mbits/sec respectively.

Analysis:

When we vary load over the network from 1 to 3. All the clients if trying to download in parallel, utilize the available link bandwidth together.

Why? Since TCP is connection oriented, Network consistently should allot its available bandwidth to this open connection. Now when you increase the load, the available bandwidth is shared by number of clients using this network.

Observations do comply. From observations, say available bandwidth is nearly 600 Mbps. It is shared accordingly when two clients are downloading, 301.2 and 298.48 sum up to nearly 600 Mbps. And when three clients are downloading, 214.2 Mbps + 214.6 Mbps + 212.3 Mbps also sum up to available bandwidth.

4.4 iperf and measuring bandwith:

Observations :

1. Available bandwidth when iperf outputs for 1 client : 899 MB/sec

2. Available bandwidth when iperf outputs for 2 client : 637 MB/sec and 475 MB/sec

3. Available bandwidth when iperf outputs for 3 client : 605 MB/sec 298 MB/sec 343 MB/sec.

To compare iperf with my file sharing application, I sent 1GB file to 1 client and got 501 MB/sec and then this bandwidth is shared depending on load. Every time they do sum upto around 500 Mbps.

The values themselves don’t conform with iperf, as there is inaccuracy in obtaining the time when download begins in my implementation (UPLOAD command gives exact same rates i.e 887 Mbps as iperf). But, overall division and sharing of bandwidth when compared with my implementation is quite the same.