

Programming Assignment #1

Due date & time: 11:59pm on Sept 23, 2014. Instructor: Sonia Fahmy TA:Shin-Yeh Tsai

Student Name: **Di Jin**

Student PUID: **0025274908**

A. Data Rates vs. File Size Answer. According to the condition, the packet size is constant, 1000 Bytes, and the file size changes through a set of values.

Observations: With a fixed packet size, we observed what is expected, that is the transferring time grows linearly with the file size when the file size is larger than 10MB. And the transferring rate is around $5.75 * 10^8 bps$. This rate should be the peer-to-peer bandwidth of the network. When the file size is smaller than 10MB, due to the buffering during the transfer, we observed a higher transfer rate. Because the `send()` returns when the data is buffered in the socket, not necessarily received. And it's same for `recv()`.

File Size: (Bytes)	1K	4K	16K	64K	256K	1M	4M
Tx: Time (usec)	25	73	118	226	1400	7649	43528
Tx: Rate (bits/sec)	$3.28 * 10^8$	$4.49 * 10^8$	$1.11 * 10^9$	$2.31 * 10^9$	$1.50 * 10^9$	$1.10 * 10^9$	$7.71 * 10^8$
Rx: Time (usec)	11	21	25	67	898	8908	53447
Rx: Rate (bits/sec)	$7.45 * 10^8$	$1.56 * 10^9$	$5.24 * 10^9$	$7.83 * 10^9$	$2.34 * 10^9$	$9.41 * 10^8$	$6.28 * 10^8$
File Size: (Bytes)	10M	20M	30M	40M	50M	60M	70M
Tx: Time (usec)	132924	280990	431329	579047	728346	877046	1025930
Tx: Rate (bits/sec)	$6.31 * 10^8$	$5.97 * 10^8$	$5.83 * 10^8$	$5.79 * 10^8$	$5.76 * 10^8$	$5.74 * 10^8$	$5.72 * 10^8$
Rx: Time (usec)	143215	290449	437816	584544	731229	878104	1025230
Rx: Rate (bits/sec)	$5.86 * 10^8$	$5.78 * 10^8$	$5.75 * 10^8$	$5.74 * 10^8$	$5.74 * 10^9$	$5.73 * 10^8$	$6.28 * 10^8$

Table 1: Fixed packet size = 1000, vary file size.

B. Data Rates vs. Packet Size Answer. According to the condition, the file size is constant, 50 MBytes, and the packet size changes through a set of values.

Observations: Changing the packet size for `send()` and `recv()` calls does not have an significant influence on the transferring time. The bandwidth usage is always close to the upper limit. This might because that the packet size used in `send()` and `recv()` has nothing to do with the real packet size used by network protocol. That is, the data is buffered and sent/received with a packet size defined by system network implementation.

Packet Size: (bits)	100	300	500	700	900	1100	1300
Tx: Time (usec)	742817	737061	736560	745333	737935	734860	736485
Tx: Rate (bits/sec)	$5.64 * 10^8$	$5.70 * 10^8$	$5.69 * 10^8$	$5.63 * 10^8$	$5.68 * 10^8$	$5.70 * 10^8$	$5.70 * 10^8$
Rx: Time (usec)	734798	732347	731785	737160	736146	730346	733543
Rx: Rate (bits/sec)	$5.71 * 10^8$	$5.73 * 10^8$	$5.73 * 10^8$	$5.69 * 10^8$	$5.68 * 10^8$	$5.74 * 10^8$	$5.71 * 10^8$

Table 2: Fixed file size = 50MB, vary packet size.

C. Data Rates vs. Load Variations Answer. According to the condition, the packet size is constant, 1000 Bytes, and the file size is constant, 50 MBytes. The connection to the host machine ranges from one to five.

Observations: When there are more than one sender to upload file to the host, all the connection will share the communication channel and the file transfer rate is affected. The more sender we initiated, the lower transfer rate we get. The first initiated sender always shares higher bandwidth than others. Because when it's initialized, there is no other competitors to share the bandwidth.

D&E. Analysis of iperf measurement Answer. For this question, iperf is used to measure network bandwidth.

Observations: More clients exist in the channel, less bandwidth each client will have. For instance, in my test, when only one client exist, the bandwidth is 570 Mbits/sec. When there are two clients, the bandwidth for one client is 297 Mbits/sec and for the other is 261 Mbits. Notice that the bandwidth to the server is not shared uniformly among clients. And the summation of total transfer rate is higher than a peer-to-peer rate.

#Sender	1	2	3	4	5
Tx: Time (usec)	732504				
Tx: Rate (bits/sec)	$5.73 * 10^8$				
Rx: Time (usec)	731423				
Rx: Rate (bits/sec)	$5.73 * 10^8$				

#Sender	1	2	3	4	5
Tx: Time (usec)	1042890	1307700			
Tx: Rate (bits/sec)	$4.02 * 10^8$	$3.20 * 10^8$			
Rx: Time (usec)	1046990	1086690			
Rx: Rate (bits/sec)	$4.01 * 10^8$	$3.86 * 10^8$			

#Sender	1	2	3	4	5
Tx: Time (usec)	842489	1736140	1726660		
Tx: Rate (bits/sec)	$4.98 * 10^8$	$2.42 * 10^8$	$2.43 * 10^8$		
Rx: Time (usec)	846006	1546800	1719560		
Rx: Rate (bits/sec)	$4.96 * 10^8$	$2.71 * 10^8$	$2.44 * 10^8$		

#Sender	1	2	3	4	5
Tx: Time (usec)	2415510	876975	2566030	2053260	
Tx: Rate (bits/sec)	$1.73 * 10^8$	$4.78 * 10^8$	$1.63 * 10^8$	$2.04 * 10^8$	
Rx: Time (usec)	881763	2405830	1809840	2335290	
Rx: Rate (bits/sec)	$4.76 * 10^8$	$1.74 * 10^8$	$2.32 * 10^8$	$1.80 * 10^8$	

#Sender	1	2	3	4	5
Tx: Time (usec)	1374360	2798560	2752770	2807680	2692700
Tx: Rate (bits/sec)	$3.05 * 10^8$	1.50^8	$1.52 * 10^8$	$1.49 * 10^8$	$1.55 * 10^8$
Rx: Time (usec)	1404910	2824590	2754460	2799390	2680710
Rx: Rate (bits/sec)	$2.98 * 10^8$	$1.48 * 10^8$	$1.52 * 10^8$	$1.49 * 10^8$	$1.56 * 10^8$

Table 3: Fixed file size = 50MB, fixed packet size = 1000B, current transfers

#Client	1	2	3	4	5
Rate: (MB/sec)	570				
	297	261			
	240	156	229		
	216	142	131	168	
	233	76.7	139	119	142

Table 4: Results of iperf.